SECURITY NOW!

Transcript of Episode #23

GRC's MouseTrap

Description: Leo and Steve close the backdoor on the controversial Windows WMF MetaFile Image code Execution (MICE) vulnerability. They discuss everything that's known about it, separate the facts from the spin, explain exactly which Windows versions are vulnerable and why, and introduce a new piece of GRC freeware - MouseTrap - which determines whether any Windows or Linux/WINE system has 'MICE'.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-023.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-023-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 23 for January 19, 2006: GRC's MouseTrap. Steve Gibson is back, and we are here finally to put a rest to the whole controversy surrounding this WMF exploit whole intentional backdoor, depending on your point of view. Steve, I think you've been in a firestorm of controversy all week.

Steve Gibson: Well, you know, yes. Certainly that's the case. It's like the world ended when I suggested that maybe this was an intentional backdoor. Or I guess the word was "deliberate." And, you know, there are a couple of things that are very hot about, well, exactly those two words. First of all, "backdoor" carries a hugely negative connotation. I mean, the only way we're used to using it is it's installed by a malicious hacker to get into your system remotely. And in fact, you know, you asked me did this mean that somebody could connect into our computer when we talked about this for the first time last week. And I explained, no no no, this would only allow someone going to a website or in one way or another getting your system to render a metafile, could be by receiving email or something, to run arbitrary code on your machine.

Now, the fact is - and, you know we talked about a benign sample or example for how you could use this. But, I mean, the idea of thinking Microsoft is malicious is nuts. I should have been careful to say that that was not what I meant. I mean, I run - we all run - gigabytes of Microsoft's object code, for which we have no source, which we totally trust to varying degrees what it's going to do. I mean, so the idea that Microsoft would have malicious intent is completely ridiculous. It was never what I meant to imply when I said "backdoor." "Backdoor" is just, you know, it's sort of the only term we have to describe this new kind of vulnerability that was discovered, you know, very early in 2006 or the end of 2005.

Leo: There's a longstanding tradition, though, of backdoors in code. In fact, many BIOSes for PCs, for instance, have secret passwords you can use to get in if the administrative password has been set. I mean, this is well known. It's not unprecedented.

Steve: Well, right. And in fact many people wrote to me saying, Steve, you know, programmers often put in their own private backdoors just because, you know, they think the code is theirs; or they have some feeling of ownership about the system they're designing; or, you know, they want to be able to get back in, not recognizing how serious doing something like that can really be.

Now, the second component was my use of the word "deliberate." Now, you'll remember that last week, what I was saying was, I was writing this vulnerability tester for the purpose of understanding what was going on and to ultimately look at the older versions of Windows, which Microsoft said were vulnerable, but they were not going to patch because they weren't vulnerable enough. The feeling I had with just the way the code worked really led me to believe that this was on purpose, that this was not, you know, anything like the traditional bug.

Now, of course, we understand that this is sort of somehow different. One mistake I made, and that was as a consequence of the fact that my test metafile only had a single record, was I talked about key, the fact that the metafile record had to, apparently, in my case had to be set to an incorrect length in order to make this happen, which further increased my suspicion, beyond the fact that, I mean, code was running in the image, which is never supposed to happen anyway. It turns out that that was a consequence of my metafile image having only a single record. The other exploit images - I should have realized this at the time - the other exploit images created by the Metasploit Framework have multiple metafile records and can have the correct length. And even Ilfak's original metafile tester, it used a length that was not one. So, you know, that was dumb. I fully recognize that that was a mistake I made. Microsoft's blog, which came out from the Microsoft Security Research Center the day after, you know, corrected this initially, before I had a chance even to get back in and look at it; but I verified that I was wrong about that.

Leo: Does that destroy your argument, or does it impact the argument?

