SECURITY NOW!

Transcript of Episode #105

Firewall LeakTesting

Description: Steve and Leo discuss the history, purpose, and value of personal firewall leaktesting. They examine the myriad techniques clever developers have found for accessing the Internet and sending data out of PCs even when those PCs are being protected by outbound-blocking personal firewalls.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-105.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-105-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 105 for August 16, 2007: LeakTests.

Hey, it's time for Security Now!, a chance to look at security in all its guises with the expert, Steve Gibson of GRC.com, now entering our third year of podcasting.

Steve Gibson: That's right, Episode No. 105, which is this one, is the first episode of year three of Security Now!.

Leo: And I understand we have some very good news...

Steve: Yay.

Leo: ...to celebrate with.

Steve: Yes. I absolutely want to start off by thanking our listeners. It is clearly 100 percent their, I don't want to say "fault," but, you know, their work that has allowed us to take People's Choice Best Science & Technology Podcast for this year. We don't know what the numbers are, that is, by how much we won. But two weeks ago we did a quick little special podcast, Leo, when I realized that this podcast awards was up. And it was, overall, it was up for 15 days. The podcast award people report that 1.3 million unique people voted. During one hour, the busiest hour, 22,109 people voted during a one-hour window. The average person voted three times during that 15 days. And of course, you know, multiple repeat voting was formally endorsed and allowed. So I said to our own listeners, vote often, vote immediately, vote - every 24 hours you were able to log one more vote. And so as a consequence - and I'm sure this wouldn't have happened if it weren't for our listeners and actually for us saying, hey, let's win this year because we haven't for the last two - we got People's Choice Best Podcast for Science & Technology category. So...

Leo: Congratulations, Steve. You deserve it.

Steve: Well, I'm really happy. The feedback we get is so positive. And I just love it that we were able to ask our listeners for this, and they came through. So...

Leo: Well, and I also like it that we didn't have to ask more than once. We didn't have to beg and plead and say, please vote for us every show. We forgot. We only asked once. But that's good. That's good.

Steve: Yeah, very cool.

Leo: So you'll get your award at the Podcast & New Media Expo, I presume.

Steve: Yes, in fact, that's my annual opportunity to hang out with Elaine, as you know, our transcriber. I hope and I expect that she will be able to make the trek out from her wilderness hideout wherever that is with the satellite link that's always giving her trouble downloading our audio files. But you're not going to be there this year.

Leo: I can't make it this year, I'm sorry to say. But I wish you all the best. And maybe now that I know you're winning an award I probably should come down at least that Friday. The reason is it's over a weekend, and I have to do the radio show, so...

Steve: Oh, of course. Well, and for me it's a drive.

Leo: You're there.

Steve: For you it's a roundtrip flight problem, so, yeah.

Leo: I'd have to fly down. So I'm sorry I won't be there, but congratulations. I think that's fantastic.

Steve: Well, I just, you know, when I saw that this was a possibility, it's like, oh, I want to win this. And so we have, thanks to our listeners. So I want to really thank them for that.

Leo: Very good. Well done.

Steve: In other errata, several people mentioned that - I don't know why, I mean, I don't even know what the capital of Australia is. I think the only city I'm really even aware of is Sydney. So without even thinking I said it was the capital. I don't even remember now what the context was. But...

Leo: I know what it was. We were talking about CAPTCHA, a different kind of CAPTCHA which asks you knowledge. And we were talking...

Steve: Oh, that's exactly what it was, right. Well, and...

Leo: You would have failed that one.

Steve: Yes. I said maybe not everyone knows that the capital of Australia is Sydney. And sure enough, you know...

Leo: It isn't.

Steve: Yeah. It's Canberra.

Leo: I didn't know that either.

Steve: C-a-n-b-e-r-r-a. So there is a finer point on why knowledge-based CAPTCHA is probably not a good idea. Not if you want me to, you know, be able to access your website.

Leo: Well, we'll know that from now on. That's wonderful.

Steve: Also, you may remember that I, in one of the - I'm not sure whether it was last week or the week before, I was excited to have discovered that Microsoft's CardSpace, their UI for single log-on interface, which was always in Vista, I was excited to tell people that, hey, it has appeared in my Control Panel in XP. And I said, you know, as long as your security patches are current, you'll have it.

Well, it turns out it's a little more involved than that. It's a component of .NET, which actually is to say that it's implemented in .NET as more and more, apparently, things will be in the future because this is Microsoft's new API initiative for Windows. And so it's only if you had gotten on the .NET bandwagon one way or another. The .NET components are optional uploads, or downloads, depending on which end you're standing on, for Windows Update. Many times you'll download some new thing that requires the .NET components in order for it to run. So people may be already on that. But if you went looking for CardSpace in your Control Panel, and your patches are all current, you may have wondered and been disappointed that you don't have it. The reason you don't if you didn't is you probably don't have the .NET component. So you need to install those, and then CardSpace will appear after you've done sufficient number of further updates.

