SECURITY NOW!

Transcript of Episode #55

Application Sandboxes

Description: Having discussed “heavy weight" virtualization technology in recent weeks, this week Steve and Leo examine “lighter weight" application sandboxing technology and the software solutions currently available to perform this form of application “wrapping." They discuss the inherent limitations of sandbox security and explain how valuable sandboxes can be for privacy enforcement.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-055.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-055-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 55 for August 31, 2006: Application Sandboxes.

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

The last day of the month, and in some ways the last day of summer. Let’s check in with Steve Gibson.

Steve Gibson: And you’ve survived your cruise.

Leo: I survived the cruise. The ship did not go down, unless it did, in which case I’m doing this podcast posthumously. And if it is, please don’t cry.

Steve: What a horrible thought, Leo.

Leo: The last podcast.

Steve: We might have lost you in the Arctic.

Leo: Well, no, you know, we’re taping this, obviously, before I leave on the cruise. But it’s going to be an Alaska cruise, and it’s a geek cruise, but it’s kind of an unusual one because it’s chess.

Steve: We’re taping this?

Leo: Shhh. No one knows. If I had – tape is not quite the word we use.

Steve: Uh huh, exactly.

Leo: I say “film” on the TV show, too. I just – I guess I’m an old-timer. I’m an old-timer. Now, we’re actually – there’s no tape involved in the creation of these. It’s all bits. Once it leaves the microphone and gets into the mixer, it all is bits.

Steve: Yeah, well, when you referred to Mitch Miller last week, that pretty much did it.

Leo: That was it. They knew. They knew. I’m – pretty soon I’ll be saying the Internet’s a series of tubes, and it’ll be all over.

Steve: The jig is up.

Leo: Anyway, the chess cruise is fun because I’ve been working on a lecture for it. And they let me go on, you know, these geek cruises if I do a lecture. And, you know, I played chess as a high school kid, but I never – I’m not a great player by any means, just a, you know, an average club player. But I always loved the game. And I said, well, I’ll tell you what, I’ll do a lecture on the history of chess-playing machines because...

Steve: Oh.

Leo: ...in many ways the history of computing is parallel to the history of these chess-playing machines. It was inspired – Babbage was inspired by a chess-playing automaton to think of a computer, of a machine that could think. Turing and many others have been, you know, heavily influenced by the idea of making a computer that could play chess. So...

Steve: Well, and in fact, of course, it was one of the – it was the holy grail of the original AI movement back in the ‘70s.

Leo: Precisely. You know, it’s interesting, what I found in my research is, it’s solved, and it is no longer used in AI as a goal in any means. They...

Steve: No longer interesting, huh?

Leo: It’s not interesting. They feel they’ve solved the problem. So what do they use now, they use the game of Go, which is hideously complex. It's a 19x19 board. And it's much more positional and intuition based. It's very hard to calculate. So...

Steve: Well, and of course they solved chess by cheating. They...

Leo: Well, they say that, but, you know, they're pretty good.

Steve: No, but, I mean, you know, the way they wanted to solve it was by getting a machine to understand...

Leo: Right.

Steve: ...chess.

Leo: Right.

Steve: I mean, that was the point of artificial intelligence. It was...

Leo: To think.

Steve: Yeah, exactly, well, to understand. And what they're really doing it, the machines have just gotten so fast and so parallel that they've got all these opening books that are preprogrammed in...

Leo: They still have judgment. They still have to do some judgment. They can't just – there are more positions possible than you could calculate even today.

Steve: Right. Well, and so they're doing alpha-beta pruning and determining what directions make sense for further exploration. I mean...

Leo: Precisely.

Steve: ...a ton of technology in there, but they didn't end up doing it the way they thought they were going to end up doing it.

Leo: No, they don't think. But I think maybe that's a moral in some ways is that machines won't really think in the way that humans think, but they can be made to accomplish the same tasks in a way that's...

Steve: To mimic.

Leo: To mimic in a way that's unique to machines. They think, but not as humans do, I guess you could say. And that's exactly what the conclusion of the talk is, is what have we learned? And it's just, it is, it's a very interesting area. I don't – somehow I'd like to get this information up on the 'Net. Maybe I can figure out a way to do that, but...

Steve: Well, are you going to record your speech?

Leo: Yeah. Maybe I'll just put a recording up. That'd be the best thing to do.

