SecurAble

Description: This week Steve takes the wraps off his forthcoming security freeware utility: SecurAble. Although he's still working to get it finished, tested, and ready for initial release, Steve describes what SecurAble will do and some of the unexpected hurdles he's encountered with the application and with details of Windows operation along the way.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-071.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-071-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 71 for December 21, 2006: SecurAble.

Security Now! is brought to you by Astaro, makers of the Astaro Security Gateway, on the web at www.astaro.com. And by Dell. For this week's specials, visit TWiT.tv/dell.

Time to talk about security. And even though the holidays are fast approaching, Steve Gibson insists that he's going to catch TWiT. And so when TWiT's off, Steve goes crazy and says, "Let's do another one." So we'll do two today.

Steve Gibson: I think we're only 10 weeks behind; right? Did you just do #80?

Leo: Just did #80.

Steve: Yeah. In fact, I participated in #80.

Leo: That's right. It was very nice to have you.

Steve: It was fun.

Leo: Yeah. You're great to have on TWiT. And I think you and Dvorak are just fun to have

together, so thank you.

Steve: And we do have, you know, we have a different approach because I generally – I infatuate with things. I love technology. I'm a little bit of a sucker that way. And John is, I mean, exact...

Leo: He hates everything.

Steve: Exactly. So I think he and I do have a sort of nice counterpoint against each other.

Leo: I agree, yeah, it's very good. It was a fun show. And if you haven't heard TWiT 80, please download that. We are going to take a couple of weeks off for Windows Weekly because I'm going to Mexico pretty much as soon as we finish taping this. And then I'll be back – Daily Giz Wiz is going to shut down for a couple of weeks after this week and so forth. We're going to take a little time off. I call it "podcast catch-up weeks," so you can catch up with the ones you've missed. And there are a lot of old issues you can go back and look through.

Steve: That's very good, Leo.

Leo: I doubt very many people have heard every Security Now!, for instance.

Steve: It's also interesting that I was thinking how much Vista is in every, well, is in our minds. I mean, you and I are talking about it. We're using it. Certainly you and Paul are talking about it with Windows Weekly. Yet it's got to be a little frustrating for so many of our listeners who are hearing about something that they don't yet have and cannot yet get. So on the other hand, you know, from a security standpoint we want to lay down the foundation of what I think is going to be a very important enhancement to the Windows workstation desktop platform with Vista.

Leo: I completely agree. So what is our topic for the day today?

Steve: Well, today I want to talk about something that I've alluded to over the last few weeks, that I've been now working on for many weeks, which has turned into, as many of my things do, a much larger project than I imagined. It's this piece of freeware that I will be releasing. It's not yet finished. But I want to talk about what I'm learning along the way because I think people will really find it interesting and because what I'm learning is significant, even though I'm not yet really sure what it means. So I'll explain all that.

Leo: Okay. That's coming up. But first some errata, I think, yeah. Or follow-ups.

Steve: Yeah. Well, of course, last week, right, last week we talked about TOR, the Onion Router network, which generated a ton of feedback. And you mentioned that you even had some callers on your weekend show.

Leo: On the radio show, yeah.

Steve: Now, most of the concern – well, there was some issue about the TOR technology. And in fact we're going to cover those questions in next week's Q&A because we have a Q&A coming up next week. So those I'm going to hold off on. But the interesting ones, I just wanted to acknowledge them, were more the issue of sort of Episode #69 where we were talking about the policy, the social implications of anonymity. For example – and I'm just going to read one person's note because it's sort of representative of this sort of issue that people had.

He says: "Free speech does NOT require anonymity! In the United States you can say just about whatever you want without anonymity, and you won't get dragged away by the police. There are only a few areas that aren't protected, such as criminally threatening somebody or inciting a riot, which aren't protected because these types of speech violate other citizens' civil rights in the process. Conversely, a person speaking under anonymity in an oppressive country does not have the freedom of speech. He is simply avoiding persecution from his government, but he never truly had that freedom. I almost stopped listening to Episode 69 and 70."

I think he meant he almost stopped listening to Security Now! forever. He says: "After hearing your argument and only later realized that you were parroting the bad logic from the Freenet Project website. They've written software with the sole purpose of violating copyright laws and trying to justify it by boiling down a series of bad logical arguments to 'You either choose between freedom of speech or copyright laws.' Given the choice, who in their right mind would choose the latter? I guess that's what they're banking on if they ever go to trial over this. Freedom of speech does not require anonymity. Lack of accountability for your speech or actions requires anonymity."

So, you know, I got a bunch of postings like that. And I wanted to acknowledge that I understand that argument. And what we were talking about, because we were talking about in Episode 70 this whole notion of sort of infinite techno anonymity, which is what the TOR network brings us, I wanted to comment, while I completely understand what that guy is saying, and I hope I have buffered some of the absolutism of that in Episode 69 when we were talking about the fact that, for example, I have no need for anonymity. As you know, Leo, I've gotten involved, for example, in my local homeowners association, and I've sent a number of letters out to all of the people in the association, 309 homeowners here, signing my name and putting my address on it. Other people, because it's gotten sort of political, have been leaving anonymous flyers on people's doorsteps. And my fellow homeowners have come up to me and really thanked me for my involvement and noted specifically how much they appreciated I put my name and address on my communication. I was taking responsibility for what I was saying, which I think makes it much more effective.

