Sandboxie

Description: Steve and Leo return to take a much closer look at "Sandboxie," an extremely useful, powerful, and highly recommended Windows security tool they first mentioned two years ago. This time, after interviewing Sandboxie's creator, Ronen Tzur, Steve explains why he is totally hooked and why Leo is wishing it was available for his Macs.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-172.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-172-lq.mp3

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

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

This is Security Now! with Steve Gibson, Episode 172 for November 27, 2008: Sandboxie. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that focuses, aims a laser beam at your security and privacy online. Steve Gibson is here. He's the security wiz from GRC.com, the creator of ShieldsUP!, also the first antispyware software and many other great utilities, including, of course, SpinRite, which is the must-have hard drive maintenance and recovery utility. Hi, Steve.

Steve Gibson: Yo, Leo. Great to be with you this week.

Leo: What are we going to talk about this week?

Steve: Well, we're going to finally have the show that I've been wanting to have and mentioning and hinting about for many weeks. We're going to talk about Sandboxie. We talked about it about two years ago. When I fired up my email to contact Sandboxie's author, Ronen Tzur, I realized, I mean, I was surprised it had been almost exactly two years when he and I had had some interaction. We did a show about Sandboxie, sort of an overview, cursory look at it. And he felt the effect of us talking about it on Security

Now!, thanks to our audience, and sent both you and me some registrations for it. I was really pleased to see that they hadn't expired, even though Sandboxie has evolved a lot in two years. When I downloaded the current version, which is 3.32, and applied the registration information from two years ago, it worked because his registration is once per lifetime.

Leo: Yay.

Steve: That is, I guess, yours or his or somebody's - ours. And then you own it for life. And also you're able to use it on all of the machines that you own. So it's not restrictive in any way. And I've spent the last couple weeks looking at it more closely and then really focusing on it in preparation for wanting to really get to know it. And I've got to say, Leo, I mean, this is - it has turned into a really significant security tool. I mean, it is for people who are interested in security and want control over what things go bump in their Windows machines. I'm going to explain it in great detail. We've got Ronen on with an interview so people can get a sense for him and what his goals were and what the product does. And then I'm going to go into, in detail, how it works, to really explain it. I couldn't be more excited. I think this is just a - it is, I think, as important a tool as a personal firewall is for network communication.

Leo: Wow. High praise indeed from the master. So, Steve, before we talk to Ronen - I'm excited about talking to the author of Sandboxie - let's find out what's going on in the world of security.

Steve: Okay. First thing, Leo, we need to get you back using NoScript.

Leo: Oh, is it that bad?

Steve: No no no. It's that good. I wanted to acknowledge all of the annoyed listeners we have.

Leo: Including me.

Steve: Well, no.

Leo: It's still running on my system, by the way. I just turned off scripting.

Steve: They were annoyed that you and I gave up on it so easily.

Leo: Oh, okay.

Steve: And I found the secret. It's the secret everybody else knows except you and me.

Leo: Oh. What is that?

Steve: You turn off notifications.

Leo: How?

Steve: It's that annoying audio gwerp sound it makes and the popping up of the bar all the time that, you know, what I was doing - and I don't know, I mean, I know better than this because I was using a script management system in IE. It was my own. But the idea being if a site, I mean, virtually every site on the planet now wants to run scripts. But most of them don't need to. So you don't want to know when every single site you go to is trying to run scripts and then, like, oh, give it permission. The idea is you only give permission to the ones where you notice something doesn't seem to work, and you care. And so what I've been doing now for about the last two weeks, after I read through lots of people saying, okay, you guys are just dumb, I thought, okay, well, let's try this again. So I'm back using NoScript.

Leo: I'm looking in the options. You tell me what to do to make this work okay.

Steve: Okay. You just go into Options, go to the Notifications tab, and you'll see the first checkbox you turn off for popup notification when scripts are blocked. Then in the second region is audio, and turn that off.

Leo: Yeah, I have that turned off already.

Steve: Okay. And now then also go back to, I think it's on the first tab down at the bottom, is Globally Enable Scripts. And it says "Dangerous " or something after it.

Leo: Yeah, I had that checked. I'll uncheck that now.

Steve: Yes. And I've got to say, Leo, I'm now two weeks into this, and it's heaven again. It's only when I notice that something's not right. So you do, sort of in the back of your mind, you need to remember that you've got NoScript going, meaning that sites are not being allowed to script. But it's only the ones that matter where you then manually, you just click on the little "S" down there. And it'll show you if it's blocking something. And it makes it very simple to enable just the things that it has blocked. And so I'm really happy with it. I mean, I really think that the notifications should be off by default rather than on by default because...

Leo: Well, here's my concern is that I'm going to be surfing around now, and I'm not going to know that I'm not getting the full page. Almost everything I use uses JavaScript. And I disagree, it's nontrivial, it's not JavaScript I don't need. In many, many cases it's JavaScript I do need.

Steve: Well, okay. So we're going to talk today about Sandboxie, which is tremendous for preventing system modifications if you go to a malicious site that wants to hurt you. The problem is, it wasn't designed to be a privacy tool. So what you really want is you want scripting disabled unless you want it enabled. I mean, I've always been saying, even before I...

Leo: Right, opt in.

Steve: ...switched to Firefox and using NoScript that, you know, scripting is a real problem. And the reason you finally yourself added NoScript was you were listening to all this and thinking, wow. And, you know, one incident after incident after incident demonstrating what a problem scripting is. So...

Leo: Well, I've had it, and I've turned it on and off and on and off for years because I just get frustrated.

Steve: Well, so all I'm saying is I wanted to acknowledge all of our listeners who are successfully using it thanks to disabling notification and selectively enabling it. And I am now, too. So...

Leo: I'll tell you one thing it breaks.

Steve: ...you can join the party, Leo, or you can stay out there.

Leo: Well, first of all, I'm using a Mac, so I'm not too worried. But second of all - and if people would just use a Mac they would have fewer problems. But second of all, things like Amazon 1-Click is seen as a clickjack. It's like, okay. So many of these techniques that are potentially dangerous are also used by legitimate websites, even Amazon, that it becomes an issue. Now, in a perfect world, Amazon wouldn't use these techniques. Nobody would use these techniques. They wouldn't be so reliant on scripting. But we don't live in a perfect world. And I'm not - I'll use it. I'll give it a shot, and we'll talk next week about whether it's been successful. I'm sure you'll do the same.

Steve: See what you think.

Leo: Yeah. I've turned off notifications, so now I have no idea what's happening.

Steve: It works for me.

Leo: Well, yeah, because you don't know what you're missing.

Steve: And if I don't, I guess it's not very important.

Leo: Well, that may be. Maybe not. Maybe you're missing 90 percent of the functionality of a site. Part of my job is to look at sites and review them. If I'm missing functionality on a site because I have a NoScript running, it's not very functional for me.

Steve: Okay.