Steve: Yeah.

Leo: And there are actually very good recordings. There's a two-hour video from the Association of Computing Machinery, the ACM, that was done a couple of years ago at the Computer History Museum, with many of the pioneers in the field. And it's fascinating.

Steve: Well, you can just start that up and then go hit the buffet.

Leo: I was thinking of that. You know, I could talk, but why should I talk when you can get David Levy and, you know, the guy who invented LISP, I mean, some really brilliant guys here telling you this story. But I'm going to – maybe I'll show some clips from it. But that's on Google Video, though. I'll put a link in the show notes if people want to know more about that.

Enough of that. We want to talk about security.

Steve: Yup.

Leo: Anything to cover from previous episodes before we head into our topic of the week?

Steve: I think not this time.

Leo: Okay. So what's the topic of the week, I guess?

Steve: This week, in our continuing series on virtual machine technology, I want to talk about application sandboxes.

Leo: All right.

Steve: It's a really interesting topic. And the way it fits into the spectrum is also sort of interesting. We started off talking about, you remember, of course, the original history of virtual machines, back when Mitch Miller was still singing. Then we talked about VMware, which I would consider that sort of like medium weight to – or maybe heavyweight virtualization, I mean, it really is...

Leo: It's kind of the current – it's the current state of the art, yeah?

Steve: Yeah. Although last week we talked about Blue Pill, which is like – in a sense it's like super heavyweight virtualization in the sense that it's even – it operates at a level underneath where VMware operates. And there's another application range which is the topic for this week, and that is sort of lightweight virtualization, lighter weight virtualization, and what the effects are, and the tradeoffs and so forth.

Leo: All right.

Steve: An interesting – now that we have, like, several perspectives on virtualization, we have enough that we can sort of – we can develop a way of thinking sort of generically about this notion of virtualization levels. For example, what the SVM technology and the Pacifica technology that AMD developed, and Vanderpool's VT technology from Intel, what...

Leo: We talked about all these last week, but they're essentially built into the chip hardware virtualization technologies, right?

Steve: Well, exactly. And what – the way to think about them is that they create virtual chips, that is, they virtualize the processor engine itself so that multiple operating systems believe these are multiple processors. So they're, like, they're virtualizing at the hardware level to create multiple instances of the computer system hardware virtually.

Leo: Hmm.

Steve: And so that, I mean, and that's an important way to look at it because then the next level up is – it's software running on normal hardware, which uses some of the processor features to create the illusion of multiple chips, so that you can run operating systems in multiple on the platform. The point is that you don't actually have the same level of hardware virtualization. So you've got to do a lot more work, which is where VMware is, in order...

Leo: So it's almost like time slicing the processor to make it look like there are multiple processors.

Steve: Well, it's doing exactly that. And it's also having to play a lot of games to make up for the fact that you don't actually have virtual processor support, which our next-generation 64-bit chips will have.

Leo: And all of that uses up cycles and so makes it less efficient.

Steve: Right. And so, as we come higher, then you might think of an operating system itself as providing virtual environments for its applications. For example, you know, I can run a DOS shell or DOS program in a DOS box. And as we talked about several weeks ago, that DOS program thinks it's on DOS.

Leo: Right.

Steve: I mean, it was software that was written 20 years ago for MS-DOS or PC-DOS. And, I mean, and it's – as I mentioned, in order to put data on the screen, it's writing characters directly to, you now, B800 hex buffer. And but the point is that the operating system has virtualized DOS. It's created a virtual DOS environment pretending to be DOS. So that's like the next level up in this notion of, like, virtual layer architecture. And so, and then you could argue that, you know, that applications themselves are able to create virtual containers, like how Word can be editing multiple documents at the same time. So, you know, the document sort of doesn't have the awareness that a running program does.

Leo: Right.

Steve: But, you know, so there are these layers. Well, one of the interesting things that we're beginning to see, and this is really an offshoot of security and privacy concerns, is this notion of sandboxes. We originally talked about it when we began this series in the context, for example, of the well-known Java Virtual Machine, the JVM, the idea being – and this was, you know, Sun's concept from the start was that they would create a virtual architecture in which you could run this Java byte code. The Java byte code would, I mean, it could be implemented in an actual hardware architecture, but it wasn't designed to. It was designed to be emulated.