Steve: Well, I don't think it does. I mean, certainly it takes some of the edge off of it. But when I did finally look at Windows - I mean, and believe me, Leo, I was holding my breath that, you know, as I said, I might end up retracting everything and be completely wrong about this - when I looked at Windows I saw, I mean, as clear an example of intention as I have ever seen. I mean, this was just code designed to do this, code designed to jump into the metafile image and run the code contained in the image.

Leo: Now, Stephen Toulouse, who blogged about this for Microsoft, said, well, sure it's intentional, but it's intentional with a benign point of view. It was to allow GDI functionality; right?

Steve: No. No one ever believed, I mean, no documentation, no common practice, no use ever had metafile images running code. I mean nowhere. I've put together and I've further fleshed out the page that I began last week. It was just sort of a placeholder page. It's at GRC.com/wmf/wmf.htm - WMF, of course, for Windows MetaFile. I've laid the whole thing out. I've got a screenshot and link to Microsoft's original documentation from Windows 3.0 and 3.1 explaining what this whole ABORTPROC thing is, and that it is for executing code in the user's application. I mean, it makes - it's crazy to think that even Microsoft at any time in the past would have thought that it made sense to mix code with drawing commands.

Leo: So the only reason you'd put this in is why?

Steve: The only reason is to run code in an image, which has never been sanctioned, never documented, and, I mean, makes no practical sense.

Leo: There's no other legitimate use of that. It's so that you could put code in an image.

Steve: Well, exactly. And, even more so, when a program runs, the Windows Loader does all kinds of fancy things, fixing up and filling out that IAT that we talked about a long time ago with RootkitRevealer, the Import Address Table, which essentially connects the application into the Windows API. If you're code running in an image, you have no advantage of Windows Loader, which basically makes it feasible for you to talk to the rest of Windows. Ilfak, in his vulnerability tester, because of this had to go through all kinds of very tricky hacker hoops in order to explicitly get access to Windows in order to just pop up his little dialogue that said you are or you are not vulnerable. It was a lot of work.

So, I mean, it just - it doesn't make sense that Microsoft could have ever published the idea of doing this; yet not only did I look at this, at the way this is implemented, but our friend Mark Russinovich from Sysinternals, he looked at it and sent me email, which I have a link to also on our WMF page. He analyzed this and concluded, just as I had, that this was intentional. He was not comfortable saying it was a backdoor. And, I mean, I respect his opinion. You know, "backdoor," as I said, is a very loaded word that carries with it all kinds of, you know, implicit malice, which I never meant to imply. But Mark, looking at the same code I have, and actually several other people, too, recognized that, for whatever reason, this is what the coder intended.

Now, I want to say that separately from what Microsoft intended because, you know, Microsoft isn't one mind. They're not like one person that even thinks coherently. And we see examples of that all the time. So I'm not saying - I have no way to forensically examine or to know what the source code ever looked like, whether there was a comment there that might have said, "This code will allow images to contain executable code." I mean, we'll just not know. Maybe it was a mistake. I mean, anything is possible. But, you know, as I said last week, when I looked more closely at this, the only reasonable conclusion is that for some reason it's intentional.

Leo: All right. So you're standing by that. And we just don't - we'll probably never know why it was put in there. And it's clearly a dangerous thing. And as we've seen, it ended up making a very big security flaw in Windows.

Steve: Well, and, you know, we were talking about Steve Toulouse's blog posting. I also have a link to that, the complete posting. And also I've taken it apart and responded in-line to some of his points. At one point he explains that Windows 9x is not susceptible to this at a critical level because Windows 9x platforms have extra security measures to prevent this from happening.

Leo: More than XP and 2000 and NT?

Steve: Well, yes.

Leo: Well, c'mon.

Steve: So you would - no, that's what Microsoft is saying.

Leo: That's not particularly credible.

Steve: Meaning that these extra protections...

Leo: Were removed.

Steve: ...have been removed, removed from the later versions of Windows.

Leo: Okay.