But at the same time we've got people saying, wait a minute, you don't need to be anonymous to have free speech, except there is that counterargument, which is, okay, there are people maybe who are shy, or who don't feel that they have the freedom to say what they want for fear of their neighbors being upset with them. Even that, not necessarily your government, we're not talking about being dragged away by the police, but for whatever reason they don't want to set their name to their feelings, yet they still want their feelings to be known. So there are two sides to it. I know that we sort of ventured into political territory with Episode 69...

Leo: And obviously upset that guy. But believe me, all you have to do is say, in an election, one of the fundamental tenets of elections is the anonymity of your vote.

Steve: It's a very good point, Leo.

Leo: And that's all you need to say. It's something that's been built into our Constitution. It's something that's part of our society. Anonymity is very important. Otherwise, if you don't have secret ballots – yeah. And I hate it when people say, "But it's safe here now." Well, true. But it's not necessarily always that way. You've got to be vigilant of freedom. You don't just assume we've got free speech, we don't have to do anything about it.

Steve: In fact, it's safe here now because we have had that freedom of an anonymous ballot, where people were able to vote without any concern that...

Leo: And it hasn't always been that way. There are many times in our own national history where there weren't secret ballots, and it was an issue, things like the poll taxes. We've got to fight for these things. They don't come automatically. So for somebody to say, well, it's safe to be, you know, you could say anything you want here in this country right now. That may be true right now. Doesn't mean you shouldn't continually monitor this stuff. And when technology provides you ways to do it, you should do it.

Now, I'm uncomfortable with Freenet, as well. I'm not running Freenet for a lot of reasons. But I don't think that there's a problem with the TOR router. I think that this is the kind of thing we've got to have so that we can preserve our own freedoms. Period. And so if the guy doesn't like it, you don't have to listen to Security Now!. That's fine. Go listen to Rush Limbaugh. I'm perfectly happy with that.

Steve: Well, and I also want to acknowledge that there are levels and shades of gray, and that...

Leo: Absolutely. It's not black and white, yeah.

Steve: Right. I can really appreciate that there are people, for example, very active people in GRC's newsgroups. I have no idea who they are. I don't care who they are. They've established an identity. They are completely anonymous. And that's just fine because over time I've learned who these people are based on their handles. We have provided the ability to prevent handles from being spoofed. GRC prevents that from happening, as I mentioned in Episode 69. So it's like they're entirely effective. And if they would rather not use their name, I'd rather hear from good people who take the option that the Internet provides in all kinds of ways of being unknown.

Leo: Let me ask you the question that our caller on the radio show asked because I thought it was kind of an interesting question. He listened to the TOR router show and clearly understood it. And he said, but one thing bothers me is, let's say a governmental agency wanted to infiltrate the TOR router, put lots of TOR routers online, and then hoped to, at least from time to time, make their router the last router on the TOR routing. Wouldn't they be able to see your message in the clear?

Steve: Yes, that's absolutely the case. They would be able to see the final decryption of your TCP traffic as it was leaving and returning from that last router in the anonymizing network. So it's not providing encryption. Now, of course you could create an HTTPS, that is, an SSL end-to-end encryption, so you're encrypting everything yourself so that – between you and the remote site. So that would create an encrypted tunnel that was then being anonymized by the TOR network. But there has been – people have noticed a couple things about TOR, and we'll be covering it in some more detail next week. For example, if you had a malicious TOR router as,

as you said, the last router in the onion chain. Then there are different things it could do. It could even use cookies and JavaScript and inject stuff back into your browser, which could potentially then, for example, if it was able to inject JavaScript in pages you were retrieving, JavaScript could run in your browser and reveal your IP back through that connection. So there are potentially bad things that could be done. But the technology of encapsulating encryption, as we discussed last week, in this onion, it's very effective for anonymity. A high percentage of routers could be compromised, and still anonymity would be protected.

Leo: So the thing to understand is that TOR is not about encryption. It's about protecting anonymity. And the way it does – and people have to understand that, even if you use encryption, very sophisticated traffic analysis programs exist that can deduce a lot about what's going on just by watching who's talking to whom when. So that's what the TOR does. It doesn't encrypt your data. It hides who's sending what to whom.

Steve: Correct. And in fact one of the things that – there's a really neat page on the TOR site which talks about the problems that they acknowledge the network still has. We talked about how, for example, all the packets being exchanged are padded out to a fixed length so that you cannot use packet length analysis in seeing traffic coming in and going out of a single TOR node in order to associate them. But because TOR is a relatively low-latency network, that is, they want the thing to still be functional and useful, the routers do not introduce random delays, that is, in how long they hold the packet before routing it. If routers sucked in packets and waited random lengths of time before sending it on, then you would not be able to use timing analysis with any – essentially it would really render timing analysis fruitless. On the other hand, it would introduced tremendous end-to-end delays as the traffic moves through the network. And so that would be a different problem because it would then drop the usability of the TOR network. And it turns out that their feeling is, okay, we need some tradeoffs here. And keeping the network fast enough to use, rather than it just being useless because it's too slow, is better than not having people use it at all.

Leo: Right. All right, any other past items we want to cover?

Steve: I think we're ready to go on to talk about what I've been doing for the last month, essentially.