So this Java Virtual Machine is an emulator of a sort of a fictitious machine that was never actually – that never actually existed. And the point is, that kind of emulation allows you to sandbox the Java, meaning that because the emulator is actually doing all the work, that is, the Java is not Intel or Power PC or whatever chip, it's not the Assembly language or the machine code of the chip, it's an abstraction of this machine. Because the emulator is reading those instructions and doing what it chooses with them, you're able to impose a very rigid set of strictures over what the code is able to do. So, and thus the so-called Java sandbox, the idea being that it's safe to run Java because, unless the enclosing emulator, the virtual machine, wants to have the program access the hard drive or access the 'Net or do whatever it wants, the program can't. It's entirely beholden to what the emulator is willing to do for it.

Leo: Unless there's a hole, of course, an exploit or a security bug. But...

Steve: Exactly.

Leo: Assuming it's a perfectly written VM.

Steve: Exactly. And so that's the theory. Well, now we have, you know, Windows or Mac or Linux, we have operating systems that are very complicated and have this concern both with privacy and security. And so the question is, how can we – and can we use this virtual machine technology in some way to create enclosures or sandboxes around our own applications, around applications where we want them not to be able to reach out and touch our hard drives or send data out on the Internet in order to work behind our backs. So what has sprouted up is a bunch of utility sort of software which pretends or purports or tries to create this sort of technology.

Now, it's confusing because in my updating myself on the state of the art here, as I did for this podcast, I found all kinds of claims being made by these sandboxing programs which are just not true. What's happened is that this – because of the upsurge of excitement and popularity about virtual machines, they're all now saying, you know, creates a virtual computer within your computer. Okay, except it doesn’t. In most cases. There are, well, for example there’s a program called Sandbox IE, or Sandboxie, as it’s normally called. There’s one called Green Border. There’s – Fortres Grand has several sandboxing programs, a Virtual Sandbox. Version 1 is free; version 2 you pay for. There’s a company called MetroPipe that has something called the Portable Virtual Privacy Machine. There’s VELite from a company called SecureOL; a product called BufferZone, and one called RunSafe, you know, there’s a bunch of utilities which are jumping onto this bandwagon of so-called virtualization, and using that term improperly.

Now, I actually mentioned one which is not using it improperly, and that’s this Portable Virtual Privacy Machine from a company called MetroPipe. They’re actually – they’ve packaged the QEMU emulator and, I mean, actually trying to get Linux to boot. In my experiments with it, it fired up a DOS window, it looked, you know, I saw the signs of a BIOS that was booting. It even began to decompress the Linux kernel. Then it all exploded.

Leo: So, well...

Steve: It’s like, okay, well – and I thought, well, maybe it doesn’t like my machine, so – or I don’t need to reboot normally very often. So I shut everything down and restarted Windows. And this is Windows 2000 that I’m still using on my main workspace. And I tried it again, and it decompressed the Linux kernel and then exploded. It’s like, okay, well, I’m not recommending that one. In any case, apparently it is an emulator and very slow. There are some – QEMU has moved forward with its technology to take advantage of more virtualization hardware that’s available. But apparently it’s still got a ways to go.

Okay. So what sandboxing does is two things. Or the way to think about these sandboxing solutions, one is security, which is for protection from malware. The idea that you could run IE or Firefox or Opera or whatever inside a sandboxed environment, where it would be unable to do anything to your machine. So security is one aspect. The other is privacy because, I mean, of real concern for people, especially, for example, if they’re at a friend’s house or they’re using a public computer or, you know, it’s not their own machine, the idea of leaving footprints behind. You know, I mean, we know that browsers have caches that keep cached images of chunks of web pages that have been downloaded. They have, you know, past URL libraries. They’re constantly storing cookies on your machine.

And of course traditionally, I mean, there have been horror stories throughout history, throughout Internet history, of employers, you know, checking employees’ browser caches to, you know, see what – where they’ve been and what they’ve been doing, and often finding, let’s just say stuff on the computer that seems non-work related. So the idea is, from a privacy standpoint, is, you know, can we use this sandboxing technology to basically securely wipe out any trace of what we’ve done?