Steve: You know, and even Vista, you know, the beta of Vista that isn't even shipped yet, Microsoft issued a patch last week, or I guess earlier this week, to fix it because they were shipping it with this same vulnerability.

Leo: Now, I got emails from some security folks who said, well, wait a minute, if WINE has this vulnerability - the Windows emulator under Linux - that means the WINE guys wrote it in. Surely they didn't perpetuate this backdoor. Doesn't that kind of blow your theory to heck?

Steve: Well, okay. Let's talk about WINE. First of all, Ilfak's tester, which in all fairness to him he wrote quickly, and he talked about how, you know, he was only able to write it for the systems he had.

Leo: Right.

Steve: It says that WINE systems are not vulnerable. It's very important for anyone listening to this to get a copy of MouseTrap, which is GRC's official Windows MetaFile vulnerability tester. It shows that WINE systems are vulnerable unless they are brought up to current patch level. So I want to make sure people know about that. We will talk a little bit more about MouseTrap. The reason I've named it MouseTrap is that I was looking for something, you know, less frightening and loaded with negative connotations than "backdoor." And so I said, well, we're talking about Metafile Image Code Execution, MICE. So that's MICE. The question is, does your version of Windows have MICE or not.

Leo: So if WINE has MICE, why would the WINE people put an intentional backdoor in?

Steve: Ah. Well, you know, those guys brag - and, I mean, first of all I've got to say they just do a phenomenal job. It boggles my mind that you can run one of my apps that takes advantage of lots of Windows characteristics under Linux on WINE, and the darn thing even works.

Leo: Yeah.

Steve: So, you know, the idea of emulating the incredibly expansive Windows API is shocking and daunting. I wouldn't have believed it possible. But, you know, they brag bug-for-bug compatibility with Windows. Well, the way you get bug-for-bug compatibility, that is, you know, even duplicating undocumented features, is you don't code to the specification because, of course, Windows bugs are not in the Windows specification. You code to what Windows actually does. The way you determine that is through reverse engineering.

Leo: So they're disassembling everything and duplicating it, the calls, yeah.

Steve: Exactly. So, I mean, and actually this lends credence to my theory because if, you know, you might expect that a rewritten buffer overflow would not be carried into WINE because doing the same sorts of things, you know, there would be enough difference in translation that, you know, a buffer overflow that's clearly a bug, you know, like a defect in the programmer's thinking, it would be reasonable to presume it wouldn't get copied across. But something intentional in Windows, where anyone looking at the code says, oh, this is what I'm supposed to do - I mean, this is what Mark Russinovich, other coders, and myself all saw. Well, the WINE guys saw it, too. They didn't ask...

Leo: So they just said, hey, this is what Microsoft wanted to put in, we're going to put it in. We've got to duplicate it.

Steve: Yes, exactly. They didn't ask why.

Leo: Right.

Steve: Because their admirable goal is compatibility first.

Leo: Ours not to ask why.

Steve: Now, in this case it bit them because they copied this intentional behavior that's been in Windows since NT4.

Leo: Pretty impressive. They actually duplicated one of the worst exploits on Windows of all. So that's a nice job, guys. I mean, but that's what they claim to do, bugs and all.

Steve: Yeah.

Leo: All right.

Steve: Okay, now, older versions of Windows.

Leo: Yeah, let's talk about that because 95, 98, and ME, there's been a lot of question. Are they vulnerable or not?

Steve: Well, that's what got me into all this was, you know, Microsoft said, you know, initially on their vulnerability report all versions of Windows were there. Then they moved off the older legacy versions that were no longer under their, you know, "we'll fix every bug" support level. They moved its designation from "critical" to "important," which essentially had the effect that they did not need to patch it. Unfortunately, it also had the effect of frightening millions of Windows 9x platform users, you know, 95, 98, 98 Second Edition, and Millennium, because they now had the feeling that they were vulnerable to something that could get them that Microsoft had chosen not to fix.

