



# Hansel minutes

Hanselminutes is a weekly audio talk show with noted web developer and technologist Scott Hanselman and hosted by Carl Franklin. Scott discusses utilities and tools, gives practical how-to advice, and discusses ASP.NET or Windows issues and workarounds.

## Text transcript of show # 13

April 12, 2006

### CSI: Your Computer

Scott uses his monacle and razor-sharp logic to find a trojan that was infecting his ASP.NET server. You'll learn how he did it, and probably discover some new tools you didn't know about before.

(Transcription services provided by [PWOP Productions](#))



### Our Sponsors



<http://www.codesmithtools.com/>



<http://dotnet.con-sys.com>



<http://www.peterblum.com>



(Music)

**Lawrence Ryan:** From [Hanselminutes.com](http://Hanselminutes.com), it's 'Hanselminutes', a weekly discussion with web developer and technologist, Scott Hanselman, hosted by Carl Franklin. This is Lawrence Ryan announcing Show #13 recorded Monday, April 10<sup>th</sup>, 2006. Support for Hanselminutes is provided by CodeSmith Tools makers of CodeSmith an extensible template based code generator for .NET. And you can get 100 bucks off with coupon code HM100 just for listening 'Hanselminutes'. Online at [codesmithtools.com](http://codesmithtools.com). Support is also provided by [www.peterblum.com](http://www.peterblum.com), "Start with better control, finish with better sites." Online at [peterblum.com](http://peterblum.com). And by .NET Developer's Journal. "The World's leading .NET developer magazine", online at [www.sys-con.com](http://www.sys-con.com). In this episode, Scott talks about the tools and techniques he used to discover a Trojan had infected his web server.

(Music)

**Carl Franklin:** Hi! This is Carl Franklin and welcome back to Hanselminutes. We're here with Scott Hanselman, of course. Hi! Scott.

**Scott Hanselman:** How are you sir?

**Carl Franklin:** The topic this week CSI your computer.

**Scott Hanselman:** Forensic analysis on your machine

**Carl Franklin:** Quincy for the PC.

**Scott Hanselman:** For our international crowd of course, Jack Klugman the reference to Quincy MD, it was a medical detective show. About a -- he worked in a morgue, he was a medical examiner.

**Carl Franklin:** Yeah, and he always went over his boss' head. Yeah. So this is about tools you can use to diagnose problems with your PC. Not necessarily code problems but just everyday problems.

**Scott Hanselman:** This was a really weird week for me when it came to debugging and odd behavior on my system. We had a ASP.NET worker process running at 100%. For no particular reason at all. We had a Trojan horse kind of evil malware virus...

**Carl Franklin:** Yeah

**Scott Hanselman:** I blogged about the Trojan at [shrinkster.com/dvx](http://shrinkster.com/dvx) and debugging the ASP.NET worker process [shrinkster.com/dvx](http://shrinkster.com/dvx) both of them involved going into the toolbox. And interestingly the toolbox consists more and more of any thing that SysInternals has ever done. I used to have a whole pile of utilities and more and more I just keep coming back to those guys because they get you right down to what's happening underneath your system. I used to say that with .NET everyone said, "Oh, it's taking you to a new layer of abstraction. Right. It's a whole new level of abstraction..."

**Carl Franklin:** Right.

**Scott Hanselman:** I always joked that it's really just managed spackle over the underlying OS. And I say that somebody somewhere has to call load library. You know it doesn't matter whether you are using Java or whether you are using .NET. Somebody had to load that such and such DLL into such and such EXE. And inevitably the level of debugging that I end up getting to is the Ports and Processes, DLL's and which DLL got loaded and kind of level. I think there's kind of two levels of debugging out there. May be you would agree, there is like you know hardware debugger soft dice kind of attached to it. Find out what entry points were relocated and what point in memory simple surfer path text stuff...

**Carl Franklin:** Likes to read the CPU registers for fun.

**Scott Hanselman:** Oh! Yeah. I mean we all learn that in school. There was a time when I could say, "Oh man, Dude he was moving. He was moving zero and AX dude..."

(Laughter)

**Scott Hanselman:** And that's totally and he jumped .You know and he went off in to the weeds." And you know frankly, I think it's difficult to be a really awesome kernel level debugger. And...

**Carl Franklin:** It depends what you are debugging. If you are debugging the kernel, great.

**Scott Hanselman:** Yeah. And you know I have always said be a Swiss Army. Do as much as you can. But I reach my limit when I get into this kind of stuff. And I do not claim to know this like I used to. Back in the day man I could funk with the best of them, you know, Windows 16 bit.