Leo: It's part of .NET 3. But, you know, it's funny because I have it, and I ran it, and I filled out a Card, and that was about it. There's not much to do with it at this point. Don't go out and download it just so you can get it. I mean, it's not...

Steve: Yeah. We're definitely in that chicken-and-egg phase where people are waiting till it exists, and they're not going to, you know, it's not going to exist until enough people have it. So ultimately I think it's a nice UI for things like the OpenID interface. So I'm glad for that.

And here we are on Thursday the 16th, so it's worth pointing out that two days ago, on the second Tuesday of the month, that is to say, the 14th, Microsoft has dropped a bunch of security updates on us. All kinds of stuff. I think it's nine different updates. There's some serious problems in the XML core library support that they've got. And, you know, and a scattering of other things.

Also, in other news, Symantec had a bunch of problems in their various security suites. There's one common control that all of the Symantec security stuff use that had a remote execution vulnerability that was discovered. So you're going to want to make sure, if you're a Symantec user, that you update that. But also certainly, if you don't already have automatic updates running, you'll want to go grab this stuff from Microsoft pretty soon because it was one of our, you know, a large month worth of fixes that were available two days ago.

Leo: Weren't there also some Vista, for the first time some serious Vista flaws, as well?

Steve: Yes. Actually it's a consequence of the fact that Vista is really just, well, it's more than a candy coat or wrapping on top of XP. But Vista is built from XP inasmuch as most of Vista is existing code. So anything that's being found in XP that shares a common code base with Vista, which is to say almost everything, that will be a problem, too. It is the case that we have seen instances where Vista's heightened security and the evolution of this tightening that Microsoft is doing has prevented code that is vulnerable and would otherwise also be vulnerable in Vista from actually being exploitable under Vista. But you're right, in this case, Leo, the exploits that were found probably on an XP platform did map into Vista. And so Vista's stuff was not protecting people in that case.

Leo: Although, you know, I have to say, given that Vista's been out now seven months, it's been pretty reliable. We haven't seen a whole lot of critical flaws.

Steve: Well, and I don't think we're going to, frankly. I think those days are over because we've got firewalls running. As soon as we went to Service Pack 2 and the XP firewall was running by default, that really, I mean, it forever changed the landscape of Windows vulnerabilities because it was, you know, Microsoft was saying, oh, anyone who sets up their network stack and turns on networking, they'll run across the wizard for turning your firewall on. And it's like, okay - that is to say, in XP prior to Service Pack 2, it's like, okay, if that's true, why are all these XP systems having their ports exposed? I mean, that was the nature of all these problems that Windows kept having was that people were running XP without the personal firewall running. Which of course, you know, Microsoft finally got a clue with Service Pack 2 and said, okay, we just have to turn this on by default. The moment they did that, everything changed. Now it really doesn't matter what nightmares you have in terms of, like, open ports and services running because they're going to be behind the firewall. Now the problem, and this is now what we're seeing, in fact, are this next generation of, you know, you go get the problems yourself by surfing to unsafe websites.

Leo: Right, right. Oh, well. That's what's interesting about this, as nasty as it is, it is always changing, and that makes it kind of interesting. And I have to say I'm somewhat grudgingly impressed by the ingenuity of these bad guys. They just keep coming at it.

Steve: Well, that's a perfect segue to today's topic, Leo, because ingenuity of bad guys is essentially what our listeners are going to really end up feeling when I wrap up a discussion of personal firewall leaktests, which is our topic for the day.

Leo: All right. I am prepared to hear about how leaky my system is.

Steve: Well, before I get into that, I did want to share a fun piece of email that came in to us with the subject "SpinRite Really Works." And I'm only going to...

Leo: Really, no kidding. I'll be danged.

Steve: Where have we heard that before?

Leo: SpinRite works?

Steve: I'll just give this guy's first name as Ken, for reasons you'll see here as I read this. He says, "Hi, Steve. I hope this the right address for testimonials. If not, please forward this to Steve Gibson as I want to thank him personally. I've been a loyal listener of Security Now! since Episode 1. Keep up the good work. I know it has opened my eyes to many different aspects of security. However, this email is meant to praise SpinRite. My father bought me a copy some time back, and I can thankfully say that personally I haven't had an opportunity to use it except for regular maintenance scans."

And of course, as we know, maintenance scans with SpinRite will very likely prevent him from ever needing to use it for recovery because it'll allow - SpinRite will work with the drive to allow the drive to discover problems before they've become critical.