Leo: Before we do that, I just want to acknowledge some of our fine sponsors. I think this may be the last week we'll hear a Dell ad. Maybe there'll be one more. They've been so good and so supportive to us. It's been a great three-month run, and we do hope they'll come back. And I also want to thank all the folks who bought Dell equipment through our links on the TWiT page because we got credit for each and every one of them.

All you've got to do is go to TWiT.tv/dell. Even if you don't see something on the Leo's Picks Page you want – the Picks Page is just stuff that I've recently bought – you can click any of the links there, go to Dell. If you're in the U.S. And I get a number of emails saying, oh, I'd like to support you from Canada or Australia, or Dell's all over the world. They only count U.S. purchasers. So go ahead and buy your Dell equipment, but you don't need to go through our site if you're outside the U.S. If you're in the U.S., I'd appreciate it if you'd go through that site. TWiT.tv/dell. Thanks to you and to Dell for supporting the TWiT network.

Also thanks to our long-term sponsor, Astaro Corporation. We're so glad to welcome them back for yet another year of support of Security Now!. They really – they were the first advertiser on the TWiT network. And signing another year-long deal really shows their commitment to Security Now!. And I just am so grateful to them. And it's a good

relationship. They make one of the best security gateways out there, the Astaro Security Gateway. Open source-based, powerful. If your small or medium business network needs superior protection from – listen to what it does – spam, from viruses, from hackers, complete VPN capabilities, intrusion detection, content filtering, and an industrial-strength firewall, and it's all in one single easy-to-use high-performance appliance, this is it for you. Contact Astaro, Astaro.com, or call 877-4AS-TARO to schedule a free trial of an Astaro Security Gateway appliance in your business. They're great. It's a great product. We're glad to have them on the Security Now! show, and we thank them for their support.

So what is this program you're working on?

Steve: Well, this is an outgrowth of a discussion you and I had about a month ago, I guess, when we were talking about 32- versus 64-bit security in the forthcoming Windows Vista kernel. You may remember that we made the mistake briefly of saying, well, you know, nobody has 64-bit systems. Those are big iron server platforms. And so the whole notion of the enhanced security that Vista offers users of the 64-bit kernel, which is significant and can be significant, we sort of saw that as being off in the distance.

Well, you got a bunch of mail; I got a bunch of mail. People were saying, wait a minute, I've got 64 bits now. Certainly there's lots of AMD 64 users who have personal workstations that are already 64 bits. Now, they may be still running, and they probably are in most cases, the 32-bit Windows XP kernel because, as we know, compatibility of hardware device drivers has been a problem historically. A 64-bit XP kernel has existed for years, yet its adoption rate has been very low because there just hasn't been a demand for it. There's the same sort of chicken-and-egg thing. The manufacturers aren't doing 64-bit drivers because the users are not demanding them because they're not running 64-bit XP, but they're not running 64-bit XP because there aren't 64-bit drivers for the things they want to do.

So certainly this is going to be solved moving forward because 64 bits really does begin, as we have seen, to give us substantially better kernel protection. As we've said before, Microsoft has been forced, in the name of backward compatibility, to not fix a lot of things they could fix in the 32-bit version even of Vista because it would break things that are coming forward from XP. Instead they've said, okay, we're going to make the really robust security enhancements that we wish we could make over on the 32-bit platform, we're going to make those on the 64-bit platform.

So the point was, after all that, I poked around and did some research, wondering, okay, how prevalent, how pervasive is 64-bit support? And what I learned was that, in fact, as people who wrote to us were correct in saying, not only are the obviously AMD 64 chips 64-bit capable, but Intel has a technology they call EM64T, which is full 64-bit support, which they've been shipping for years – not 10 years, but two or three...

Leo: Some time. I remember talking with Andy Grove, and he was so mad that Microsoft wasn't supporting the 64-bit capabilities of their processors.

Steve: Well, exactly. So what I realized was many people who've been purchasing computers recently, in the last few years, may in fact have 64-bit support and not know it.

So, okay, so there was that. Then there's this whole issue of software versus hardware DEP. DEP is the Data Execution Prevention technology which – and this is where I'm going to spend most of our time today, talking about the significance and the quirky details of that, which are beginning to be revealed as I'm actually messing around with this stuff, and through a whole bunch of our great newsgroup posters over in GRC who have been messing with a couple of the early releases of this software already and sort of getting themselves revved up and finding out

what's going on. So there's an issue of hardware versus software support for preventing buffer overruns. And that's certainly a very important security feature. And then finally we've talked about the VMX, or as AMD calls it – no, wait. Is it VMX? It's something like that.

Leo: I can never keep track of all these acronyms.

Steve: I know. Intel calls it VT. But in either case, they are compatible – these are the virtual machine extensions which are now being supported, just now being supported, I mean, Parallels now supports it. I heard that Virtual PC, Microsoft's product, now supports it. Essentially, this allows virtual machines to run at identical performance as the host OS is. It's a tremendous performance boost. And Microsoft has made noises – as far as I know these are unofficial noises – that they will be able to use the virtual machine enhancements in a future version of probably, and I'm assuming only, the 64-bit version of Windows Vista, maybe 32, I just don't know. But they're talking about introducing a hypervisor technology to further lock down the kernel.