Leo: But I'll try it. I'm open-minded.

Steve: Give it a shot.

Leo: I'll give it a shot. All right.

Steve: Okay. Second bit of errata, before we get into the security news, is a couple days ago PayPal started acting differently for me. When I logged in, I gave it my email address; and then, because I've got my little PayPal football, I type in my password, and I know how I can then just add the next six, the six digits from the football to the end of my password in order to bypass the separate screen which says, oh, you're registered to use the security token, what is the current token value in the display window? Well, that didn't work.

Leo: Yeah. It's changing on mine, yeah.

Steve: So the reason it didn't work is PayPal suddenly became aware of my VeriSign VIP card.

Leo: Right. Well, that's why. It wants me to choose. All of a sudden I saw that. It wants me to choose.

Steve: Yes. And so we've been waiting for this. So I thought, oh, how cool, let me try that. Because now it's allowing me to choose any of the registered tokens, the security authentication tokens that are associated together. So I, for the first time ever, chose the VeriSign VIP, the credit card dealie that we've talked about using eInk. And the problem is, I pushed that little button so many times that it's gone off into the twilight zone.

Leo: It's out of synch, right, right, right.

Steve: I mean, PayPal has no, I mean, PayPal is using VeriSign's back end in order to provide these services. So what it did was it asked me three times, and I kept giving it the number. And it says, don't think you're right, try it again. I kept giving it the number. Then it took me to a screen where it asked me for two in a row. And that was the synchronization process we've talked about. And it worked perfectly. And I now have my VeriSign credit card authentication and my football. And this is the problem we talked

about where what if I didn't have my football with me because it's kind of bulky and a pain; whereas my little VeriSign credit card is in my wallet, and it's with me all the time. So now PayPal and eBay know about both.

Leo: I think that it was - I may be wrong, but I think I've been using my VeriSign card on PayPal for a while. So I think they knew about it. But I think what's happened is they're now explicitly asking you to choose which one you're using. Because I believe I have been using the PayPal - because the football's here, but I carry the VeriSign with me in my wallet. So when I'm at home I've been using the VeriSign. And I think I've been doing that for some time now.

Steve: I don't know how they would know which is which because...

Leo: They wouldn't.

Steve: Yeah, right.

Leo: Which is probably why they want you to do the dropdown now.

Steve: Exactly. Now you have to tell them explicitly which one that you're using.

Leo: They're both six digits; right? They're both the same length.

Steve: Yes. There are six digits. And one is time based, and one is sequential based. The card does not have a clock in it; so every time you push it, it generates the next cryptographic string in a sequence, which is why it needed a couple in order to synchronize itself. And actually the football does have - is time based and has the clock in it. So it's transparent to the user. Although if you were to push the football button a lot between uses, no one would care because it's time based. If you push the credit card a whole lot and, like, really go off into the future of sequences, then you would be forced to go through this resynchronizing process when you next tried to actually use it for authentication.

Leo: Yeah. It's a wonderful thing. And what's really nice is to be able to use either/or.

Steve: Yes.

Leo: Because, as I said, I have one here and one there.

Steve: Well, and for a long time you could register only one. So I wanted to let people know that PayPal is now up to speed, and the little credit card format tokens are functioning and work great.

Leo: Fantastic.

Steve: Security news. Bunch of interesting things. I wanted to mention that Apple Safari had a major update, both for Windows and Mac, to something past 3.2 - probably 3.3. I know that anything prior - I'm sorry, no. Now to 3.2 because anything prior to 3.2 had some critical remote code execution vulnerabilities that have been cured in 3.2. When I fired up my Mac this morning I did my routine check for updates. And it says, oh, we've got a new version of Safari for you. So I checked into what was going on. And that has been since we last talked. So Safari users will want to update themselves, whether they're on Windows or Mac.

MSNBC carried an interesting story about a report that Symantec released that I thought people would find interesting. It turns out that hackers that are breaking into eCommerce-enabled enterprises are no longer necessarily rummaging through the database, trying to install trojans, trying to just wreak whatever havoc they can. When they realize that they've cracked into an eCommerce-enabled enterprise, they stay on a down-low, and they actually commandeer the enterprise eCommerce system to use for verifying stolen credit cards. They've set themselves up as a third-party service. They charge $10 per card, Symantec reports, to get lists of stolen cards from other card-stealing people. And then they will quietly use the backdoor that they have established into an eCommerce system to verify the validity of the cards, which an eCommerce system is able to do. I wrote my own from scratch, and so I've used the API that is available for merchant services. And you're able to verify cards, that they're valid, without putting any charge on them. And so what hackers are now doing is they're breaking into enterprises that have merchant service backend access and commandeering it, using it quietly for their own purposes, and not wanting to be discovered because they want to keep being able to use this service.

Leo: That's really interesting.

Steve: Isn't that interesting? Yeah.

Leo: Well, they're so, you know, I hate to give them any credit at all. But when there's money involved they get pretty clever and wily.

Steve: Yeah. Microsoft announced that the November version, that is, this month's version of their MSRT, their Malicious Software Removal Tool, removed what Microsoft described as phony security software, which of course was malicious, from nearly one million Windows PCs in the first nine days of its deployment.

Leo: Wow.

Steve: And so a million machines had some sort of phony security software installed, and the MSRTs - I always kind of think that it's like, okay, well, is this really doing anything at all? You get a new one every month. And you may remember that two months ago, in September, it removed the Storm trojan from about 300,000 machines.

Leo: This was the - this time it was the Antivirus 2009; right?

Steve: Right.

Leo: That's a weird one. And I'm not surprised to hear there's a million of them out there. In fact, I bet you there's a lot more. I would get calls all the time on the radio show from somebody that says, hey, you know, Antivirus 2009 found some bugs on my system, and I installed it. It was like - and it has certain behaviors when it's broken. I mean, fortunately, as wily as these guys are, they're still crappy coders. So there are certain symptoms of this thing that when people call me up I go, oh, you got it, you got bit.

Steve: Well, and then there's a - we have a first to report. Dutch authorities, operating in the Netherlands, convicted a 19 year old for operating a bot network.

Leo: Yay.

Steve: Excuse me?

Leo: Yay, I'm glad they've convicted him

Steve: Yes. And here's what's very cool, and this has never been done before. After they arrested him, they used his command-and-control system of his botnet to inform all the infected end-users that their machines were infected with a powerful botnet. They redirected them to a different website which provided instructions for how to remove this thing from their machine.

Leo: That's interesting.

Steve: And that's something we have never seen before. It's questionable whether it's legal in the U.S. You know, there have been - everyone is always saying, hey, why can't we use these things to, like, disinfect themselves? And the idea is that, well, you're communicating with software technically in violation of the law by having any modification. Maybe it's the fact that they didn't remove it, but they just redirected people to a site which said, hey, you're here because the Dutch police have figured out that your machine was under control of a bad guy. Now it's under our control. We don't want it to be under anyone's control, even ours. So here's how you remove this.