And he says, "However, I work for a company that has employees that travel in Canada and Ecuador who depend heavily on laptops for communications and data analysis. Well, Friday an employee who was working in Ecuador had his computer crash. It would not boot Windows no matter what he tried to do, the lovely unmountable boot volume being the error of choice," says Ken. He says, "I suspected a few bad sectors had developed on the hard drive. And when they ran some diagnostics, it only further confirmed my suspicions. Unfortunately, repair was not an option for a number of weeks. So I felt that SpinRite was our best and only" - he says in parentheses with an exclamation point - "option. I didn't have time to purchase a site license." And he says, parens, "Sorry. But after this I had no trouble getting approval for it!"

Leo: Yeah, I bet.

Steve: "So I emailed my personal copy" - that was, you know, the present from his dad. He says, "I emailed my personal copy to them with usage instructions. Well, they ran the software on the machine this morning. And about two hours later I received an email from our employee from his now-booting computer."

Leo: Oh, hallelujah.

Steve: "Thanks so much for making a wonderful product. Chalk up another win for SpinRite." And he says, "If you choose to read my message on the podcast, please only use my first name (for security, of course)." So he says, "Thanks again," and we'll just call him Ken. And thank you for the report, Ken, I really appreciate that neat testimony.

Leo: That's really great. That's really wonderful. I've had, I'll tell you, a lot of people love Security Now!, and even more love SpinRite. So it's a nice combination. We're glad that SpinRite allows you to do Security Now!.

Steve: I really am, too. Okay.

Leo: Yes. Leaking. Leaking testing.

Steve: Okay. First of all, a little bit of history because I don't know whether our listeners know it or not, but I'm also the person who I guess coined the term "leaktest" because I wrote the first one that there ever was.

Leo: Right. Which is still your No. 1 downloaded program, I think.

Steve: Well, yes. In fact, that's one thing that I wanted to mention is it has always been the most downloaded tool we have. Which, as our listeners will understand in a second, this sort of strikes me as being sort of bizarre because my LeakTest is dumb. I mean, it's just - it does what it does. But it's like, oh my goodness, I mean, if the firewall can't pass mine, it has no business occupying any code on someone's computer. But I wrote it because once upon a time there was only one firewall that did pass, that is to say, which did not leak in the simple way I was testing. But to make, well, to finish that aspect, LeakTest version, I think it's at v2 because I made some changes a while back, it has been downloaded 6,715,096 times.

Leo: That's amazing.

Steve: So, and except for example when the Windows Metafile thing surfaced a year and a half ago, Leo, at that point then the MouseTrap tool that I created to deal with that, that briefly floated to the top of our most popular downloads. Then it sort of sifted back down in the list and took its rightful place, LeakTest resuming its No. 1 position. But I noticed just now when I brought the page up, the free popular page sorts the freeware that GRC offers by popularity. For some reason at the moment Wizmo is in No. 1 place. But it must - it could only be that someone somewhere wrote about Wizmo.

Leo: I think we mentioned it on the show or something, on the TV show.

Steve: Ah, that would do it. So at the moment it's outpacing LeakTest. It's seeing 1,286 downloads per day versus 724 for LeakTest. But Wizmo will calm down again, and LeakTest will once again be No. 1.

Leo: But you'd like to tell people to stop downloading it.

Steve: Well, I mean, people do. And if they want it, I'm glad that it's there. it's another one of my - it's like 25KB in size.

Leo: Well, there you go.

Steve: But here's the story of leaktests, that is to say, the very first LeakTest in the industry. I was using ZoneAlarm at the time, when ZoneAlarm was very new. It was out of beta, and it existed. And the thing that ZoneAlarm did was something that many other firewalls at the time also did, which is, you know, the whole reason ZoneAlarm was cool was it gave you outbound protection. That is, any firewall or NAT router or even Windows XP firewall will provide protection against inbound packets because, as any listeners of Security Now! will understand, the way that works is any unexpected packet, that is, a packet coming in that is not part of a connection that was initiated from the inside out, those packets will simply be dropped. Well, that's the way NAT routers work, as we know. Which is why the moment you put a NAT router in front of your computer, things get a lot quieter on your network because there's all this nonsense and, as I call it, "Internet Background Radiation," IBR, which is just junk packets floating around the Internet and old exploits that are scanning for still-vulnerable computers and so forth. So, you know, the Internet's just now, unfortunately, in the last five years become full of this debris. So a personal firewall, like even the firewall that's in XP, it'll block you from all that stuff.