Because as we have talked about recently, Microsoft's Kernel Patch Prevention, the KPP, or also known as PatchGuard, it doesn't actually prevent patches. As we've seen, every five to 10 minutes it checks to see if there have been any. And if so, it just shuts Windows down. So it turns out that that's what Microsoft is doing now. Ultimately they should be able to literally lock the kernel using a hypervisor, that is to say, something running at a level even lower than the kernel. Well, that's got to be the hardware, which is now becoming available.

So what I realized, pushing back from this, is there are three things which are really important security hardware features: whether your chip has 64 bits; whether it supports hardware Data Execution Prevention, DEP; and whether it has the virtual machine extensions. And so I started to write a program called SecurAble, which is how securable is your system based on the hardware capabilities? And so what SecurAble does is it's going to be a piece of freeware, probably available in a couple of weeks. It's running. Newsgroup people have been using it. I'm pursuing the final details. As is often the case, it's ended up being more interesting than I expected because of how much I'm learning about what's going on inside of these chips in order to really make this be a really cool little turnkey robust utility, like I like all my little freeware stuff to be.

So what SecurAble does, it's, I don't know, it's like 26K or something, all in Assembly language, of course, although it's going to be bigger because I'm digitally signing it. And as it turns out, I have had to write a kernel driver for this also, which will be digitally signed. I did verify, by the way, Leo, we were talking actually last week in Toronto about whether Microsoft's requirement that Windows 64 – oh, no, it's all Vista. It's Vista drivers, all Vista drivers – no. It can't be all the...

Leo: We were talking about the issue on 64, whether Microsoft had to sign the drivers or just anybody sign the drivers.

Steve: Exactly. And it's not 32 bits because that would break everything. It's just 64-bit. So Microsoft in Windows Vista, and I think it's also the case in 64-bit XP, is that only signed drivers will be loaded by the kernel.

Leo: But signed by whom is the question?

Steve: I did verify Microsoft has a certificate that you can freely download from them which knows about your existing Authenticode certificates. For example, I have a digital certificate,

cost me about $500, which I get from VeriSign. And that allows me to sign my EXEs so that they're recognized by Windows. One of the things that Microsoft has been doing is they're beginning to raise the bar in cautioning users about running code they've downloaded from the Internet. And my older freeware, if you download it and run it, it'll always pop up a dialogue because it...

Leo: I've seen it, yeah.

Steve: Well, I mean, on anyone's software, typically any freeware that you download, Microsoft tags it that it's from the Internet and so pops up a dialogue saying, hey, this is an unknown publisher, are you sure you want to run this?

Leo: You see that a lot. And this is in 32-bit as well as 64-bit.

Steve: Yes. And again, so we're seeing this now in current versions of XP, for example. So signing this means that, when you download the software, you still get the dialogue. But instead of saying "Unknown Software Publisher," it says you've downloaded something from Gibson Research Corporation. And so it authenticates that that's where the software is from. It's also doing a full cryptographic verification to confirm that there's been no modification to that executable from the time I signed it until it was checked in the user's machine. So it really does provide good authentication and integrity checking for that. And I think you're able to say, "Always trust software from this publisher." You can certainly say "Don't ask me about this specific program every time I run it." And so you're able to say, okay, yeah yeah yeah, we know who Steve is, we like his stuff, we're going to run it without any questions. Or you can say, well, ask me, but only ask me the first time I run stuff that we download from GRC or whomever.

So the point is that that authentication is extendable using another certificate which is freely downloadable by Microsoft. You sort of double sign this with Microsoft's certificate and with the one that you get from VeriSign or any of the other people who provide this Authenticode certification, and then without any further involvement of Microsoft you can sign drivers. And so, and I'll know all about this firsthand because this is what I'll be doing here very shortly. So SecurAble...

Leo: So it's free, is the key, though. You don't have to pay Microsoft a bunch of money.

Steve: Well, yes. There is that...

Leo: Do you have to get an Authenticode license?

Steve: Yes, so it's not free. And this is being controversial out in the open source community...

Leo: Yeah, because if I'm a free program, I have to – how much is the Authenticode certificate?

Steve: It's $500.

Leo: Hey, hey. Geez, Louise.

Steve: And in fact it might even be more. I might have paid $799. But I think I got it for many years.

Leo: That's dumb because that's going to discourage shareware authors, I have to say, and certainly open source authors.

Steve: Well, here we are faced with a tradeoff. First of all, the 32-bit systems will continue to work as before. So it's only the 64-bit systems that have the burden of requiring signed drivers. The flipside is that certainly – I mean, it's raising the bar. What's annoying, it's sort of like airport security now. It's not clear that you are absolutely secure after putting up with all this because you could imagine that somehow credentials are going to get loose in the world, and there will be malware which will be signed. You can imagine that the malware authors are going to find a way to get their junk signed if that becomes important in order for them to be able to make this happen. And you could argue then that – and this is the argument that's being made, is that, as you say, it's the open source, the free source guys who are going to be inconvenienced because the whole idea is they want this to be a free and inexpensive platform. And in fact I think that's one of the points that's being made by – you mentioned it yesterday on TWiT, Leo, or in last Sunday's TWiT, that the Free Software Foundation had launched a campaign against Vista.

Leo: BadVista.org, yeah. Bad Vista.

Steve: BadVista.org. And this is one of the issues they raise is that it's becoming more and more hostile to free software.