So I looked at this. Okay, here's the whole - here's the absolute truth. None of those early platforms are vulnerable at all. Microsoft says, oh, they had, you know, extra precautions were taken. Okay, mumbo jumbo, I don't know, you know, we don't know really what that...

Leo: Yeah, the extra precaution is we didn't write it into that version.

Steve: Yeah, the extra precautions we forgot later. I've used Microsoft's tools. Microsoft makes symbols available for their kernels that allow their debuggers to show you, as you're stepping around inside of Windows - this is for programmers to figure out why their code is not working. I've gone through. There is absolutely no possibility. I mean none. You have as much possibility of jumping from Earth to the Moon as exploiting this under early versions of Windows. So, I mean, I don't know if...

Leo: You're brave, Steve. I don't know if I'd ever say "never" with hackers these days. But all right, I'll take your word for it.

Steve: Well, I just, I mean, I've literally, I mean, I've sat there and stepped one instruction at a time, trying to find a way, because I wanted to, you know, make my vulnerability tester as effective as possible.

Leo: You know, this raises an interesting issue. Isn't this exactly how these kinds of exploits are found? People just step one by one through all of the different routines in Windows, looking for holes?

Steve: They must be...

Leo: You're basically doing what hackers are doing, trying to reverse engineer the code, looking for exploits.

Steve: Combing, combing - exactly. Combing through the code, you know, looking for anything that they can interpret in a way that's to their advantage.

Leo: Right.

Steve: So, I mean, really none of these old versions of Windows, no one using them has anything to worry about. Now...

Leo: Can you run your MouseTrap on Windows to see?

Steve: Absolutely, all versions of Windows.

Leo: Okay.

Steve: MouseTrap will detect Windows MICE in every version of Windows, including on WINE. I ought to mention also that there were some versions of WINE we encountered during testing earlier this week that did crash when trying to run my app. I mean, again, I'm not surprised. For that reason I created a command line version, just called MouseTrapCmd, which anyone - well, first of all, it runs under Windows in general. But specifically for WINE guys, if they've got a version of WINE which is serving their needs, but for whatever reason they can't run my MouseTrap and they want to check the vulnerability of their current build of WINE, they can use the command line version, also available on my site, in order to do that.

Leo: And just to underscore this, this isn't testing for all vulnerabilities with Windows MetaFile. This is testing for the existing one that we know about, just to see if it exists in the version of Windows you're using.

Steve: Just MICE.

Leo: Just MICE.

Steve: Just testing for MICE. Now, the one version of Windows that was left hanging out in the breeze is NT4. Earlier this week an earlier version of MouseTrap detected NT4 as vulnerable. Microsoft says no, not a critical vulnerability, don't worry about it, don't need to patch it. Well, the reason they're getting away with that is that - and we've talked about this, many people have talked about this - that NT4 may not have a default viewer which would cause it to display a metafile. As we know, it's the act of causing Windows to display a metafile, or even, for example, Google's desktop search indexing the metafile caused the exploit to occur. So NT4 users, I would say, I mean, if it's a server that has no user interaction, nothing is likely to cause it to display a metafile; and, of course, if it's not poking around under strange, dark website, you know, in the seamier sides of the Internet, you probably have nothing to worry about.

I don't want to, again, overhype this and cause people to worry. But I absolutely want anyone using NT4 to know they are vulnerable to this exploit. Microsoft's not going to fix it. Because the machines are so old and there are so many versions of these files, I'm not going to try to tackle that, either, especially since the target of opportunity, again, as I've said, is relatively shallow. And we've got two workable, useful, although, you know, less than ideal patch solutions which are available. Ilfak's original Windows Metafile vulnerability patcher works great on NT4. And the NOD32 guys - I can't remember his whole name, it's Paolo something, I'll have a link to it on our page. So there are two patchers which anyone using NT4 can safely use to suppress this. And when either of those are running, and you run MouseTrap to check, my MouseTrap utility will say, "This system has no MICE." Before running that, "This system has MICE."