But the real point of personal firewalls as such, the reason that people still install additional firewalls, even when they've got, for example, XP's firewall built in, is they want to know who is sending data out. They want control over which applications are being permitted to use their Internet connection. And you can argue that that's a really useful thing. I mean, the fact is, that is the way that I discovered the little bit of spyware which WinZip, the Windows version of WinZip installed on my machine - I'm sorry, PKZIP. It was a Windows version of PKZIP that installed some spyware on my machine. When I was beta testing the very first version of ZoneAlarm, it popped up a note and said, hey, tsadbot.exe wants permission to use your Internet connection. And I'm like, what?

Leo: Who? I don't think so.

Steve: And, you know, I had permitted Eudora and IE and a couple other net-using utilities. So the whole idea was that the LeakTest - I'm sorry, the personal firewall was doing outbound protection. And Symantec's did, and Norton's did, and Sygate's, and I mean, McAfee, you know, basically that was a feature that they were all boasting about. And there was probably 15 of them around at the time. However, I became aware of the fact that only ZoneAlarm was actually checking to see whether the application you had permitted to send data out of your computer was the one that was sending it. In other words, we all know that Internet Explorer's filename is IExplore.exe. So it turns out that fooling any of the outbound blocking firewalls except ZoneAlarm was as easy as a trojan or any other malware renaming itself IExplore.exe. In which case it really was able to explore. I mean, IExplore was the right name.

Leo: So all it was looking at is the process name. And once the process name had been approved, it didn't matter who it was.

Steve: Exactly. It was just looking at the name of the executable that was running. And in fact you could almost assume that IExplore.exe would be given permission, even if the person were, for example, a Firefox user.

Leo: Well, that's a good point. You just look at it and say, oh, Internet Explorer, I don't know why it's running, but I'm going to say okay.

Steve: Well, or you need to use IE for some things, you know, Windows Update and so forth. So even if you're not using IE all the time, you've probably had an occasion to give it firewall permission. So anything was simply able to rename itself to any application that had permission to exit the firewall. And no dialogues would pop up. Nothing would happen because the firewalls were just checking, as you said, the process name.

So when I realized that only ZoneAlarm was actually doing - I think it was in MD5 they were doing, as we know from our crypto series, that's a message digest. They were actually checking - they were running a cryptographic hash on the actual executable program to see if it's the same as it was when you permitted it. So, for example, if you said yes, IE is able to use my Internet connection, at that point ZoneAlarm would take a snapshot of the executable code, reducing it to a cryptographic hash. And it would store that along with the permission so that subsequently, when a program got itself going and tried to use your Internet connection, and it was called IE, ZoneAlarm would look in its list of permitted executables, see that IExplore.exe was permitted. But then I'm sure the first thing it did was check the length, which is a very fast thing to do, to verify that the length is the same. And if not, it would then generate a cryptographic hash and make sure that it matched the same file that had been originally given permission.

So, I mean, this was, you know, this is what you would expect. But it turns out not one other firewall did that. So I wrote, in the course of a few hours because it's a trivial tool, I wrote the very first LeakTest, which it just said to people, look, try this. Take any file you have and rename it to something that is permitted. And in fact the idea was that LeakTest itself, LeakTest was a very simple, benign, outbound tool. So the idea would be, you would run LeakTest, and up would pop a dialogue box that would say, hey, I'm LeakTest, do you want to permit me to use your Internet connection? So basically there were many different ways you could use this. But the idea was that it was able to test whether your firewall was simply verifying the process name or not. Well, I got in trouble with Symantec. They called because they were upset that I had...

Leo: How dare you test our software.

Steve: Exactly. They were really annoyed with me.

Leo: How dare you.

Steve: And they were saying, well, wait a minute, this is irresponsible. You've let everyone know...

Leo: That we don't work.

Steve: And it's like, wait a minute, yeah, this is not a buffer overflow or overrun problem. This is a feature which was implemented with no thought at all to actually preventing its exploit. I said, I'm sorry you're upset. Let's see how quickly you can fix it. Well, what's cool is that, with ShieldsUP!, GRC was enough on the map at the time. And the news of LeakTest just went like wildfire through the Internet, everyone realizing how dumb their firewalls were, if they weren't using ZoneAlarm. I'm sure many people switched to ZoneAlarm. But I also know that, within a very short time, every single other firewall fixed this problem.

Which is why I created LeakTest, was I wanted to demonstrate how trivial it was to bypass all other firewalls' so-called outbound protection, and in the process bring immediate pressure on them to get this fixed, which it was a couple weeks later all of them had released updates which added this very simple, you know, check the length, check the message digest, do a hash of the EXE and make sure that it's solid, and so this very obvious way for trojans to operate. Oh, and we were aware of trojans in the wild that were using this approach, just renaming themselves as a permitted application in order to function.

Leo: So now I see why you say who needs LeakTest, because that was an easy fix, and everybody's fixed it.