Leo: You shouldn't have to pay a Microsoft tax to be a software developer. It's going to backfire on Microsoft. I don't think they understand how important – it certainly is on the Mac side, and Linux – these independent, you know, people like you who want to develop good, free software. And, now, you can afford to buy an Authenticode certificate, but not everybody can. That's going to kill that entire ecology. And that's important.

Steve: Well, you'll get a kick out of this, Leo. For the longest time, I mean, I've had Authenticode certificates for years. And I've never been able to get myself to use them because they increase the size of my code by 4K.

Leo: They double it. Oh, okay, 4K. So from 26 to 30K.

Steve: It just bugged me. It's like, okay, it's another chunk of bloatware. It doesn't have to be – there's nothing that needs to be 4K. It doesn't have to be a big thing because there's nothing large...

Leo: It's probably a GUID, probably not much in there.

Steve: Well, there was some discussion of this in our newsgroups recently. And I mentioned

that, when you put in a minimal PGP signature on a piece of email, that's not a 4K blob. It's a couple lines of little cryptic-looking stuff. That's all you need in order to have a publicly signed hash of some content. So anyway, it's people who don't care about size end up with big things. And so that's what Authenticode is. So...

Leo: Anyway, that's BadVista.org, if you want to read more about that. And I'm kind of in their ball- you know what'll happen is just people won't do open source software for Vista.

Steve: Well, arguably, though, to take the flipside, because I think it's important we look at both sides, it will increase security.

Leo: Yeah, and I understand that.

Steve: Requiring signed drivers, requiring that somebody's name be on this...

Leo: So give away Authenticode. Have a free Authenticode certificate for free and open source developers.

Steve: You could imagine that, like, SourceForge would have something like that.

Leo: Maybe that's what they'll do, yeah.

Steve: But the problem is, as soon as you do that, as soon as SourceForge gets together and does a single certificate that everyone uses, well, then it's completely available to the hackers.

Leo: That's a good point, yeah.

Steve: This is, for me, my Authenticode certificate is my private key. It is a private key that is associated with Gibson Research Corporation. I protect it dearly. I would never want to lose control of this because it would allow bad people to claim that their software was from me. So it is something that I keep locked up tight. It is in its own encrypted location, and protected, and I treat it very seriously.

Leo: As you should.

Steve: So the problem is, if we end up with Authenticode credentials which are floating around freely – and we know it's going to happen, a couple years from now when it becomes important that'll happen – then people are going to have to start looking at who it is that signed this, not assume that anything signed is good, and then we're going to have to have revocation lists that say these are the known bad keys which have gotten away and which malware is being signed with. So again, it raises the bar. It's an inconvenience for people. It's not a black-and-white perfect solution because there isn't one. But it's probably a good thing to do. Oh, and you can disable this administratively, and you can enable it on 32-bit systems administratively.

Leo: Oh, you can.

Steve: Yes.

Leo: Oh, I'm going to make a note of that. So you don't have to run 64-bit Windows to have the benefit of this signed code.

Steve: And now I have to back off from that because I'm not sure if that's only developer – if that might be the development build. You can imagine that developers don't want to have to be continually signing their code while they're debugging it. So it may be that the debugging build is...

Leo: Well, I'm more interested in the ability to turn it on than the ability to turn it off. I could see why they'd want to turn it off, but turning it on is great.

Steve: It's absolutely the case that 32-bit XP and 32-bit Vista, XP and Vista can require all drivers be signed.

Leo: I think that's great. We need to publish the code required to do that. I think that's fantastic. For 32-bit, turn it on in 32-bit, you'll be much safer; right?

Steve: Well, the problem is, since it's never been on by default, no drivers are signed.

Leo: Oh, that's true. So you'll have the same compatibility issues that you're going to have in 64-bit anyway.

Steve: Yeah. Now, it's interesting, though, because the signing technology is not a big deal. You know, if I were really uptight about this, I could sign other people's drivers just so that they have a signature, then turn that on in XP. And then if anything else, if any hostile junk tried to get itself installed, it would be blocked. And then I could say, wait a minute, what's this that I have not signed?

Leo: So an advanced user could have his own certificate. Would he have to have the Authenticode, or could he just use the free one from Microsoft?

Steve: No no no. The free one from Microsoft goes hand in hand with...

Leo: You still need Authenticode.

Steve: Yes, with a commercially purchased certificate.

Leo: But a business – here's a good use for it. A business, the IT department could

purchase one certificate, sign acceptable stuff, turn this feature on in 32-bit, and then nothing else will run.

Steve: Yeah. And that's something that you are able to do at the – what's the name in Windows for the whole bunch of extra settings that IT can control? It's not...

Leo: Oh, Policy Editor.

Steve: Yes, it's in Group Policy Editor. That's where you're able to set a policy. And it could be pushed out corporate wide.

Leo: I think that's good. That is a very smart thing to do if you're in an IT department.

Steve: It's a cool trick. It's worth mentioning, though, you were talking about how I can afford Authenticode because, you know, this is what I'm doing, because for me certainly as GRC, having my stuff signed is important.

Leo: You're a business.

Steve: But it's not like it's really that free even now to develop for Windows. It's like, well, yes, you can write code. But I pay Microsoft $2500 a month...

Leo: A month? A year.