**Carl Franklin:** Yeah

**Scott Hanselman:** But nowadays I defer to the experts. And you know the experts right now really keeps coming back to Mark Russinovich at SysInternals. He is actually speaking at Tech-Ed at a pre-con at [shrinkster.com/dvj](http://shrinkster.com/dvj) and he is going to be doing a pre-con on debugging and fighting mal-ware with advanced detection and removal techniques.

**Carl Franklin:** Wow! Cool

**Scott Hanselman:** Which is going to be rocking sweet. He is going to talk about I am sure his rootkit stuff he did with that Sony CD recently...

**Carl Franklin:** Right

**Scott Hanselman:** And different kinds of malware. So my very cursory write-up of fighting that Trojan at [shrinkster.com/dvx](http://shrinkster.com/dvx) not nearly as interesting as the stuff he is going to talk about. But even though his exploits are awesome, I like reading them and I struggle to understand them all, I still think that there is a basic level of understanding I think that we all can use to debug, you know 80%, 90% what goes wrong in the average .NET developer's life. And...

**Carl Franklin:** Is this knowledge that will transfer over to a non programmer types or you really sort of have to be a geek to get this week's show.

**Scott Hanselman:** Well right there you implied there that a non-programmer type can't be a geek and certainly there a lots of folks that don't program. IT guys that may listen to this podcast and others that are not necessarily .NET programmers but yeah I think that this is a geek person ...

**Carl Franklin:** Okay

**Scott Hanselman:** ...type of a thing. If you like the task manager, if you leave the task manager running all the time you might be a geek, right? And if you sort by CPU, if you sort by virtual memory size you might be a geek.

(Laughter)

**Carl Franklin:** I think there is a comedy routine in there somewhere.

**Scott Hanselman:** Very- very likely.

**Carl Franklin:** Well okay. What's next?

**Scott Hanselman:** So, in the comments of my recent post at [shrinkster.com/dvp](http://shrinkster.com/dvp) a debugger better than I whose name I am cannot quite pronounce looks like Matthijs van der Vleuten made a comment in some of my screen shots that the function that I thought was causing trouble in this particular issue with ASP.NET worker process going to 100%, was not in fact the function I thought it was. He looked at the off set and said, " Well, I don't think the function is that large. And I think what's happening is that you are seeing the nearest exported function and you need to load the symbols." And I had completely forgotten about symbol servers. When you use it tool like process explorer from SysInternals, it is the task manager on steroids you can get that at [shrinkster.com/dvm](http://shrinkster.com/dvm). You can right click on a process and hit properties. And then from the properties dialogue box you see things like the path like where did that process get loaded from, just because you see fu.exe doesn't mean it's the one you think it is. It could have been loaded from a different directory. But you can click on the threads tab and you can see the functions that are doing the work.

**Carl Franklin:** Cool.

**Scott Hanselman:** And it turned out that I was just looking at the kind of the raw information there and not seeing truly what was going on. So, I made some assumptions. So, he pointed out that well you can download the Microsoft debugger tools and you can get details on this at [shrinkster.com/dvw](http://shrinkster.com/dvw). This is a really interesting support article at [Microsoft.com](http://Microsoft.com) and the only support article I have ever seen that includes embedded video. So, it gives you an idea how complicated this is. It's actually a KB article at support at Microsoft with an embedded streaming media demonstration of how to this. This is setting up a symbol server. Microsoft has got a global server at [Microsoft.com](http://Microsoft.com) that you can set up a path and environment variable machine wide in your system and point it to a local folder, you'll see all the details there when you visit the link. So I have set this up and I just made a folder in my system called C:\Symbols so then when a debugger whether it be visual studio, whether it be win debug, win DBG that comes with visual studio tools and DSTK or process explorer itself, decides to get more information about a Microsoft executable that you are trying to debug they will automatically download the symbols.

**Carl Franklin:** Now you -- that means that you have to be plugged into that server I mean that doesn't work for everything right?



**Scott Hanselman:** So this particular symbol server works for Microsoft symbols.

**Carl Franklin:** So anything that comes with Windows anything that's part of the OS will...

**Scott Hanselman:** Will be available there, so I have set that up and I am looking at my symbols folder right now and I see a directory called ASP NET WORKER PROCESS.PDB.

**Carl Franklin:** No kidding that is cool.

**Scott Hanselman:** And inside that is the PDBs for ASP.NET.

**Carl Franklin:** So you actually will be able to pull those up in the visual studio and walk through the code?

**Scott Hanselman:** You can't walk through the code; we are talking about the symbols so this will help you get call stacks where you couldn't see call stacks before.

**Carl Franklin:** Okay.

**Scott Hanselman:** It will give you, you know for the most part not quite a line number but it will tell you where functions are blowing up.