Steve: Yes, exactly. I mean, and it's freaky to me that here we are five years later, and it's still the No. 1 downloaded tool, and we're...

Leo: Well, I think people don't - now that you've explained it, I think just people just don't understand what it's testing and why you don't need to test it anymore.

Steve: Well, okay. So here's the cool thing, Leo, is this was just the beginning of leaktests. There are now something like 15 of them, all of them more useful and also much more powerful, many of them very clever. And what doe annoy me is, unfortunately, none of them have ever achieved enough real attention to cause firewalls to address them. Meaning that even though, for example, there's a website, firewallleaktester.com, maintains a list of all of these leaktests, talks about what they do, and shows this massive chart. And I've got links, by the way, in Episode 105 show notes to these things for people who are interested and concerned. This site does an inventory of all known firewalls and, well, certainly the most popular firewalls, I mean, a huge list of current, contemporary personal firewalls, and shows what version was tested, cross-referenced by all these other leaktests, which show that right now these leaktests cut right through these personal firewalls that are saying they're blocking these problems which are well known, and they haven't bothered to fix them.

So, I mean, my conscience every so often is pinged a little bit by this because I think, wow, you know, if GRC were to publicize these problems, we might again be able to get those firewalls to get themselves up to speed because we have the security-oriented profile to bring enough pressure on the likes of Symantec and McAfee and Sygate and on and on and on. I mean, just all of the firewalls. I don't mean to single those guys out. They just come to mind because they're so popular. But to say, look, it's time now to fix these problems. But I want to give our listeners a sense for how these other leaktests work, what they do, and essentially a little bit of a sense of the futility of this whole issue because the real issue is, and we've said this on the show so many times, but it's always worth repeating because it's such a truism, is once something is in your machine, it's almost too late.

Now, assuming that this something that malware, mal-thing that gets in your machine would trip an alarm on your personal firewall, then that's good. I mean, you know that it's there. And it's certainly better to know that it's there than not know that it's there. On the other hand, what are you going to do about it? Because these things are now virtually impossible to remove. They sink their hooks so deeply into your machine. Many of them are very sneaky and are really, as I said, they insinuate themselves at a level that makes it just so difficult to do anything other than roll back to a previous image. And in fact even the Windows built-in restore points are now being infected by malware, so that that system can't be used.

So, okay. One firewall that does something which was obvious but was not initially being caught is called TooLeaky. And all it does is it uses a freshly launched instance of IE to send personal data out as parameters to the query, which is, you know, sort of clever. That is, we know, for example, that when you have a URL and then the URL finishes, like .htm or whatever, and then you do a question mark, anything following the question mark are called "parameters to the query," meaning that the browser stops parsing the URL at that point, so it's bringing up the page that is defined up to the question mark, but it continues sending anything else after the question mark as part of the query. Well, so if this is http://evilserverinrussia.com/randompage.htm, question mark, your credit card number, well, that's a problem.

So the idea is, what TooLeaky does is it just launches IE and puts that HTML parameter containing your data in the query, and out it goes. So the problem is that your personal firewall may be set up so that it absolutely will not allow anything but IE to talk on the Internet; but this uses IE, the real IE, not a renamed IE, the real IE, to make a query. And the moment that query is allowed out, your personal information has escaped your control. So there's an example.

There's another one which was actually written by a friend of GRC. Robin Keir hangs out in our newsgroups a lot. He's a super sharp security guy. He wrote, also sort of following on the concept of LeakTest, he wrote one called FireHole, meaning a firewall hole. This demonstrates, and it was one of the very early demonstrations, something called "cross-process DLL injection." The idea in Windows is that, back in the good old days when we didn't have security problems, Microsoft was creating all kinds of features into Windows API. One of the things you're able to do in Windows is you're able to use a debugger to reach across process boundaries, that is, out of its own debugging process into another process. And it's able to attach itself to that process. You're able then to stop the threads of execution, basically stop that program from running and basically take it over, making the execution of that program go single step-by-step so you're able to watch it execute. That's what a contemporary debugger is able to do.

Well, one of the other things sort of in this mix of capabilities allows one process to essentially inject a DLL of its choosing into another process space, and cause that DLL to run. Well, what that means is that Robin's FireHole program, it looks around, and it sees, oh, look, servicehost.exe is running. Well, servicehost, for example, is where DNS queries come from. So many people see servicehost.exe popping up and asking for permission to use their firewall.

Leo: All the time.

Steve: All the time. Exactly. And so what Robin does is he injects his own - now, in this case it's not malicious, but in the case of malware it would be - he injects a small DLL into the servicehost.exe process and runs it. Now what happens is, because the DLL, his own code is running in the context of that servicehost.exe process, which has permission, the firewall that is doing outbound blocking sees that it's servicehost.exe making an outbound query. And that's permitted. And so basically you've broken out, the malware has broken out of its own process space by injecting code into another process. There's also one called YALTA, which is an acronym for Yet Another Leak Test Application.