Now, I ought to say that VMware or Virtual PC, you know, the real industrial-strength true virtual machine systems, which are available from VMware and Microsoft, those are creating a true OS boot environment and really offer robust containment. Unfortunately, none of these other sandbox utilities do that. What they do is clever and useful and valuable, and in some cases free. And I’ll talk about some specifics here in a second. But I want to explain the technology so that people understand the limitations and don’t ask or trust these things further than they should. What they’re doing is exactly like what we’ve talked about many times in the past, in fact just recently when we were talking about Blue Pill, and I was talking about how rootkits get into the kernel and intercept functions in the operating system for the specific reason of hiding themselves. They will unlink themselves from the process list so that simple process exploring won’t reveal them. They will filter the application program interface, the API, that the operating system publishes to its applications, they’ll filter that so that any reference to them is removed in the applications, you know, listing the directory contents of the drive.

Well, what these sandbox programs are all doing is the same. Oftentimes there are kernel-level components because they need to get into the system in a deep way. But they are – they then filter the operating system functions which programs need to use in order to store stuff persistently on the hard drive. So, for example, they will intercept the file open and the file read and the file write calls that applications make to the operating system in order to perform disk I/O. They will intercept and filter the registry reading and updating, modifying, writing calls in order to – essentially to impose a layer of supervision between the application and the operating system.

Now, it's cool, and it works. But you can't trust it. And that's the problem, is that they're overselling what the technology can really be used to guarantee. And so I want to, I mean, I want to end up talking about how valuable these things are and where they're valuable, but to explain that this is something which is inherently error prone and accident prone. If applications – and it sort of comes back to what Joanna was talking about with Blue Pill, the idea being that, if software knows – if some software knows what other software is doing, it can work around it. And so the problem with these sandboxing programs, which are not true – which are not creating true virtual machine environments, is, you know, the problem is that the program is still running on the operating system. And if the program wanted to work itself around containment, it almost certainly can. So we're back in sort of that cat-and-mouse game where, if the program, you know, the traditional virus/antivirus program, malware/anti-malware, cat-and-mouse, the idea being that you're trying to erect an artificial barrier to restrict the things that a program can do, but you're using software to impose the restrictions. Well, there are other ways to achieve the same things using nontraditional or undocumented operating system calls that may not be filtered and interposed or are too burdensome to filter and interpose. So I would never absolutely trust something that was one of these pseudo-virtualization solutions from a security standpoint.

Now, that's different than from a privacy standpoint, and that's where I think these can have real value. So I just want to make sure that I've been clear. These pseudo-virtualization sandboxes, and I should say that I've got them all listed with links and URLs and some comments on the episode notes page, you know, for this episode, No. 55, on Security Now!. So I've got them listed. I ran through them briefly, but people can go there. And I'll be talking about one in particular in a second that I like most, based on everything that I've seen and sort of doing a quick review of all these options that I've been able to find.

But the true virtualization, for example, that VMware creates and Virtual PC creates, where they're actually booting an operating system, and you are then – or again I should also say the VMware player that we talked about several weeks ago, VMware player that's booting Ubuntu Linux and then running Firefox. I mean, there you're talking seriously good security because the browser is now running in an entire operating system which is running inside of a virtual machine environment, and the security is virtually absolute. The user still has to be responsible for not allowing network exploits, that is, you know, this thing is only useful probably if it's on the Internet, so you need to make sure that it has access to the Internet and not back into your own computer's network, or it can, you know, play filesharing games or whatever.

Leo: Let me ask you about that. What kind of barrier is there? I mean, if I get a Sasser, a Zotob, a network-aware virus in the virtual machine, can it spread to the real machine?

Steve: Yes.

Leo: Okay.

Steve: If you enable those features. For example, one of the things that VMware allows is shared file access. It's very convenient, and in fact I use it all the time. As I've mentioned before, I'll have many virtual machines created, each with a different personal software firewall installed, and I'm jumping between them. Well, it's very convenient to allow all of them to see a shared directory on my real physical hard drive. So, you know, so VMware wants to offer that feature because it's a convenience. But of course with that comes a security consequence. And similarly...

Leo: So you shouldn't assume a virtual machine is absolutely impervious. It isn't.

Steve: Right. I would say the best way to express it would be that a true virtual machine environment, like VMware or Virtual PC or Parallels or, you know, one of those where you're actually booting an operating system within that containment, it is a virtually secure tool. But like any tool, it can be misused. If you download a file in that machine and then drag it...