The problem is it was a nasty bot. It was a rootkit bot. So it evaded scanners and antivirus things. It turns out it's difficult to remove, at least at this point. Maybe somebody would create an auto-remover. But you'd probably have to, like, boot a CD kind of thing in order to get it out of there. Because, I mean, it really set its hooks deep into the operating system. But I thought that was really interesting, that they said, okay, we're going to use the network to let people know they've been affected and let them do whatever they want to do with that information. But we're going to give them the

information.

Leo: I suspect that's a first. But it seems like since they're not modifying their machines, just notifying them, that would be okay.

Steve: Yeah. And then some Australian - you may have heard about this - some Austrian, rather, some Austrian firewall vendors found a kernel memory overwrite bug in Vista.

Leo: Yeah.

Steve: There's an API, an Application Programming Interface, there's essentially a subroutine that you're able to - that a program can call, called Create IP Forward Entry 2, which is it allows programs to modify the Internet routing table in Vista. You can also, if you use the ROUTE ADD command from the command line, and you give it a netmask larger than 32 bits, you will get a Blue Screen of Death. It will crash Vista. Now, what we know is that this is actually overwriting kernel memory, and that that's a dangerous thing to do because even though Vista has got a - and we've talked about this in detail - has some substantial anti-hacking technology, like the Address Space Layout Randomization, where DLLs are always being loaded in different locations so that it's not possible to do a highly successful jump to a certain location. Also it requires either full admin privileges in order for this to work at all, or you need to be a member of the Network Configuration Operator group, which most people aren't. So, and thanks to Vista, the way Vista operates, as we know, most users are now not operating as admin. They only elevate themselves to admin privileges during a time through the UAC that they actually need that facility. So...

Leo: You know, I saw that in the article, that, oh, this is nothing to worry about because nobody's running as admin. Is that really the case? I think most people do, in fact, run as admin, even on Vista. It's easy not to, and I tell everybody not to. I don't think the default install of Vista sets you up as a limited user; does it?

Steve: Yeah, it does.

Leo: Oh, it does.

Steve: Yeah. You are not - I believe in Vista, unlike XP, you are normally not an admin. And then you use the UAC in order to elevate yourself when you need to.

Leo: Oh, I hope that's true.

Steve: I think that's the case. And but I forgot to also mention, as far as we know, it is exclusively a local attack. Meaning that it's not something that can be done remotely. So code in your machine would have to be already in your machine in order to exploit this. As a consequence of all these mitigating factors, Microsoft just yawned a little bit and

said, eh, we'll fix this in the next service pack. This doesn't seem like something, you know, yes, we'd rather Vista didn't crash. You could experiment with it using the ROUTE ADD command and give it long netmasks. And if you're an admin user, you'll crash your Vista. Well, okay. Anyway, Microsoft is not...

Leo: Doesn't sound - mostly because it's a local attack. Somebody has to have physical access to use this one.

Steve: Precisely. And, well, and the point is - or code just has to be running. Well, if the code's running, then you've already been compromised.

Leo: Right. Too late anyway.

Steve: So it's like, yeah, okay. And of course I've got a - always try to find a fun and interesting SpinRite story to share. We've got Edward McCall wrote with just a simple subject: "SpinRite." And he said, "SpinRite just completed its analysis and low-level corrections of my hard drive. And I was delighted to find it corrected two Cyclic Redundancy Check, CRC, errors I had encountered. I was trying to copy two important files when I encountered CRC errors on those two files, and they refused to copy. SpinRite corrected those, and now I'm able to copy the files. Thank you, thank you, thank you. You have a wonderful product." All in caps. "Thank you again."

So we get so many people who write in where Windows won't boot, it's sort of an all-or-nothing sort of thing. Here was a different one because he had specific files on the system, not in the boot process, but on some important files that he absolutely needed. So he ran SpinRite on the whole drive, specifically targeting just those two, and it fixed it for him. So that was neat.

Leo: I just wish that there was some way - you know I use SpinRite all the time, and it just saved Dane's drive on his Vista machine.

Steve: Yup, I got email from...

Leo: Colleen.

Steve: From Colleen, yeah, she was asking me...

Leo: She's become, by the way, she's become the biggest SpinRite advocate you've ever met. I think when she first started working here she might have kind of heard of it. She certainly knew who you were. But now, man, she's like a SpinRite fanatic. Every drive that comes in the door, SpinRite, SpinRite. But I've got an iMac, and the newer iMacs are very hard to get to the hard drive, which is really frustrating because I know if I could SpinRite that drive, all would be well. But I just can't get to it, so I'm going to have to take it into the Apple store. And they'll probably replace the drive unnecessarily.

Steve: Ouch.

Leo: Yeah. Just, you know, if it's a SATA drive, it should be easy to access. The good news is it's not really debilitating because it's got Firewire 800, and I just bought an external drive, and it runs and boots off that just fine, so...

Steve: Ah, cool.

Leo: Yeah, it's not the end of the world.

Steve: Colleen sent mail because she wanted some clarification on SpinRite's S.M.A.R.T. monitoring screen to know whether it would be worth running SpinRite. And from what she said, I said, well, what you're looking at is data that was never meant to be seen. SpinRite services sort of the internal workings of the S.M.A.R.T. system. And so we don't even state what it is we're really showing. We just say, okay, this is stuff going on behind the scenes. We do also show the normal S.M.A.R.T. data. But the stuff behind the scenes is substantially more sensitive.

And so, as I explained to her, it's useful if you had a couple drives of the same make and model. You could compare the actual error rates that normally nothing shows you when you're running the drive under SpinRite to get the sense for which one is in better shape. Comparing similar makes and models would mean that the data that they were showing you was the same. Or if you looked at a drive at Day 1 and Day 120 and Day 240, it would also give you a sense for the rate at which the drive's condition was deteriorating, if it was. So comparing over time the same drive or the same make and model, it's very sensitive. Anyway, I said, so don't worry that those numbers seem high. Give SpinRite a shot. And I got email from her, I guess the next day, saying yippee, it just worked. So that was neat.

Leo: Yeah. She is a very, very happy SpinRite customer. You could add her to your list of testimonials. I mean, she is just a big fan of SpinRite. Steve, let's say hello to Ronen Tzur. He's calling us from Israel.

RONEN TZUR: Yes, that's right.

Leo: It's nice to meet you. I guess you and Steve have talked already.

RONEN: Yeah, yeah. And it's great to be on the show.

Steve: Yeah, we've been exchanging email for a couple weeks. And because I had originally hoped to have him a couple weeks ago.

Leo: Well, a couple of years, really. I mean, I think we've talked about Sandboxie for a while now.

Steve: Well, it comes up from time to time. And actually my motivation, we originally - we did an episode on it, sort of talked about it a couple years ago.

Leo: Right.