Steve: I mean, I'm sorry, a year...

Leo: Oh, boy.

Steve: ...a year for the MSDN package, which is a big blob of DVDs which they are constantly revving...

Leo: Do you have to have that? Couldn't you just buy Visual Studio? Do you have to have the full MSDN library? Is there that much new stuff that you need to have?

Steve: No, you're right. What I'm getting – you're exactly correct. What I'm getting is, like, all the OSes and all the versions of everything and the developer tools. You're right. You could get a just...

Leo: Just buy Visual Studio. But that's still...

Steve: Their development stuff.

Leo: That's still expensive. I mean, it's not cheap.

Steve: No. In fact, they're increasing the price, and they're, like, changing the way it works. It's many hundreds of dollars for that. So...

Leo: I may be wrong, but I just think that an operating system relies quite a bit on these independent, third-party, sometimes very small developers, including open source and free software developers. Those are important.

Steve: Well, and the developer community has got all these tools that are available, too. So there are free compilers; there are free libraries. You can definitely do it on the cheap. But if you go the normal sanctioned route, it's already not free to write software for Windows.

Leo: You're right. Good point.

Steve: So anyway...

Leo: Which, then, by the way, the complete opposite on the Mac platform. They give you all that stuff.

Steve: No kidding.

Leo: Yeah.

Steve: Oh, very cool.

Leo: It comes on your install disk.

Steve: Well, and for example, there have been other compilers than Visual Studio. Watcom was available; certainly Symantec has made a compiler for years.

Leo: But they're all dead now because everybody uses Visual Studio. I know AnalogX still uses – Mark still uses Watcom. But still, most people...

Steve: He's off blazing his own trail.

Leo: Does he still use Watcom?

Steve: As far as I know. I can't imagine him changing.

Leo: Because it hasn't been updated in ages.

Steve: Yeah, but it works.

Leo: And he's got all the macros and the libraries and everything. He just doesn't need anything new. All right. So...

Steve: So what SecurAble does is – and this will be the forthcoming freebie from me. It's only a couple weeks away. I'm just nailing down details at this point, and there's some issues of chip identification because I wanted to tell people exactly what kind of processor they've got running in their system. What it does when you run it is it shows you three things about your system, about the system it's running on: whether it does have 64-bit emulation technology, which Intel calls EM64T; or whether it's a 64-bit AMD system that might be running in 32-bit mode so you didn't know it. So that would tell you whether, at your choice, you had the option of running either 64-bit version of XP or the forthcoming 64-bit version of Vista that will definitely give you much more security. I know that you said in TWiT last week, Leo, and we talked about it also up in Toronto, you've got the 64-bit Vista, and you're rubbing your hands together, getting ready to load it and see how it goes.

Leo: That's my project today. In fact, I'm just, right now, as we speak, Drive Snapshot is making an image of my Vista 32-bit off the laptop so if it really is a horrendous experience I can quickly go back to 32-bit. And I've got the Vista 64 disk sitting right in front of me. Huge, though, by the way. You need a DVD. It's almost 4 gigs.

Steve: Yes, it is a lot big- well, and even 32-bit is 2.678 gigs or something, so...

Leo: Is the 64-bit code inherently bigger?

Steve: What I've heard, and this is just anecdotal, I've heard that the 64-bit system for whatever reason also includes 32-bit versions of all of the apps in addition to 64. So you're getting double the trouble, essentially. You may get there before I do. I very much want to be running this. In fact, I probably will have to before I'm able to put the 1.0 version stamp on SecurAble. My supposition, though, is that Microsoft has accumulated so many drivers for existing PC hardware that certainly Vista 32 knows all about the hardware that I've exposed it to so far. It's running on all kinds of stuff. Even an old tablet PC, my old HP TC1100 is running it perfectly. So I would imagine that Microsoft has taken all of those drivers, compiled them in 64-bit version, and that the 64-bit Vista may be very hardware compatible, which would really help it a lot.

Leo: Wow, very interesting.

Steve: So SecurAble will tell you when you run it whether you've got 64 bits hiding in your chip that you may not know, that would allow you to run the more secure 64-bit version of Vista, or for that matter the current 64-bit version of XP. It'll tell you whether you've got hardware DEP. Now, DEP is, I think, the single most important security feature to happen in a decade. I'm not kidding.

Leo: Data Execution Prevention.

Steve: Yes. What this is – and we've talked about it in passing, in fact you and Paul talked about it, it was one of the title acronyms in last Windows Weekly a couple weeks ago – what this does is essentially it allows the system to stop virtually all buffer overruns. And that's big. I mean, all the security problems that we encounter with incredibly small exception are buffer overrun attacks. Anyone who's been listening to Security Now! for the last 71 weeks has heard buffer overrun, buffer overrun, over and over and over. These are the kinds of problems which catch programmers.

And we've talked about how the stack works. And as a consequence of the way it works, if somehow it's possible to inject a longer string of data or a longer response into some sort of control that isn't expecting that, that doesn't specifically limit it and clip it off – and more often than not it's something the programmers have to deliberately do – if something's able to do that, it's possible to cause the computer to be remotely compromised. And every second Tuesday, I mean literally every second Tuesday of the month, Microsoft is patching multiple buffer overrun problems. They're just all over the place.