Leo: Right.

Steve: ...out of the virtual environment onto your real desktop, well, you've just breached that barrier.

Leo: Right, got it.

Steve: So, okay. So I wanted to just be very clear about what this distinction is. Now, these virtual – these pseudo-virtual environments that unfortunately tend to oversell themselves because they just love that word "virtual," because everyone thinks, oh, that's good, the problem is they're not running in a true secondary instance of an operating system that you had to install. The benefit is that they're much lighter weight, that is, they're easier to install. Remember I mentioned that VMware is a heavy installation. I mean, it's creating virtual adapters, and it's, you know, it takes a while. You've made a commitment when you've even stuck the VMware player on your system. It's something that, you know, you don't just casually walk around with a USB dongle and do it to machines that you encounter.

Leo: Although that time may be coming, given hardware support for virtualization. That may not be so hard to do down the road.

Steve: Oh, I mean, that's a very good point, Leo. You could imagine in this next generation with 64-bit hardware virtualization, you could say, give me an empty machine and run this operating system in it, and it would be, you know, instantaneous. It'd just be like hitting a hotkey.

Leo: Yeah.

Steve: Okay. So what these pseudo-virtual sandboxes do is they interpose themselves between the application and the operating system, attempting to hook and filter any way that the application could make a permanent change to the operating system. You want them not to be able to do that. So, and they really vary in their capabilities. Based on what I saw, one that I would recommend that people play with, if they're interested in experimenting with this, is the first one I called. It started off being a sandbox for IE. Therefore its name was Sandbox IE, or Sandboxie. It's just Sandboxie.com. What I like about it is that it appears to be very good, I mean, these guys have got ActiveX running in a virtual container. It's gone way beyond IE. You can pretty much sandbox any browser and application that you want to. It is very lightweight.

It's about a 245K – 245K – so less than a quarter megabyte download, which expands to about a megabyte, about 900K, in a directory. You can install it without rebooting, and it worked just beautifully for me.

There were several others that, I mean, just ground my system to a halt. All kinds of errors coming up, files couldn't be found, I mean, basically it was a disaster. The one that I was hoping I was going to be able to like and recommend is about an 8MB download, and it's called the Virtual Sandbox Free Edition Version 1. It's from this Fortres Grand Corporation. And I've got a link on my show notes. The problem was, I mean, it offered lots of control, nice UI, but it just didn't work. I mean, it was like, okay. And what was really disturbing is I couldn't figure out how to get rid of it. It put nothing in the Add/Remove programs list. It had installed itself deeply in my machine with all kinds of DLLs and stuff, I mean, it was a serious install. And I thought, oh, no, now I'm stuck with this. Well, the good news is, running the setup program a second time, it saw that it was installed, and it offered to uninstall. But there was no documentation I could find anywhere that indicated that that's how you do that. It was, you know, it was just like, okay, I hope this works.

But anyway, the one I like is this Sandboxie. It is fully functional in its free version and offers additional features if you register, but you really need not register. The way it works, I would describe it as file-level sandboxing. When you set it up and, for example, you want to run – you want to sandbox IE or Firefox, you want to sandbox your browser, the first time you use the browser, basically it's monitoring everything the application does. if the browser, for example, opens its cache files in order to create a place for the browser's caching of images and other web objects to go, Sandboxie makes a shadow copy of that file, whatever – any files that your applications open, Sandboxie makes a shadow copy of that file. Now, if the file is open for read-only access, meaning that that file handle that the application has created can only be used for reading, Sandboxie is smart enough to see, oh, you know, this handle that's been created cannot be used for modifying the file. So it will – it's smart enough not to make a shadow copy. Unfortunately, many applications will open a file for full access, for read and write access, even if they're only going to be reading. Sandboxie has no choice but to make a copy of that file in its shadow land in order to then intercept any modifications the application makes.

So the point is that what I found in using it is that it's a little sluggish getting going because it's having to make copies of stuff that the application touches when it starts up. But once you've sort of got that little bit of sluggishness behind you – and I should mention that I'm using, you know, an older – I've got a dual 866MHz Pentium, is it 3 or 4? I can't remember now.

Leo: Well, anything would be sluggish on that.