Leo: Good. So it works.

Steve: Yeah, works great.

Leo: Great. All right. So to recap for the less technically inclined - including myself, because this is a little hard for us to follow because really you have to have some knowledge of how programmers work and so forth. To recap, it still looks like whatever was done in Windows MetaFiles to make them be able to execute code was done intentionally, isn't a bug, but is an intentional feature.

Steve: Everyone, not just I, everyone who has seen this and looked at the code...

Leo: Mark Russinovich and everybody.

Steve: ...yup, agree that it was intentional.

Leo: No one knows, including you, why Microsoft did it; and no one is averring that Microsoft in some way was doing this maliciously. It's just it's there. We don't know why.

Steve: And no one knows who put it in. I mean...

Leo: It may not be, well, it was Microsoft; but it may not have been a, you know, could have been a renegade employee, I guess.

Steve: Could have not - yeah, exactly. It might not have been officially sanctioned. And, I mean, we certainly hope it wasn't.

Leo: Yeah. It is obviously a major security flaw. Obviously a data file shouldn't be executing code. We don't like that. And but Microsoft has patched it for Windows XP and Windows 2000, and there are two very good patches for it for Windows 95, 98, and ME. And there's even...

Steve: Although, remember, those earlier versions don't need it. You only need it for NT4.

Leo: That's right. So would you recommend not patching it for those earlier versions?

Steve: Oh, yeah, because these patches are actual programs that run on the fly. What I was going to do, if I needed to, was actually modify the Windows code permanently on disk so that it would just no longer do this, so that people would have, essentially, the equivalent of a Microsoft patch that actually fixed the problem rather than suppressing it.

Leo: Right.

Steve: It's important to recognize that both Ilfak's and Paolo's patches, they are not really patches, per se. They are programs you run in Windows to suppress the vulnerability, to keep it from having a chance to happen. They don't remove it. So, yeah, I mean, anyone running now currently either of these patch utilities on their earlier versions of Windows, under the assumption, as Microsoft has stated, that these systems are vulnerable to this, can absolutely safely remove those and stop running that code. So, you know, it is a program you're running. NT4 users will have no choice but to run that for the rest of their lives, if they're worried about this biting them. And, you know, you have to decide how concerned you are about that.

Leo: Right. Anything else you want to say about this?

Steve: I think this closes the chapter, Leo.

Leo: And a lousy chapter it's been. I'm glad we're closed, first of all because it was such a nasty bug. I do hope people have patched it. And gosh only knows how many people were impacted by it and have infections they don't know about and may never know about.

Steve: Well, and two really good things, or three really good things came of it. We have GRC's MouseTrap, which will now be available on our site forever, that will allow anyone at any time to check whether their Windows system has MICE, Metafile Image Code Execution. So we have a new utility, a secure utility. Anyone setting up Windows from the beginning will always have MICE until they bring themselves to Microsoft's current patch level.

Leo: That's right.

Steve: Remember, this will always be there. It's in the code from the beginning, ever since NT4. So that's the first thing that came from this. The second is we've answered the question for all users, I mean, the millions of legacy Windows users. They are, despite what Microsoft claims - I mean, I don't know why Microsoft is saying this. I mean, it's been suggested that maybe they're attempting to scare people into upgrading to versions of Windows that are being currently maintained. Obviously people still running Windows 9x machines have chosen for whatever reason not to. So I want to put to rest forever the issue and the concern that maybe they're vulnerable. They are not vulnerable. And if they're running those dynamic exploit suppression code patches from Ilfak or Paolo, they can remove them.

Leo: They don't need them.

Steve: And the third...

Leo: So Microsoft said - just before you go to the third step - Microsoft did say that they were vulnerable, or they said they didn't know if they were vulnerable.