Steve: And it was just maybe a couple months ago that I noticed there was some dialogue in GRC's newsgroups. And when I jumped into those to ask people what they thought, they were like, wow, completely addicted to it. And so I thought, okay, we've been talking about Firefox; we've been talking about NoScript. It's time to revisit Sandboxie and really understand what this is about. So Ronen and I fired up a dialogue. I've spent a number of days, parts of a number of days getting my, I mean, really coming to understand what it is that Ronen created. And I'm really, really bullish about it. It's what I recognize now is that it's a - what he created is a universal security tool that encapsulates program behavior in Windows in a way that prevents things, malicious things, or inadvertent things, or privacy things, I mean, there's so many ways this can be used. So after we talk to Ronen, I'm going to explain in detail exactly how this functions, sort of from the user's Windows perspective. But I wanted to get a sense, for example, like what Ronen's original motivation was behind Sandboxie because this is, as I mentioned, four years in the making at this point.

Leo: How does it feel to have a program people are addicted to, Ronen?

RONEN: Well, it feels very good.

Leo: Yeah, I bet.

RONEN: Yeah, the addiction aspect of it is something that I recognize because once you start using it and you get the feeling that whatever you do with these protected programs inside the sandbox, they can't harm your computer, then without this tool you feel almost naked after a while.

Steve: And, I mean, a perfect example is just this morning I had received email that included a PowerPoint presentation. It was something someone had forwarded to me. He didn't create it. This was something roaming around the Internet. And it's like, oh, I mean, I'm nervous when I launch anything in email because you just don't know what it is. And when I initially clicked on it, I got a notice from Sandboxie saying powerpnt.exe does not have permission to run in the sandbox. Because I've got Eudora right now, my email client is in a Eudora sandbox, and Firefox is in its sandbox. And I had tightened it down so that only Eudora and Firefox could run in the Eudora sandbox.

But now Eudora wanted to launch the PowerPoint viewer. So I said, oh, yeah, well, I want to do that. So I'm evolving the settings a little bit of Sandboxie as I go. So I added powerpnt.exe to the list of permitted executables that could run in the Eudora sandbox, and then clicked on the link. Well, so now it worked this time. PowerPoint was allowed to run in the Eudora sandbox. And I knew that no matter what this thing did, it could not hurt me. Its view of the system - this is what I'll be talking about after we're through talking to Ronen - its view of the system is essentially Eudora sandbox-centric. And it is not allowed to make changes outside of the sandbox which are persistent. So, I mean, it's just - it's really spectacular.

Leo: Ronen, a lot of great software is written to scratch your own itch. Is this how Sandboxie started? Was it something you did for yourself?

RONEN: Yes, that's really the drive behind it because early on in 2004, when I just started developing the product, Sandboxie, at that time, or just sometime earlier than that, I was hit by spyware.

Leo: Oh.

RONEN: And I - yeah. So I couldn't get rid of it. And you know these traditional antispyware tools, they take some time to catch up with everything. So what I seemed to get at that point in time was something that traditional and at least the more common tools, they did not identify. So I ended up with a compromised system. And I didn't like it very much. So then...

Leo: How did you come up with the idea of a sandbox? Was that your first thought?

RONEN: Yeah. Pretty much, yeah. I thought if I could get Internet Explorer to run in isolated space, inside a sandbox, some kind of disposable space that I wouldn't care if it gets infected with spyware or anything else, that in the worst-case scenario I just throw it away and start over. So that was the initial thought.

Steve: So over - now, that was four years ago. At some point, then - were you originally thinking of making it a commercial product? Did you just do it sort of as something to solve your own problem? Where did it transition from, okay, now I've got Internet Explorer running in a contained environment, maybe this would be useful for other people?

RONEN: Well, I always had hopes for it as a commercial product. But I have to say that initially it was just a proof-of-concept product because I couldn't tell in advance that this is going to definitely work, and I'm going to be able to do it. For instance, there was a lot of talk on the Internet for a while how Internet Explorer is part of the operating system, and you can't really take it out of the operating system. And so, you know, this raises questions whether something like Sandboxie is even possible. So there was the initial idea. But then I had to work on it to see that the implementation actually works. But once I got that going, and I saw that it definitely was possible, and there are no real technical limitations to getting that done, then yes, I certainly was hoping to turn it into a commercial product at some point.

Steve: And so how has it evolved since then? From then to now?

RONEN: Well, it's been four years now.

Steve: Right.

RONEN: So there has been some evolution. Like I said, the first version was really a proof-of-concept code, and not something that you could really use. And that was in June of 2004, about six months or so after I started working on it. And at the time I felt that

the idea is really something new and something novel, and it's going to take some time to really become popular. So I thought even though this is not really a finished product I might as well start early getting the idea out. All right?

So over time I took that idea of isolating the browser, and I applied it to programs in general. And in fact that's what today most of Sandboxie is concerned with, that is, creating a sandbox that can run almost any program. And running the isolated Internet Explorer or the isolated Firefox is just a specific case of the general idea. Well, I think you touched on that earlier when you spoke about Sandboxie, Steve.

Steve: Well, yeah. And in fact, in talking with you, I didn't realize, for example, that you could use it, for example, to sandbox the setup or installation of a program. If you run an installer in the sandbox, then everything the installer does, including a sophisticated installation of an application, it's in the sandbox, so...

RONEN: That's right.

Steve: ...so and is not really making changes to your system. It thinks it is, but that's the trick and the beauty of what you've done is it thinks it's installing the software on your system. It's in fact making modification sort of deltas in the sandbox. And if you decide you don't want this, then you just delete the sandbox, and it's like this never happened.

RONEN: Exactly. That's the core idea of Sandboxie that, unlike other security tools, it doesn't try to stop the malicious program from working because then you have false positives when real software gets identified as malicious, or malicious software falls through the cracks. So Sandboxie takes another idea. It says, I'm going to let this program do whatever it wants. And if you don't like it, you can easily get rid of it with just a single mouse click.

Steve: Right.

Leo: Does it help now that you've got hypervisor and built-in hardware support for virtualization? Does that make it a little easier to do what Sandboxie does now?

RONEN: No, unfortunately, not at all because the hypervisor is designed and developed really for virtual machines, which is a similar but different technology from - similar perhaps from the concept, conceptually similar to what Sandboxie is trying to do, which is isolation. But from an implementation point of view virtual machines are different, so the hypervisor doesn't really make any difference.

Steve: One of the ways that we can think about this - and I was going to ask you how you would compare Sandboxie with a traditional virtual machine approach. In a virtual machine, what we're virtualizing is the chip. And so that's what is being extrapolated. And so to use a virtual machine, you're essentially virtualizing at the chip. You then have a sort of a clone of the chip, which you then install an operating system into, which you then install applications into. And that's sort of the traditional virtual machine approach. Here you're - it's sort of like you're virtualizing the API, that is, the operating system's Application Programming Interface, so that you're inserting yourself between the application and the OS, which solves, well, many problems that the traditional virtual machine has in that it's sort of an inherently much lighter weight solution. You're not having to commit a gig or half a gig of RAM to a virtual machine that is then taken away

from the host operating system. Nor do you need a second copy of an operating system and second copies of all your applications installed in that separate virtual machine.