Steve: Yeah, I mean, no, I've just – I've got so much installed here I'm reluctant to move to a different hardware platform. It's like, you know, that nightmare of setting up Windows from scratch. But anyway, so in my experience this thing really works. The guys who've developed it had been working on it for years. They've got interception for DCOM and RPC, ActiveX support, Comm object support. I'm very impressed with Sandboxie and with what it does for free. I would – and the other application that I like so much is, because it's lightweight, it installs without a reboot, it really would be possible for you to either have it on a dongle or to, like, maybe email yourself a copy to your Google Mail account so you'll always have it available if you were at a friend's house, you were at some other corporate location. My point is that this is a sandbox which installs smoothly and cleanly.

I need to mention it only does run under newer operating systems because it uses all the API functions of Windows 2000 and on. So 2000 and XP are the platforms where it can be used. None of these things that I have found will run back on '95, '98, and ME. They're just, you know, those are just too old now.

But the beauty of it being so lightweight, and the fact that it really does work, is if you wanted to use someone else's computer over which you didn't have long-term supervisory control, you could quickly install Sandboxie and do anything you wanted to within the sandbox with absolute confidence that, when you shut it down, and these files are deleted, no change has been left on the hard drive, no change has been left in the registry. And so from a standpoint of real easy lightweight privacy, I think it's a tremendous solution. I mean, I don't know of any reason why it wouldn't also protect you from a security standpoint. I just know that that's inherently dangerous. It's, you know...

Leo: Don't assume that it will.

Steve: Right, right. I mean, you know, they say it does. I respect these guys. But again, you know, there are principles of security which are inviolate. And one of them is, using software to block other software is inherently prone to error. It's just, you know, it's trying to filter what the soft- what malicious software wants to do. And if malicious software knew it was in a Sandboxie sandbox, or any of these other ones, for that matter, I have to imagine there was a way around it. But for the purpose of privacy protection, I really like this because it's lightweight, it's easily portable, it would allow you to use someone's computer with complete confidence that, once you're done, you have left absolutely no trace behind. Because there, on the privacy side, I'm sure their technology is robust.

Leo: So how much is Sandboxie?

Steve: It's free.

Leo: I like it.

Steve: Yeah. It is free. You get some additional features if you register. And I don't remember, I think it's maybe $19 or $20 to register. I mean, very inex- maybe even less than that. I don't remember.

Leo: Is it -xy or -xie?

Steve: Sandbox IE. Because it started its life as an IE sandbox.

Leo: Ah, I get it.

Steve: But that was back in Version 1. They're at version 2.65. They've even got a version that now runs on 64-bit Windows, or 64-bit platforms, rather, Windows on 64-bit platforms. I'm very impressed. I think, you know, given that all the ones that I was able to find, there were some that were, you know, that were much larger, much heavier weight. Some were frightening, I mean, in what they did to your system. My feeling is, if you want your system to have, you know, a industrial-strength sandbox, then VMware player with Ubuntu and Firefox.

Leo: Right, right.

Steve: We talked about, I mean, it's – you install it once. Then you've got a killer sandbox which is running the full operating system in containment. Or, you know, Microsoft's Virtual PC, similarly. So if it's on your own system, and you want real security and the same kind of sandboxing, because of course those also give you the same sort of privacy guarantees, in a cleaner fashion, inasmuch as it's a virtual machine, and this stuff really does work from VMware and Microsoft, that's what I would recommend. But if your interest, for example, is portable privacy, where you want to be able to use a machine temporarily and absolutely know, you know, you want to use Google Mail, which is web-based, and know that no images that you display, nothing that you've downloaded, no cookies that have been left behind, no login credentials, I mean – oh, these things also say that they, you know, are protection from keystroke loggers. Well, I mean, that just makes me roll my eyes. I mean...

Leo: Because they're obviously not. That's ridiculous.

Steve: You just – you can't. It's just, you know, you can't...

Leo: Because it could be hardware, there's so many ways to do this.

Steve: Exactly. So they're making claims that are not possible for them to guarantee. But from the privacy standpoint, I think these are – I think Sandboxie – I mean, and the other ones, too. But the other ones are frightening, I mean, they just – they hurt your system. It's like, whooo...

Leo: So this you could run off of a Flash, a thumb drive.

Steve: Well, you can't run it off of the thumb drive. But it's so small...

Leo: You could install from it, I get it, I get it.