Leo: But a good acronym. I like it.

Steve: It's a good acronym. PCAudit is another that also uses cross-process DLL injection. There's one called AWFT. It injects a thread into an existing running copy of a web browser like IE or even into Windows Explorer. That's another thing you're able to do using the Windows API. You're actually able to inject a thread of execution not requiring a whole DLL. You're actually able to take a chunk of memory and map that memory into an unused area of another process, and then tell that memory to start executing a new thread of execution in that memory in order to get something to work. There's another one called Thermite that's also a thread-injector.

And there's one called CopyCat that uses code injection without thread creation because some firewalls were becoming sensitized to the idea of threads being created by other processes. So they blocked that. But some guy said, okay, I can get around that. So he writes CopyCat, which basically it borrows an existing thread from the application. Basically, like a debugger, it stops that thread, notes where it was, saves the thread's registers and its so-called context, moves the thread over to its own code, runs that thread on its code, thus avoiding the need to create a new thread, then puts the thread back where it was and says, okay, go on about your business.

Leo: Unbelievable.

Steve: So these are the kinds of things that are being done. There's even one called MBtest, which uses the WinPcap drivers that I referred to a couple of weeks ago, very popular sort of raw packet monitoring and injection software which has been developed over the years. It installs that down in the network stack below the firewall in order to bypass the whole process. It just puts its own driver in the system. Meanwhile the firewall is sitting there, you know...

Leo: Everything's fine. It's all good.

Steve: Exactly, watching all the traffic going by in the IP stack, but not being installed low enough in the system, and not doing anything to explicitly block that.

Leo: How common is that exploit?

Steve: It was mostly done as a proof of concept. You need to bring along a bunch of drivers. And, for example...

Leo: Pretty sophisticated.

Steve: Yes, a non-admin cannot install this driver on the fly. So it would need to do a privileged escalation exploit in order to escalate its privileges to admin level. Then it would be able to install this driver on the fly. But the driver's actually very well behaved, and it's a beautifully written driver. So, again, it takes advantage of that and just does something that the firewall was not set up to expect.

There's one called WBreaker which uses Windows Explorer to launch Internet Explorer, or the command to launch Explorer, which then launches Internet Explorer. The reason this chaining occurs, it's one of the things that firewalls do is look to see who is launching a permitted application. Because notice that, remember earlier we talked about a piece of malware would be launching IE and using it to send its information out? So again the firewall said, oh, we can fix that. Some of them did. And then the leaktesting guys just said, okay, we can get around that, we'll use Windows Explorer to launch Internet Explorer because that's normally what does launch Internet Explorer is Windows Explorer. So that got around another set of firewalls that thought they had been tricky in preventing that. It's like, nope, we bypassed that.

There's a second version of PCAudit that uses - because after some firewalls blocked PCAudit, which is one of those that uses cross-process DLL injection, the guy said, okay, I can get around that, and so he did with PCAudit2. There is one called DNStest, which is very clever. It uses the DNS client service, which is called servicehost.exe, it's one of the many things that servicehost.exe hosts is the DNS client service, well, your computer has to be able to make DNS queries because that's what it's doing all the time. You go www.anything.com, and your computer needs to look up the IP for that. Well, it turns out that you could make a bogus query which contained, for example, someone's credit card number as the URL domain that you're looking up. And if it was set up right, it could be, for example, creditcardnumber.malwarerus.ru, and what happened would be malware, if there's a site malwarerus.ru were running their own DNS server, they would receive the query for the machine name within that domain, which is your credit card number. And, for that matter, any other information that they want to tack on there because it could be credit card number dot expiration date dot residential address information or billing address dot dot dot dot, and send out a long DNS query that is your personal information. And Windows doesn't know the difference. It thinks that's a valid URL, and it slides right out through a DNS query.

Leo: Unbelievable. Wow. There's no way to block something like that.

Steve: No, I mean, you would have to literally filter DNS and make some sort of heuristic decision about whether this looks reasonable or not, is that something that you want to allow to happen or not. And, I mean, and the list goes on. So, I mean, there's something called Surfer, there's Breakout v1 and v2, there's something called Jumper. Oh, get this. Jumper, for example, there's a registry entry called AppInit. AppInit registry entry allows DLLs to be registered for automatic loading by Windows. Well, now, there are reasons that would be useful. For example, there are core DLLs in the system, like kernel,dll, user32.dll, gdi32.dll. And those are fundamental parts of Windows.