Steve: No, Microsoft still, even in not only their vulnerability report but in Stephen Toulouse's blog, is claiming these systems are vulnerable.

Leo: Okay, so let's clarify that. They're not.

Steve: They're just not.

Leo: Okay.

Steve: Okay. And the third thing is NT4 users need to recognize that they are in fact vulnerable, that NT4 will run code contained in a metafile, if for whatever reason it ever in the future encounters one. So NT4 users should run, if they're concerned, you know, if they're, like, using NT4 to surf the 'Net, as opposed to sitting in a back room being a server, if there's a chance of it encountering malicious metafiles, you'll want to run one of these dynamic vulnerability suppression utilities.

Leo: Because there's no other patch for it. That's the only way to get rid of it.

Steve: Yup. I'm not going to fix it, and Microsoft's not going to fix it.

Leo: And finally, at a higher level, I think we've learned something, which is there is always this risk in closed source code that stuff's going on that you don't know about. To me it's still a good argument for why open source is a good idea.

Steve: Yeah, I find myself gravitating toward those kinds of solutions. I'll mention here, as we wrap this up, that people have written asking about OpenVPN because, of course, prior to this we were doing a series on VPN solutions. I just wanted to mention, I am in love with OpenVPN. I've got it working and dancing and prancing. I mean, it does everything I want it to. It's a beautiful open source solution which will allow people to do the things, I mean, really closely approaching what we were hoping is the holy grail and in some ways, I think, even surpassing it. So I ask them to be patient. Obviously I've been a little distracted here. It's like, okay, Steve, where did your first three weeks of 2006 go? But I'll be getting back to...

Leo: Deep within the Windows code.

Steve: I'll be getting back to nailing all this down, documenting how OpenVPN works, and producing some how-tos; and we'll be talking about it in the future.

Leo: All right. And next week are we going to go back to our originally scheduled programming?

Steve: Oh, who can predict? We've been wrong the last three weeks.

Leo: Well, it's Mod 4 anyway. So I think normally we would do questions and answers next week.

Steve: Ah, we've got to do questions and answers.

Leo: Yeah. And then after that we are going to get to kind of the underlying concepts behind how the Internet works, which I'm actually very interested in. I think that...

Steve: Well, and many people have written saying that they are. I'm very excited that we're going to be able to lay some foundation that will then allow us to build on cool things, like how crypto works and how to verify security certificates and all kinds of neat technologies.

Leo: Well, this has turned into a college-level course on security. But I'm learning a lot. I hope you all are, too. And I thank you, Steve Gibson, for doing this for us. Of course, more information and the show notes are available at Steve's site, GRC.com/securitynow.htm. You'll also find 16KB versions of the show for the bandwidth-impaired.

We do thank our friends at AOL Radio for providing us the bandwidth and broadcasting the show on their podcast channel at AOLmusic.com. And Steve and I are thinking about making this a stereo podcast so that he can, you know, our voices will be slightly separated. And if that's the case, we'll continue to offer a mono version, a 16KB version, but the stereo version will be on AOL Music. That seems to be the trend. People want higher quality, not lower quality.

Steve: Well, I get a lot of requests, in fact, many people are wanting, you know, the podcasting for the lower bandwidth links. But I agree, I think certainly high quality is the step in the future.

Leo: Yeah. For those with the broadband, it's easy.

Steve: We ought to mention also that, from my side, at least, this podcast is also brought to you by SpinRite, which pays all my bills and is responsible for making it possible for me to screw around and poke inside Windows and reverse engineer VPN systems and all that.

Leo: Yeah.

Steve: So, you know, for 17 years the SpinRite just keeps on going and helping, you know, literally tens of thousands of people whose hard drives are in trouble.

Leo: And rightly so. Absolutely the best hard drive maintenance and recovery utility in the world, SpinRite from GRC.com. Thanks, Steve. We'll talk next week.

Steve: It's always an adventure, Leo.