RONEN: That's right. Also a virtual machine is inherently a different computer. It may be a simulated computer; but it is, at the end of the day, what you have, you have two computers now that you're managing, one computer as your primary applications, and then you dedicate the second computer, which just happens to be virtual and exist within your main computer. But in that second computer you have to install a second set of your applications and then keep them up to date. Now, what Sandboxie does is let you have roughly the same kind of isolation, let you have a logical virtual space that cannot - that is one way, that cannot touch back and make modifications to your primary system, but is isolated. But you don't have to dedicate any of the things you mentioned there. It's not a separate computer. You don't have to install a second operating system and a second set of applications. You don't have to dedicate...

Steve: Oh, and also a whole virtual networking system. When you use, for example, VMware, suddenly you've got virtual adapters that are, like, popping up all over the place.

RONEN: Right, right. It's really a complicated and advanced tool, and I don't think it was ever primarily designed to make life easier on the Internet, you know, to secure. It was, I think, it's - it may be used this way.

Leo: It's a side effect.

RONEN: Yeah. But Sandboxie, on the other hand, has been designed from the ground down - from the ground up. That's always been the core idea, to let you be safe on the Internet so that you can use your web-facing or Internet-facing programs in a secure way, where they are isolated from your system, and they can only do as much as you let them, and never more than that.

Steve: Well, then I guess the way it's evolved beyond that original motivation is that you realized that, because you've created this containment system which is robust, and you generalized it beyond Internet Explorer, for example, and even beyond email, to running anything in a sandbox so that any changes that anything you're running in the sandbox makes are transient. They are not permanent. They don't actually affect the system in any way outside of the sandbox.

RONEN: That's right. Most of what I had to do with Sandboxie, most of the challenges I had to address for Internet applications - for Internet Explorer, for Firefox, for Thunderbird, all kinds of Internet applications - because at the end of the day these are programs that are running on the operating system. So the bulk of the work that I had to do anyway to get just these programs to run. And I did some additional work to make Sandboxie in a sense generic. So the extra work that other programs may require, I also addressed that because I thought, why not, it's just the little extra mile, the one extra mile to go so Sandboxie can have much more applications than just using it for Internet security.

Steve: Yeah. I just - it's a tremendous solution. I'm really glad that I took the time to look at it again. I now have Firefox sandboxed, and Eudora sandboxed. I'm not a big IE user anymore. I was sort of noticing it's convenient that the name Sandboxie, that I guess began as SandboxIE, you know, it's convenient that the name is also sort of generic, and it doesn't really say IE that much because you've really moved it beyond

just being a solution for Internet Explorer.

RONEN: Yeah, I thought it was a cute name, so the IE aspect of the name. But it's really a minor thing because you have to suffix the Sandbox with something.

Leo: Sandboxer. Sandboxit. Sandboxie is good. I like it.

RONEN: Yeah, yeah.

Leo: So you do pronounce it "Sandboxie," not "Sandbox IE." We need to clarify that.

RONEN: Yes, Sandboxie. Yeah.

Steve: And where do you go from here?

RONEN: Well, I think the most challenges of Sandboxie have already been met. It does most of the things it set out to do. And but there is one particular aspect that I would like to address in a future version, which is even easier configurability than what Sandboxie has today. In some cases you have to configure Sandboxie to let the sandbox program talk to the software outside the sandbox. For example, a common case is PDF printing software, which has a part running outside the sandbox and a part running in the program that you are now running sandboxed. For example, part of it comes up in the sandboxed browser. And then when you click "Print," it tries to communicate with this part that runs outside the sandbox. Now, the strict isolation of Sandboxie initially prevents this communication. And the failing - I'm sorry. The printing fails. So what you would have to do is tell Sandboxie in this case to relax its isolation just a bit to make this printing possible. So right now what you have to do is you have to visit the Sandboxie website and get instructions for the particular piece of software that you are interested in, and then apply these instructions into the Sandboxie configuration. And then you have it the way you need it. But what I'd like to advance in that area is to make it somehow more accessible to offer these prepackaged configuration packages inside a program. All right?

Steve: Yeah, that sounds great. I should mention two things. First of all, it is, also from my own experience, I had exactly this issue, for example, with generating PDFs from inside Firefox. It is possible to use the tools that you have provided. There's like a restrictions monitor or resource monitor log where, as you're trying to do something, you log things that are blocked and allowed. And so it's possible for someone who understands the Sandboxie user interface, as I now do, to see what things were attempted and blocked. And that gives you literally the strings that you then add in the UI to permit those things that were blocked. And by doing a little bit of experimentation, even without, for example, going to the Sandboxie website, it's possible to develop your own little pinholes through the sandbox in order to get things to work that weren't before.

RONEN: Yeah, but I think it's also - it's worth mentioning that for people who want to just visit the website and get the pre-prepared list of instructions, that's also possible. You really don't have to...

Steve: Yes, I agree.

RONEN: ...go through monitoring.

Steve: Yes. And in fact I use, as it happens, Macro Express as my keystroke macro program. And I've got a bunch of macros that I use for helping me with email. And after I sandboxed Eudora they didn't work anymore. And I thought, I think I remember seeing something about that. And so I went to the Sandboxie site, looked under the compatibility, and there you had two strings that I needed, and step-by-step instructions, two strings that I needed to add to the Eudora sandbox specifically to permit Macro Express to work. And, bang, I did that, restarted Eudora, and now it works perfectly.

RONEN: Right. So it's already - it's fairly easy. But I want to make it even easier by offering these things directly in the program, without you having to visit the website. Because occasionally there are updates, and I just think it would be better.

Leo: Well, and that is what scares people off from security software in general. We've talked about that before, Steve, with NoScript. The harder it is to get it working, the more hassles and hitches you have, the more likely you are to turn it off eventually.

Steve: And so the idea, Ronen, would be that you would have, in the UI, like a list of known programs or activities that require some cross-sandbox operation. And you'd just be able to, like, turn on checkboxes, like if you - it would say Macro Express. And if you use Macro Express, you just turn the checkbox on, and that allows Macro Express to work across the sandbox.

RONEN: Yeah, yeah. Exactly like this.

Steve: Very nice. I should mention also to all of our listeners that I'm really pleased with your licensing policy because the license, once purchased, allows, first of all, that copy of Sandboxie to be used on all the machines owned by the licensee. So, for example, with your permission I get to use Sandboxie on a couple of laptops and various machines. I, at this point, I don't want to be without it, you know, truly. And the other thing is, it's a license for life. That is, you get it for Sandboxie and anything you do in the future is covered by that single license.