Well, Microsoft thought, well, what if something comes along in the future that's another DLL that we just want any program to have access to? So literally there's a registry entry called AppInit where anyone who wants to can add random DLLs to that. And every time an application is initialized, Windows will dutifully look up the list of DLLs in that registry key and load all those DLLs into that application space. Well, malware says, hey, what a cool thing. Let's just register our self. And the next time an EXE starts, we'll get our DLL loaded. And that works.

Leo: Unbelievable. Just amazing.

Steve: So, I mean, it really is just phenomenal that essentially there are this many ways, known ways for malware to right now today get past every single personal firewall out there. I will have the link in our show notes to this chart. People can look up their firewall and see which of these exploits had even today never been handled despite the fact that all firewall makers know they exist. It's very well known. This site gets press from time to time, but never enough to force the firewall makers to fix the problem. And their argument is that some of these things are outside the purview of a firewall. They're more what's called HIPS, which stands for Host-based Intrusion Protection Systems, where the idea is something is in your system and is taking advantage, malicious advantage of features of the operating system for its own malicious purposes.

Well, there are non-firewall HIPS management tools, which for example attempt to prevent these sorts of cross-process vulnerabilities from being exploited, basically by hooking and filtering the API themselves and verifying whether applications have a right to do this. And it's worth noting that this is one area where Vista has really moved forward. They have this notion of a protected application which is essentially beginning finally to put barriers around sensitive applications like your email application and like your browser, and preventing their exploitation by other processes. And in fact we discussed this when we were talking about all of the additional DRM features that Vista has because that whole protected video and audio subsystem, it needs to protect itself from being, for example, debugged by debuggers who are going in and rifling through memory to find the HD-DVD decryption keys.

But I guess the point is that it's - I don't mean to raise a lot of alarm on behalf of our listeners, in our listeners' minds. But the fact is, almost without, I think without a single exception, when I looked last, which was only a few weeks ago, there was no single firewall that was invulnerable to all of these known exploits. Meaning...

Leo: So the leaktest is kind of pointless.

Steve: Well, yes, that's the problem is that we have a fundamentally insecure operating system in Windows. And you might argue that other operating systems that have not really rigorously maintained security as their No. 1 focus, they may similarly have become lax in various ways. You could forgive Microsoft in the case of Windows for this because once upon a time security wasn't an issue. Windows machines were not on the Internet. There was no Internet. And so what Microsoft's developers did was they said, wow, wouldn't it be handy if we could just run DLLs that we want to every time a process starts up? Oh, yeah, let's add AppInit to the registry, and it'll do that. Or wouldn't it be cool if we could, you know, we want to be able to debug any program in the system, not one that's, like, asking for us to debug it. So let's just allow a debugger, which is to say any running program in Windows, to reach across into another process and take over its threads. Whoa, wouldn't that be handy.

So, I mean, these things are very handy from a developer standpoint. And they're your worst nightmare when you're trying to contain processes, and you want malware that gets into your system not to be able to really mess things up. The fact is, Windows is insecure, and you might argue insecurable at this point because - and this is what Microsoft fears - if they lock it down too tight, it will break things. And Microsoft never wants to break anything.

Leo: Right, right, right. So are you saying that there's no point in doing a leaktest anymore?

Steve: Well, it's why I have not taken the trouble to update mine, because you...

Leo: You just can't test enough.

Steve: Well, yeah. You get into that virus/antivirus, spyware/antispyware mode where it's like, okay, yes, I could write LeakTest 3 or SuperLeakTest or LeakTestPro or whatever you want to call it that does all these things, does everything I can think of. The problem is there's still going to be other ways to bypass a personal firewall. I mean, it's Spy vs. Spy at that point. It's something in your system, and because of the nature of Windows there really isn't enough protection in the system to prevent malware from just doing what it wants to.

Leo: Right. Very interesting stuff. I guess that - my sense is, if you can't test for leaks, a software-based firewall is kind of essentially worthless.

Steve: Well, we don't want to say that because it's certainly the case that, if you get a really good firewall which has taken the trouble to block many of these known exploits, then certainly it's the case that malware that is using those will be blocked. Hopefully it raises an alert on your screen saying, whoa, something is trying to inject a thread over in Internet Explorer. Here's the process name. You need to go find out what that is. So, I mean, and this, of course, this is what ZoneAlarm did for me when I was using the very first version in beta, was it said, hey, tsadbot.exe wants to use your Internet connection.

Leo: Does sound like, though, that hackers and bad guys are well aware of this, and if they want to they can get around it.

Steve: Yes. Well, I mean, for example, just issuing a DNS query that sends data out as the DNS query.

Leo: That's very clever. You can't block that because DNS queries have to work. I guess you could parse them somehow. But even that wouldn't work because you don't know what a URL's going to look like.