Steve: Yes. You could literally store it on a 256-meg thumb drive, just to always have it with you. On the other hand, it is so small that if you had a server that you know – where you knew you could access it, and that's why I sort of like the idea, you know, either one of those...

Leo: Just download it real quick, yeah.

Steve: A free file site, or just send it as an attachment to yourself in Google, and then you could always access that...

Leo: Oh, good point, yeah.

Steve: ...in order to install. I mean, Google ends up sort of being like a free file server for that.

Leo: Can you save your Sandboxie virtual machine and move that around?

Steve: You cannot do that. So in that sense it's not the robust solution that VMware...

Leo: See, that would be great. Then if I had, you know, if I bookmarked things or I had cached files, I could take them with me instead of leaving them behind, which you have to do.

Steve: Yeah. And I have to say that I was trying to cover as many of these as I could as quickly as I could, so I did not go as deep into Sandboxie as certainly is possible to do. They've got a nice website, FAQs, you know, documentation, I mean, I just – I feel good about...

Leo: A lot of people have told me about this. So I think you're right, I think this is probably the kind of preferred choice.

Steve: Yeah, I feel good about it. Again, trust none of these...

Leo: Trust no one.

Steve: Trust none of these for security.

Leo: Right.

Steve: But I would say trust all of them for privacy. That part I think they've probably got nailed. And it's entirely feasible to filter Windows API in order to enforce privacy, in order to, like, keep shadow copies of files. And I would – if the idea of, like, this kind of portable, transportable, privacy shell is appealing to people, I really recommend our users check out Sandboxie. I mean, it's free. I would – and I would, really, I would support them. If you end up finding that you use it, you know, go back and give them your donation because this is the kind of thing we want to keep moving forward. And it's just a – I was very impressed. It did exactly what I was hoping I could find a sandbox to do.

Again, it says it will protect you against malware. And, you know, and why not? I mean, certainly, I mean, I guess my point is, it will as long as the malware isn't, like, really good about getting around sandboxing. And I have to imagine there is a way. So I'm not saying it's useless and isn't useful for protecting against malware. I'm really sure it is. But I would never say it's absolute protection because this technology can't give you absolute protection against malware. It can, and I believe it does, give you absolute protection or enforcement of privacy.

Leo: Excellent. Excellent. Very useful, very useful tool. And once again you've given us something to download, install, and use to protect ourselves. Thank you, Steve. I appreciate it.

Steve: I really think people are going to like it, Leo.

Leo: I want to, before we go, I just wanted to mention something. You're familiar with the Chuck Norris Facts website? You know, these are facts, kind of, you know, fun facts about Chuck Norris, like guns – you've seen this? Guns don't kill people, Chuck Norris kills people?

Steve: Okay, no, I...

Leo: Chuck Norris does not sleep, he waits? Anyway, people who love Chuck Norris, or even who don't, just love these. They're very funny. Well, security expert Bruce Schneier has been – he's the latest to be Chuck Norrised. There is a site now, Bruce Schneier Facts. And actually they're great for people who are security insiders because they'll get them, you know. Bruce Schneier knows Alice and Bob's shared secret. Or Whitfield Diffie and Martin Hellman use only their surnames out of fear of Bruce Schneier. You know, Bruce Schneier mounts side channel attacks through the front channel. Just if you're interested in security, Bruce Schneier's posted a link on his blog.

Steve: That's cool.

Leo: Bruce Schneier once factored a prime number. Bruce Schneier doesn't even trust Trent. Trent has to trust Bruce Schneier. Bruce Schneier can solve NP-Complete problems in NlogN time. They're just totally geeky, but I just thought I'd pass that along to you. I'm a little late on it because, well, because we're pretaping this. And when you hear this, it will be old news. But I thought I'd pass it along.

Hey, Steve, thank you. I will see you next week in Toronto.

Steve: Yup, yup.

Leo: And we won't record there, but we're going to – we'll party down, man. I think we're going to appear jointly on the techPhile podcast, as a matter of fact.

Steve: Cool, that'll be fun.

Leo: Frank has asked us to...

Steve: Do I bring my own mic?

Leo: No, I think Frank is...

Steve: Because, you know, my...

Leo: Frank is well...

Steve: I've got a big mic.

Leo: You have a big mic.

Steve: I've got a big one.