RONEN: Yes. Any future upgrades to Sandboxie are covered by that license. It's a lifetime registration license.

Leo: How has response been? Are you selling a lot of copies?

RONEN: I'm doing well. I'm doing all right.

Leo: Good. That's always nice.

RONEN: People like the software.

Leo: That's great.

RONEN: Yeah.

Leo: Is it your full-time job now, or are you still working in the business?

RONEN: No, it's my full-time job.

Leo: Fantastic.

RONEN: It's my full...

Leo: Isn't that a nice feeling, that you can write something like that - well, Steve, you know that feeling pretty well - and make a living on it is great.

RONEN: Yeah, it actually does feel great. And the positive feedback has been overwhelming, so that's also very nice.

Leo: Yeah, you're doing some...

Steve: Well, I'm really glad to be able to shine a bright light on this again, Ronen. You really deserve, well, I think you deserve success with this. And as a security-conscious person who's talking to a very security-conscious audience, I'm excited to be able to bring this back around, really explain it to our listeners, as I'm about to, and make it a tool that more people are going to be able to use.

RONEN: And I appreciate that very much. So thank you, Steve and Leo, for having me on the show.

Leo: Thank you, Ronen. Sandboxie. And it's - is it Sandboxie.com?

RONEN: Yeah.

Leo: It's easy to remember.

RONEN: It is.

Leo: Sandboxie.com. Thanks a lot, Ronen.

RONEN: Thank you.

Leo: Take care.

Steve: Bye bye.

RONEN: Bye.

Leo: That was Ronen Tzur from Israel, the creator of Sandboxie. Now we want to find out how Sandboxie works. This is where we get geeky. We're famous for this.

Steve: Yeah. And this is why I get excited, Leo, because this is, as I mentioned earlier, and also while we were talking to Ronen, I am truly, honestly excited; and, I mean, my world has changed. I'm now using Sandboxie for Firefox and for Eudora.

Leo: If you use Sandboxie, do you need to use NoScript?

Steve: Okay. Here's, I mean, that's a good question. The answer is yes because Sandboxie is not a privacy enforcer. It's a security enforcer.

Leo: Ah. Makes perfect sense.

Steve: So it prevents something from changing your system. But the program in the sandbox still has full access to everything in your system. So you can think of it sort of as like it makes it read-only, so it's able to read everything. So if you had malicious scripts running that were trying to steal information, they could still do so. So it's not a replacement for NoScript. And I am using NoScript in Firefox inside of Sandboxie.

Leo: All right, Steve. Sandboxie. How do it work?

Steve: Okay. The way I would like our listeners to think of this is as a very powerful, general purpose, security enhancement solution. As Ronen indicated, he initially started because he went to some site and clicked on some link, and his system was deeply infected by malware that he was unable to remove, some sort of spyware in Internet Explorer. So he thought, okay, this is bad. I don't know whether he reformatted his drive or what. But I think he indicated that he was really unable to satisfactorily remove this. So he thought, I'm going to see what I can do to prevent anything bad from happening to my machine.

So the way this works, what a sandbox is in Ronen's implementation, is imagine that you've got a program running in Windows, and it opens a file with read/write permission. It's possible, for example, in Windows to open a file for read-only access, where Windows itself then gives the program a handle to the file, which allows the program to read the

contents, but not to alter them. In that case, Sandboxie does nothing. It just says, okay, this program that is sandboxed is going to only read, so it's permitted. If, however, the program in the sandbox tries to open a file with read/write permission, so that it might modify the file, Sandboxie transparently copies the file into the sandbox. That is, and what the sandbox basically is, is just a set of folders on your hard drive. So it makes a copy of that file and subsequently redirects all access to that file rather than to the original copy out on the system. So anything [clearing throat]. Excuse me. Anything the program does, it does to the copy of the file, not to the original one.

Well, that concept is then extended to all aspects of the system, for example, to the registry, so that any keys that are edited or created in the registry - which of course is what malware does when it's trying to arrange to install as a rootkit to get itself started up surreptitiously - any changes that are made, anything that writes in any fashion is actually written in, like, it's intercepted and written into the sandbox. So from the view of the program it has successfully made these changes because, when the program tries to look at them, Sandboxie realizes that it's got its own local, sort of like a locally cached copy. You can really think of it like a cache, where it's getting the contents originally from the operating system, then caching it in the sandbox. And modifications are contained in the sandbox, so that the application thinks it's succeeded, and it's chuckling to itself, when in fact it didn't do anything to the system.

So what Sandboxie does, another way to think of it is it sort of creates a fork of the state of your system, where the program that's making modifications, it sort of forks off of the real condition of your system and begins to see its own local private view, which only it sees. Nothing outside of the sandbox is aware of the sandbox or has access to it. It's just the normal system. But the thing running in the sandbox sees a different view. It sees its own private view. So what this means is that you've isolated any actions which are taken by this program, and nothing outside is actually modified, even though that's transparent to the program running in the sandbox.

So in practical terms, the way this works is, for example, with Firefox or Opera or IE, whatever web browser - and generally what's so cool about this, even though this started out to be, as Ronen indicated, a program to sandbox Internet Explorer because that's where his first problem was, and how many times have we said on this show that the web browser is now the biggest security problem in the system because it's the way - it's what you use to go out onto the Internet, where unfortunately not everybody is a good guy. And as you click on things and go to websites, you're oftentimes, as we talk about with scripting, you're running code, you're running ActiveX controls which are being downloaded, whatever you're doing, I mean, that's the big danger.

But Sandboxie is a universal solution for any executable that you want to run in your system. And when we were talking to Ronen I gave the example of using it to isolate the installation of programs. If you run a setup program or an installer in a sandbox, it thinks it's installing the program in your system. But in fact all the changes it makes are constrained within the sandbox. And he even has the ability - it modifies, for example, the Start menu. It thinks it's added things to the Start menu. But if you look at your Start menu, they're not there because your Start menu is not in the sandbox. But Ronen in his popup menu has a sort of Start menu clone which merges the updated Start menu in the sandbox with the external Start menu that hasn't been merged. So there you can see this thing you've just installed is in the internal Start menu, and so you can run it. So this allows you, for example, to do what many of us do with a virtual machine. In fact, remember that I installed Chrome in a VM initially. And you were saying, wow, you must not really...

Leo: You don't trust anything.

Steve: You don't trust anything. Except I was doing it for containment because I know, and old-time, old-school Windows users know, that when you install things sometimes you can't really get rid of them completely. Well, I was installing Chrome in a VM. Now, today, knowing Sandboxie as I do, I would install it, I would run the Chrome installer in a sandbox. And in fact that's trivial to do because Sandboxie adds to the right-click context menu "Run Sandboxed."

Leo: Wow.

Steve: So literally...

Leo: That's so great.

Steve: ...any program on your system, you can right-click on it. And rather than, like, double-clicking on it if it was sitting on your desktop, like you just downloaded some installer, rather than double-clicking on it you could right-click on it and just select from the context menu "Run Sandboxed." And it fires it up in this containment environment so that all the changes it makes are constrained. Now...