Steve: Yup. And again, they could also easily, you know, add ASCII 20 to every character or rotate the character so they no longer look like, you know, perform a simple encryption of your credit card number, turning it into other letters and characters. And so it would no longer match, I mean, I've seen these firewalls, I just sort of shake my head, that would say, oh, enter in your personal information, and we're going to make sure...

Leo: And we'll look for it.

Steve: Exactly. We're going to look for it on its way out. And it's like, okay, all you have to do is add "1" to every character, and it turns it into gibberish that then passes right through that filter. So, I mean, that doesn't do you any good.

Leo: Although on the other hand, I mean, you could make that argument about any security softwares. Of course, once the software is out there, and bad guys see what it's doing, they can write around it. And so that is that escalating, that vicious circle that you've been talking about.

Steve: Yeah, it really does. And, I mean, again, what you want is you want to prevent stuff from getting into your machine. Now that we've got firewalls up, we've got routers in place, we know that random buffer overflows that have existed and probably still do in all versions of Windows, those are not remotely exploitable because incoming unsolicited packets are dropped.

Now the problem is, as we've said, it's things like scripting exploits, the fact for example that Microsoft ActiveX controls can be instantiated by IE and run code in a way that is unsafe. The way that happens is you've got scripting on your browser, or you visit a malicious site. Apparently it's happening to people all the time. So, you know, I would say the attack surface, as the term is, is much reduced in Windows, so that we're no longer seeing Code Red worms, you know, just literally exploding across the Internet in a few hours. That's not happening.

But hackers are using everything at their disposal to still get into people's machines by having essentially themselves invited into the machines. And once there, I mean, as these links on the show notes page demonstrate, there are ways around outbound through every firewall, which is not to say having a firewall there that's doing the best job it can is still not better protection than none at all. It certainly is. But I don't run one.

Leo: I'll tell you what I run. I run that Astaro Gateway. That's what I run. And I just cross my fingers. But, you know, I have to say, I mean, I haven't been bit. I think a lot of it is - in a long time, I mean, it's been years since I've had even a spyware infestation. I'm sure that's true for you. And I think so a lot of it really is user behavior and how you use your computer, what you do on the Internet.

Steve: Yes, it is absolutely a matter of recognizing and appreciating the dangers, as certainly our listeners do, and moderating your behavior so that you're doing, I mean, that you're acting safe. I have to say, Leo, I mean, there have been situations where I have known I was going to be exposing myself to trouble, but I had to go somewhere unsafe. I did it on essentially an expendable machine. And sure enough, this thing downloaded a bunch of junk in, and I saw services starting up. And, I mean, it was obvious to me that I had just done something really wrong, but I had to go where I needed to go. In this case I used essentially a Typhoid Mary machine that was my disposal image, and then I re-imaged it afterwards, after getting done what I had to get done.

Leo: That's another good thing. Anybody who listens to Security Now! probably should have a operating system image that's clean, known clean, and can be quickly reinstalled. And keep that up to date. That's a very handy tool in this kind of thing.

Steve: Well, of course, and a true, a virtual machine that is...

Leo: That's a great way to go, yeah.

Steve: It's a safe place to play, yes.

Leo: Yeah. It's unfortunate, I mean, it's the people who get bit by this the most are the least sophisticated users, and those are the ones who are buying PCs that don't come with Windows install disks. They don't know about making images. They're the ones that are going to get bit, and they're the ones who are the least prepared for it, frankly.

Steve: Right.

Leo: Anybody who listens to this show I think is probably safe. But there you go. Let me remind you that the place to go for all of your security software, and even LeakTest, should you decide to buy it, or download it for free, I should say, is GRC.com. That's also where you'll find 16KB versions of this show for the bandwidth impaired, Elaine's great transcriptions so you can read along as you listen, and Steve's many free security problems besides LeakTest - Wizmo, Shoot The Messenger, DCOMbobulator, and of course ShieldsUP! That's tested over 40 million PCs. That's an amazing number.

Steve: Yeah. I think we're at - I don't remember now what the number is. I look every so often, so. And again, I just want to say one more final thanks to our listeners for coming to the call and voting for this podcast, Security Now!, as...

Leo: The best technology and science podcast in the world, ladies and gentlemen, and you're listening to it. Isn't that great?

Steve: I love it.

Leo: Nice feeling to get that. Thank you very much to all the voters and to the PodcastAwards.com for giving us that award. I mean, this is, I think, their third or fourth year that they've done this. And it's just a - it's important, I think, to do this, to acknowledge all the hard work people put into their podcasts. Or netcasts, as we call them.

Hey, thanks, Steve. We'll see you next week will be our Mailbag episode. Get your cards and letters in, folks. Thanks for joining us on Security Now!.