Leo: If you want the big mic – maybe you'd better bring the big mic. But I'm sure Frank will have his own little mic. Frank Linhares does the great techPhile.ca podcast, and we'll be appearing on that sometime in the near future.

Steve: I love my microphone.

Leo: I know you do, and I love mine. You and I have the same microphone.

Steve: We do.

Leo: Yes. Steve Gibson, have a great week, we'll see you next Thursday.

Steve: Okay, Leo.

Leo: Oh, wait a minute. I can't stop – wait a minute, stop.

Steve: Hey, where you going?

Leo: Where am I going? I've got to thank a lot of people. First of all, we don't do it enough, I really want to make sure I do it more often, thank AOL. Because really we wouldn't be able to do any of these podcasts. They provide us so much bandwidth, more than a terabyte a day now of downloads. And...

Steve: Wow. And you're adding two more podcasts, so...

Leo: And they're going to take them. I asked them, and they said we'd love to take them.

Steve: Fantastic.

Leo: Three more they're going to take. So that's really, you know, I have to say they're very generous there with their bandwidth. And I know if we had to pay for it, well, frankly, we wouldn't have so many podcasts. So thank you, AOL, for the great job you do.

We also want to thank our sponsor, who provides us with the funds to make this podcast possible, Astaro Corporation. Early adopters of Security Now!, and we're glad to be your adoptees. Astaro.com. They're the makes of the Astaro Security Gateway. If your small or medium business network needs superior protection from spam, viruses, and hackers, as well as a complete VPN, intrusion protection, content filtering, and an industrial-strength firewall, Astaro's the way to go. Open source, powerful, it really is fantastic and easy to use, and it comes in a single appliance. You just plug it in, and there it is. Contact Astaro, Astaro.com, or you can call them toll free, (877) 4AS-TARO to schedule a free trial of an Astaro Security Gateway appliance in your business. You'll see what Astaro can do for you. And of course non-business users, as always, could download the open source software version of ASG for home use at Astaro.com.

Steve Gibson's site is GRC.com. That's where you can go to get show notes for this at GRC.com/securitynow.htm, or get Elaine's transcripts and the 16KB versions. But while you're there, make sure you check out all of Steve's free security programs, like UnPlug n' Pray, the DCOMbobulator, and of course ShieldsUP!. How many ShieldsUP!? Is it over – it's over 40 million, it must be by now.

Steve: Oh, no, we're at 44.8.

Leo: 44.

Steve: Last time I saw. Almost 45 million.

Leo: 44 million systems scanned and protected. And if you have a hard drive, and I think almost everybody does, you might want to check out SpinRite. It's your hard drive's best friend, the ultimate disk recovery and maintenance utility, SpinRite, because Steve couldn't spell when he wrote it. But he knows how to spell it right now. I like it.

Steve: You realize that it was an 8.3 name.

Leo: 8.3, that's right, it had to be eight letters. Oh.

Steve: It had to be eight letters back in the old – back in, you know, 20 years ago, when SpinRite was first born.

Leo: Well, it still does, doesn't it? Because you boot SpinRite – when you download it, it's a very small file because he writes everything in Assembly. Just what is it, 30K? 60K? It's tiny.

Steve: It's gotten a little bigger over time because there's now a Windows side for creating a bootable CD and a bootable thumb drive and so forth.

Leo: I forgot about that, yeah. So it's – you run it, SpinRite 6, you run it in Windows, and it says, okay, what do you want to do? And you create a bootable drive or a thumb drive. I made a bootable CD. And there's DOS on that, there's FreeDOS or something on that, right?

Steve: Right, it is FreeDOS, and it is FAT32. So I could have used a longer filename now.

Leo: No, no.

Steve: But, you know, R-i-t-e, that's the name.

Leo: SpinRite.info, which is not in fact 8.3 compliant. Or you could go to GRC.com. Times have changed. We're not stuck with those three letters at the end anymore. Really a great tool. And if you want to read more testimonials from happy, happy customers, SpinRite.info is the place to go.

Steve, I think we have now wrapped up this episode of Security Now!.

Steve: Always good, Leo. And next week we are back to our Q&A that we punted on when we had that busy week in security a couple weeks ago.

Leo: Does that means we're going to have 24 questions?

Steve: We're going to – no. We'll do a dozen. It'll be Q&A No. 10, Episode 56.

Leo: I can't wait.

Steve: For September 7.