Leo: Is there any limitation to this, I mean, that you've found? Are there any kind of places where that doesn't work very well?

Steve: Well, when you think about it...

Leo: Programs assume that they're accessing the whole system.

Steve: Right. And there's no place that I have found where there is a problem you can't overcome. One of the things that users will see, and which I have seen, is this has all the feeling of being a very mature product. This is not a 1.0 or a .99. It's at 3.32. It's been around long enough, and it's had enough of a user base, that it's matured so that the various things that might cause problems have solutions.

Now, one of the things we all do is we download files using our web browser. So here's a problem. Because we've downloaded a file, but inherently we don't want that to stay in the sandbox. We want to, if we trust the file, if we believe it - well, for one thing, when we download it and execute it, you might want to execute it in the sandbox, in the browser's sandbox, if we're not sure we can trust it. But often, like I'm downloading an MP3 from you, Leo, well, it's an MP3, it's from you, I want it out of the sandbox. So there is a - you have the ability to define special directories where anything that is written to that directory you're notified about. A popup comes up and says, oh, you've just written something to your desktop from your browser. Do you want to recover this? He calls it "recovery." Do you want to recover this file from the sandbox? And so with just a click you can say yes. And then it appears on your desktop. Until then it was contained within

the sandbox. So he has provided ways which are convenient to allow the migration of things from inside the sandbox to outside. And they're very convenient.

One of the other things that is very cool, for somebody who's interested in, like, doesn't trust some software or wants to maybe sort of do some forensics, you know we've talked for example about running Filemon or Regmon and then, like, scanning through it to try to figure out what did this program do? Thanks to the sandboxing technology, which is file based, after you run something, like I've been running Eudora in its own sandbox for a while now, I can - and Ronen provides through the Sandboxie user interface the ability to explore all the things that have been going on in the sandbox. It's a tree-structured file system. So you can just - you can open up your C: and your Windows and your system and, like, explore through and see what things have been modified because any modifications that would normally be made to your system are intercepted, and they create the modified files in the sandbox. So it allows you to literally do like a forensic exploration of every single thing that has happened in the sandbox during its lifetime.

There's also the notion of persistent and transient sandboxes. That is one of the options, and it's the one I've got set up for Firefox, is when I close Firefox the sandbox contents are deleted. So any accumulation of debris from surfing, it might be malicious, it might not, but any accumulation is just completely flushed every time I close the last instance of Firefox running in the Firefox sandbox. But you may have an application where you want more persistence. For example, imagine that you wanted to create, like, a private, a very secure surfing privacy area. Well, you're able to define and create different sandboxes. You name them. And so I've got a Eudora sandbox, a Firefox sandbox. You could also create, like, a private surfing sandbox. And Sandboxie allows you to define where the sandbox lives, that is, on what drive it lives.

So one of the examples, and this is being done by Sandboxie users, is they'll use TrueCrypt to create an encrypted area of the drive, of your drive, as a drive letter. You can then aim Sandboxie at that drive letter. And that means that everything done in the sandbox is, thanks to TrueCrypt, is encrypted as it's being written and decrypted as it's being read. So at no time is anything permanent being done on your hard drive that could ever be recovered by anyone doing any kind of forensic analysis. So it allows you to create, thanks to TrueCrypt, not only do you have this notion of transience, but you could set up the sandbox to be permanent. And in fact there's an option to prevent its inadvertent deletion. So you can turn that on, and you won't be able to delete it unless you go through a bunch of, yes, I'm really sure sort of jumping through hoops involving turning that off and then confirming that you want to and so forth. And that would allow you to create an environment where any programs running in this privacy enforcement sandbox, they don't - they're not special programs. They don't know - they don't use encryption. They don't need to use encryption. It might just be like a photo album, for example, where for whatever reason you want photos through your photo album that absolutely only you have access to. And since you've got TrueCrypt, you'd have to, in order to get access to that sandbox, which is running in a TrueCrypt drive, you'd have to provide the TrueCrypt password in order for that sandbox to have any opportunity to access that drive. And then from then, any programs you run within that privacy enforcement sandbox get all the benefits of absolute privacy enforcement and the use of TrueCrypt, even if they're not aware that they've got this going on. It's completely transparent for them.

Leo: Wow.

Steve: I mean, it is a universal, very powerful security enforcement appliance. The one

thing that he talked about that I want to reiterate is I will say that it takes some time to sort of bring yourself up to speed. When I was first using it, and I created a PDF, I got an error from Adobe Acrobat. complaining that it was unable to create the PDF. And I thought, okay, what do I do about this? So I did some looking around. And he provides all the tools for basically analyzing what the problem was. There's a log that Sandboxie - that you're able to open when you do something that goes ouch. It'll appear in the log and show you what it was that was blocked. And then Ronen provides, through the user interface, the ability to, for example, give access down a certain path on the hard drive to a file, or to allow interprocess communication, an IPC event. And so the tools are there, although they're a little tricky to use yet, which is why - what we heard him say was that he wants to make that process more automatic so that you would just check, oh I need Adobe PDF. You would just turn on a checkbox in a sort of like a compatibility list.

But what's cool is we're not dependent upon Ronen to provide that in the future. All of that exists now, though it does require a little bit of experimentation in order to understand basically how the system works. So I would say that it's not trivial to use, but it is something that you could set up for someone. And after you made sure that they could print and that they could do the things they want to, it is sort of a set-it-and-forget-it. The way Sandboxie functions, one question I forgot to ask him, but I already knew the answer to it, is could this be set up in a portable mode.

Leo: Oh, yeah.

Steve: Could you have Sandboxie on a USB stick, for example?

Leo: Wouldn't that be cool, yeah.

Steve: Unfortunately, in order to pull this off, Sandboxie is a driver that requires a reboot of your system. It's a kernel driver...

Leo: That makes sense.

Steve: Yes. And there's a service running in the background. Because, for example, one of the things you can do is you're able to say, and I have, don't ever allow Firefox.exe to run outside...

Leo: Without Sandboxie.

Steve: Yes, without sandboxing.

Leo: That makes sense. You'd had to - because you have to catch calls. You'd have to be running as a service or a driver in the background.

Steve: Yes. It is a low-level piece of technology. I mean, basically he's sunk his hooks into the OS. It is Vista compatible. I'm running it under XP. He sunk his hooks into the

OS so that, for example, he's catching an executable starting. I don't have to, like, have some funky shortcut or something to, like, get Firefox to run in a sandbox. When the system - when Sandboxie, which is down in the kernel, sees Firefox.exe starting up, knows to insert it into the Firefox sandbox, or to create one because now I've got Firefox sandbox set up to be transient so that it just deletes it after my last use of it closes.

Leo: Ironically, he's using a lot of the techniques that bad guys use.