So the idea is, the architecture of our computers is based on 4 Kbyte, what's called "pages" of memory. And the idea is that this system deals with memory in these 4K pages. It's 4 Kbyte pages which are swapped out to the virtual memory partition as the system is shuffling its memory around. Everything is done in these 4K pages. These pages are controlled by the system's memory management system, which knows which pages are swapped and which ones are out, where they're physically located in memory.

Traditionally pages have had bits associated with them that sort of governed what the system could do. There were read permission and write permission bits which individually allowed, on a 4K page granularity, it allowed a 4K page to be marked as readable and/or writeable. So it was possible for the system to protect pages from being written and make them read-only, or to protect them from being read, for example, and make them write-only, although that's not commonly done. And essentially, whenever any code – and this is enforced at the hardware level, that is, in the chip's silicon, which is what makes it so powerful – any code that violated the rules that had been set by those bits governing that page would generate a hardware exception, as it's called, that would yank control instantly away from the program that had misbehaved and transfer control to the operating system along with information about what exactly it was that had just happened.

Leo: So it doesn't shut the system down.

Steve: No.

Leo: It just takes control away.

Steve: Yes, it just says "bad code."

Leo: Bad code.

Steve: Bad code. So what has happened is, recently, and this is only in the last couple years, Intel and AMD have added another bit to the page management system. In addition to read and write permission, there is now execute permission. And this is huge. I mean, it is such an

obvious thing. It's the kind of thing, it's like why didn't we have that 20 years ago? I mean, it would have been trivial to put in, just didn't occur to anyone.

Leo: Well, there wasn't a problem 20 years ago.

Steve: Okay.

Leo: Or was there?

Steve: Well, there was 15 years ago. Yeah, there's always been a problem.

Leo: I guess. It's just bad code. If it's not a hacker problem, it's just a problem.

Steve: UNIX predates Windows. It's 20 years old. It's had rootkit problems from day one. And those have been buffer overrun exploits. So it's always been a problem.

Leo: It's bad programming. It's things like using STR copy instead of STRN copy. There's no range checking. It's sloppy programming. That's what gets me.

Steve: Well, I don't mean to disagree with you. But as a programmer, I can tell you that, I mean, nothing...

Leo: Well, you're in Assembly code. You've got no range checking unless you write it by hand.

Steve: Well, yes. But I live in fear that I'm going to screw up. My point is that it is so easy to take your eye off the ball for just a minute. And the idea, when we talked about several weeks ago why is security so hard, it's that the programmer's, the developer's mindset is "get this thing to work." Now, I have a heavy layer of " and it better not have any security flaws in it." And certainly any conscientious programmer who's writing Internet code today has got to be aware of security. That's going to be in the forefront of their mind. That's going to be a focus for them. So certainly they're aware of that. But it is easy to make a mistake.

Leo: All right. I'll be charitable.

Steve: So here's what's cool is that when the system supports DEP, all of the pages which the system does not expect to have code executing in can be marked as non-executable. So...

Leo: That would be the stack, for one thing.

Steve: It'd be the stack. It would be something called the "heap." A heap is something we've never really talked about, but it's the way the system makes memory freely available for allocation by program. So, for example, if I need a 4K buffer, I'll allocate a buffer. I can allocate

it from the stack, or I can allocate it from the heap.

Leo: Now, unfortunately, a lot of programmers put executable code in the heap. It's not an unusual technique.

Steve: Well, I wouldn't say "a lot." I mean, it certainly is the case that it can be done. And...

Leo: Even in the stack, I mean, there are some. And that's why those programs break when you use DEP.

Steve: Well, yes. And historically, in fact, the very early Windows GUI, when processors were 4.77 MHz, and we were trying, it was just painful to drag a window around the screen – in fact, you didn't drag the whole window. You just dragged a little dotted outline because you couldn't drag the whole window. It would have just been really excruciatingly slow. Back then, Microsoft's clever – there was a thing called a bitblit instruction, or the blitter is the thing in old GUI terminology that moves rectangles of pixels around the screen. The blitter was actually written on the stack on the fly in order to make it as fast as possible. So old Windows actually, by policy and by technology, used the stack as an executable scratchpad.

So it turns out, though, that many programs no longer do that. And as you said, Leo, it's true that some do. It's certainly the case that virtually all exploits are buffer overrun mistakes, which all would be caught if the system had hardware data execution prevention. So, and it turns out that many of the chips introduced in the last few years have it.

Now, what we're learned, however, is a couple things. Many of the BIOSes that the people in our newsgroups have, on systems that have hardware data execution prevention, they've got an option in the BIOS for turning it off. And in fact, I bought a little HP Pavilion here not long ago, it has the option for turning it off. At least mine was on by default. Many people are finding that theirs is off by default, meaning that at boot time the BIOS disables the hardware data execution prevention option in the chip, such that when the computer looks at it, it believes it's not present.

So anyway, it turns out that there's something called model-specific registers, MSRs, which is what the BIOS uses for turning this off. User mode code, that is, like my program SecurAble, is unable to access the model-specific registers from user mode without blowing up. It generates a data exception error, and it will shut down the program. So I'm unable to see from user mode whether this bit has been set or not. Thus I've written a kernel mode driver which SecurAble brings along with it. It's still a single EXE, it's still small, it just drops it out of the bottom of itself, briefly loads it in the kernel and checks the model-specific registers, sends that information back up to my little SecurAble app, and then removes the driver. So it's just installed in the kernel briefly. In a blink it's removed. And that allows me not only to tell users whether the system can see whether they've got data execution prevention, but also whether they might actually have it, if they think they don't, but their BIOS is one of those which defaults to having it disabled by default. So anyway, I've rolled my sleeves up...