Steve: Oh, yeah, I mean, it is...

Leo: We should have asked him. I didn't think of it until now. But we should have asked him, where did you learn to do all this?

Steve: And one of the other cool things, it sort of shows his roots, all of the use of that configuration UI, it all boils down to a single sandbox.ini, a sandbox.ini config file. So you're able to take your mature configuration and easily copy it to another system.

Leo: That's nice.

Steve: And thanks to his licensing, a license is a lifetime license for all of the past and future of Sandboxie. And you're able to run it on all the machines that you own. And I mentioned also earlier, I just want to say it again because it was a neat - I'm still having to get used to the fact that I've got this level of security now because last night someone sent me a PowerPoint that they were passing around the Internet. It was some Einstein quotes PowerPoint. And it's like, okay. And it's funny because I had configured my Eudora - I'm using the Eudora email client. I configured my Eudora sandbox to only allow it and Firefox.exe. And I didn't even know about Firefox, but I clicked on a link in email, and I got a notice saying, whoops, Firefox.exe has no permission to run in the sandbox. And I thought, oh, that's right. So I went over to the UI, and I added Firefox.exe. Then when I clicked on a link, Firefox ran just fine because it was running in the Eudora sandbox.

So earlier this morning, when I clicked on this Einstein.ppt, this PowerPoint file, I got a little popup that said powerpnt.exe does not have permission to run in the sandbox. It's like, oh, that's right. So I added that to it. So there is some, just very much like with a software firewall, where over time you need to give it the permissions that you choose to give it in order for it to understand who you are and how you want it to operate, there's that same sort of evolution of the settings of your use of Sandboxie. But now, as a consequence, I did click on the link, and PowerPoint opened.

Now, PowerPoint was opening in the Eudora sandbox. Meaning that no matter what this did, if there was an exploit, like even an unknown exploit - and this is the point of where this is the way I want to operate, rather than relying on an AV program to, like, scan the Einstein.ppt file on the way in to see if it contains something malicious. The problem, of course, is that necessarily AV patterns lag behind. So what I've got now is better than that because, if there was an unknown vulnerability in powerpnt.exe, and I had just received an exploitive PowerPoint file, and it tried to do something to my system, I'm protected because it's going to be contained within the Eudora sandbox and never get out.

Leo: Isn't that sweet.

Steve: I mean, it is a tremendous solution, Leo, for our listeners who are willing to spend some time. He takes you through a nice tutorial when you first start using it. It's a little bit rough at the beginning, so it's not like just completely smooth sailing. And so there's some learning curve associated with it. But this is, I mean, it is the most significant security tool I've run across in a long time. The alternative, as we talked about, for example, is a heavyweight virtual machine. And, I mean, I've got VMware installed here. I'm not running my browser in it because it's just too clunky. It's too heavyweight. I want to run my browser in my operating system where I can easily move files in and out of it, and I can click on links and do things and not have that in my way. And a virtual machine, as we've talked about before, requires a commitment of a block of your system's RAM while it's running because it's unable to share RAM with the main system. And you need a second copy of your OS, and you need your various applications all installed in there, too. And then you've got to keep them all up to date. The beauty of Sandboxie is that it is running with your existing OS, no heavyweight virtual machine, no precommitment of RAM while you're using it. It's just it's not in your way, but it is preventing things from getting out of containment. It's just a tremendous security solution.

Leo: Of course writing it requires a great understanding of the deep guts of Windows. And it's not...

Steve: Oh, yeah.

Leo: It would be nontrivial to port it to other operating systems. But boy, I think it'd be great to have something like that on other operating systems, Mac and Linux as well. It'd just be really useful. But maybe now that the idea is out there, somebody will do it.

Steve: Well, we've got it for Windows, which is where I am and where a lot of our listeners are. And again, it is - you can download it. You can use it for free for 30 days. After that there's some sort of - after 30 days, if you continue using it without registering it, there's some sort of an annoying popup. I've never seen that. There are a couple additional features which you get with registration. He's got some, I mean, as much as I've just talked about this, there's a bunch of stuff I haven't talked about, like you're able to turn on automatic CD and DVD protection, so that anything that runs from your CD or DVD drive is automatically sandboxed. And so there's like a whole bunch of other cool little features which are the kind of things you see in a truly mature product where people have said, hey, what about this, and Ronen says, oh, I like that idea. And there it is a week later.

Leo: I'm so glad he can do this full-time because that's what it takes.

Steve: Yes.

Leo: That's really great. Really great.

Steve: Yes. I mean, I have a new security tool which I recommend without hesitation, with the only caveat that it's going to take a little time to kind of get comfortable with it, to understand how it's changed your system. But once you've done that, once you've made that investment, what you've got is something that truly protects you in a way that nothing else we have, except creating an entirely separate virtual machine and doing everything in a virtual machine, this is a much easier, not in-your-face, light way to get that level of protection. And I can't recommend it highly enough.

Leo: Yeah. Well, thanks for highlighting it. It's funny, we've been talking about it for a couple of years at least.

Steve: Now we really know what it is.

Leo: And now we know the details, and we got to meet Ronen, which is really cool. As usual, Steve, you do a great job of bringing this all to light and sharing it with the world. And I hope many, many, many, many thousands more people will be using Sandboxie. Of course...

Steve: And I hope so for their sake, not even for Ronen's sake. I mean, this thing is an honest-to-god serious computer security tool.

Leo: And so far I've been using NoScript for another half hour, surfing around. You're right. Turning off the notifications seems to make it quite a bit more usable.

Steve: Oh, it ought to be turned off by default, yes.

Leo: It's a significant difference.

Steve: It's really, well, we'll see how you last, Leo, with it, because I'm...

Leo: I can't even tell it's running now.

Steve: Yeah, exactly.

Leo: Yeah, yeah. No, that was definitely the thing to do. And I can live with the 1-Click kind of giving me a warning, what the heck.

All right, Steve. We are going to wrap this thing up. We do invite everybody to check out Steve's website, GRC.com, if you would like to know more about the subject.

We've got of course, show notes, transcriptions of every show, 16KB versions for those of you who are bandwidth impaired. And there's lots of other great stuff, including Steve's great security forums. That's where to go, by the way, to pose your question. If you want to ask a question for our next episode, GRC.com/feedback is the place to go. Hi, Steve. Are you waving at me? Hi, Steve. And, let's see. Oh, don't forget, SpinRite's there, too, as well as many great free programs Steve makes available.

Steve: Colleen's favorite new utility.

Leo: Yeah, no kidding. I'm going to have her start doing the ads for SpinRite. She just loves it. And that's about it.

Steve: Talk to you next week.

Leo: Don't forget, you can catch this show and all of our programs free on iTunes. Just search for TWiT in the iTunes store. They're all free, and there's lots of great stuff, including Security Now!. We'll talk to you next week, Steve. Thanks very much.

Steve: Okay, Leo. Thanks.