Leo: No kidding. This sounds like sleeve-rolling time.

Steve: Yeah, it's been sleeve-rolling time. So basically it's going to be a complete presentation of these three security features: whether you've got 64 bits, whether you actually have hardware DEP support, and whether your system supports the virtual technology extensions. And it turns out, though, that this has also sprung into a second piece of software, which will be called – I think it's going to be called "inDEPth", and that's going to be something that's going

to verify all these DEP things for users. That is, it's one thing to know your system has the capability. It turns out, though, that this DEP has many different modes of operation, and some of this is very mysterious. For example, you can run it, as you mentioned about opting in and opting out, there are opt-in and opt-out modes. The opt-in is the normal case, and no things are opted in except some of Windows binary, some of Windows code...

Leo: That they know won't run unless you turn it off.

Steve: Well, no, that they know will run...

Leo: Oh, by default, I see.

Steve: Yeah. So their binary, some of theirs are enabled for DEP because they know they're safe, and they would like any buffer overruns that occurred in them to be caught. But, for example, rest of the applications, lots of applications and ActiveX controls are where these problems are. It turns out that you can then use a mode called Opt-Out, where Microsoft says that everything not opted out of will be checked for any buffer overruns, which would lead you to believe that everything would be checked, except there's a third mode called Always On. And it turns out – get this – that you would think that Always On would be the same as Opt-Out, if nothing had opted out. It's not. Many people have discovered that there are programs they use which work fine with Opt-Out but do not work with Always On.

Leo: Why not? So they're not checking those programs?

Steve: It's a mystery. My hypothesis is that the opt-in/opt-out might only be checking the user mode buffer overruns and not device drivers. But when you use Always On, it's checking everything system-wide, and there are device drivers where there are problems. For example, the current build of Ad-Aware fails this. Users in our newsgroups are posting their findings as we're experimenting with this. I will have a complete comprehensive presentation to offer our Security Now! listeners a few weeks from now, once I've unraveled all these mysteries. But what's interesting is there is zero documentation of this on Microsoft's site. I mean, we have been scouring, looking for any explanation, for example, for why Always On is different from Opt-Out, where nothing opts out. And Microsoft says nothing about it. So it's going to end up being, okay, do a bunch of experiments, figure it out the hard way. And that's what I'm going to do. So anyway, I'm having a ball writing some new free software that arose from Security Now!, and it'll be made available to everyone, and certainly to all of our Security Now! listeners.

Leo: Excellent, excellent, excellent. Absolutely free, as always. And a lot of nice features, it sounds like. In fact, it sounds like the kind of program you'd want to run just to kind of know what's going on in your system.

Steve: Well, and one other thing, too, is that people have mentioned, hey, you know, they could send this to someone whose computer they were going to purchase, or you might be able to get a dealer to run it. Or if some...

Leo: Tell me what I'm buying here. That's a great idea, yeah.

Steve: Exactly. If a friend of yours said, hey, I just got a really cool system, you know, from

Dell, for example. Someone could say, hey, well, run OptOut – OptOut is a long time ago. Run SecurAble from GRC, tell me what it says. Because, I mean, it's a very simple display. It says

"64, yes, yes" if you've got everything, or "64, yes, no" if you've got DEP but not the virtual technology extensions. So it's be a very super quick way of telling you some things you care about, about the security of the system you're on.

Leo: Right. And we'll look for that in the next couple of weeks. But you're going to take some time off for the holidays, I hope. Please don't work on...

Steve: No no no. I'm right in the middle of this. I'm having a ball.

Leo: Oh, Steve, Steve, Steve. What are you doing for Christmas, anything?

Steve: I'm just going home for a couple days, up to see my family in San Mateo.

Leo: All right. Well, Steve, as always, what a great pleasure having you on. And I just can't thank you enough for doing the work you do to secure us all. I mean, it's little things like SecurAble that really – you give them away. And it makes a big difference, I think. You probably don't even know the kind of impact you have made. But I do, and I thank you for it, and...

Steve: Oh, I love doing it.

Leo: It's been a great year of shows. We have one more before the end of the year. What are we going to talk about next time?

Steve: Well, it's #72.

Leo: Oh, we're going to Q&A it.

Steve: It's our Q&A #14 next week.

Leo: We'll do that next week. Unlike all the other TWiT podcasts, Steve is sticking around. And so is SpinRite. And, you know, people have hard drive problems even on the holidays, don't they, Steve.

Steve: Maybe even more so when the kids come home from school.

Leo: Or they get a new computer and it's just dead on arrival. SpinRite is the solution. It's the ultimate disk recovery and maintenance utility. It's Steve's day job, and he works just as hard on it as he does on all the security stuff. SpinRite.info if you want to know more. A lot of good testimonials in there. And of course GRC.com is Steve's site, where you'll find

the 16KB versions of this show for the bandwidth impaired, the transcripts thanks to Elaine, and more information, as well. GRC.com.