**Carl Franklin:** Okay.

**Scott Hanselman:** I'll post about this in the future.

**Carl Franklin:** I always thought that PDB files contained enough information to reconstruct the source code for .NET applications anyway.

**Scott Hanselman:** They contain information about up to and sometimes not including line numbering, but I am sure one of our listeners will correct me and give me the exact details.

**Carl Franklin:** Okay.

**Scott Hanselman:** I also -I'll put up a before and after screen shot of debugging the ASP.NET worker process with the symbols and without them.

**Carl Franklin:** Sweet.

**Scott Hanselman:** And we will see how much more information you can see.

**Carl Franklin:** Okay.

**Scott Hanselman:** It also will allow you to take like when you get dump file you like a DMP....

**Carl Franklin:** Right.

**Scott Hanselman:** Your system is blue screened you can load that into process explorer debug view and you might be able to get the information about whereabouts that happened. That might be as simple as getting the information about a driver that you couldn't see before.

**Carl Franklin:** Yeah. That's been a big problem for me...

**Scott Hanselman:** Now you can set this up machine wide there's an environment variable called NT symbol path you can learn about it at the URL [shrinkster.com/dvw](http://shrinkster.com/dvw). You can set it up at a solution by a solution basis inside a visual studio, there's debug symbol files search these paths and then enterprises can set up symbol servers for themselves. Like we have a very large application called Voyager here at Corillian and there is a number of different versions. So we have an internal symbol server so that when someone is running a particular version of Voyager they can set there environment variable to our main symbol server and it will find the right versions of the PDB's for that particular version of Voyager.

**Carl Franklin:** Cool. That's very cool.

**Scott Hanselman:** Yeah, there is so much more there than I even claim to understand now that we are in such a mixed unmanaged, managed kind of world but dealing with PDB's like this is a particularly complicated issue and having a symbol server as Mathius pointed out to me would have given me even more information and even though I solved the problem I didn't truly understand what was going on until I started digging into that.

**Carl Franklin:** You know I think one thing this is prompting me to do is to set up the symbol server and just write some code and throw some exceptions and see-- you know just mess around with it grab some, grab a stiff cup of coffee and just start playing with it.

**Scott Hanselman:** You know and another thing about PDB is that it has always caused me a lot of trouble is this issue with PDB's and the GAC, you know loading stuff into the GAC.

**Carl Franklin:** You know let's pause there for just a second and let me ask you, now you are a web developer...



**Scott Hanselman:** Sure.

**Carl Franklin:** And you obviously have talked about putting things in the GAC before I was always under the impression may be it's because it's a more of Windows developer than a web developer but that you know loading things into the GAC can you know just cause more problems I guess because of versioning in stuff and updating.

**Scott Hanselman:** Well, for us it's not about the copying around of the files there is something to be said for having an ex-copy deployment being able to take just entire bin folder of your ASP.NET applications and just dump stuff over there...

**Carl Franklin:** Right.

**Scott Hanselman:** For us it became an issue around interop, we have a bunch of com DLL's that get used and if you just go into visual studio and say add reference and then pick a com DLL, you have got foo DLL and that's a com DLL, and you pick it out of the add reference you will automatically get a interop assembly generated for you. It will automatically run that TLBIMP it will make an interop for you, but if you are doing anything funky with com marshalling of custom types arrays or variants, sometimes you want to write these interop assemblies yourself and if you do that you want to mark them as being what's called as PIA or Primary Interop Assembly.

**Carl Franklin:** Yeah.

**Scott Hanselman:** And when you do that you need to strongly name that and those options become shared because com DLL's are registered system wide.

**Carl Franklin:** And they are shared right.

**Scott Hanselman:** So then the associated .NET interop DLL it's nice to mark that as being shared and those often go into the GAC.

**Carl Franklin:** I guess that makes sense.

**Scott Hanselman:** So I think of the GAC as being the place where the system wide things go...

**Carl Franklin:** Right.

**Scott Hanselman:** Things that are shared amongst many web-- you know websites on a single webserver.

**Carl Franklin:** Yeah.

**Scott Hanselman:** Secret management things like that may be a database access layer that a number of different applications are going to use. Once you get things in the GAC the people feel that the GAC is this strange place like we talked about last week, you can you know subst a drive to the GAC and see that what really going on in there.

**Carl Franklin:** Right.

**Scott Hanselman:** One of the things that people forget though is that if you have the DLL in your local directory and in the GAC it will get loaded out of the local directory and that can be confusing about where things get loaded from and often you want to put the PDB files in the GAC.

**Carl Franklin:** Right, now I may be remembering this wrong but it was my assumption that at least in .NET 1.0 that the loader looks in the GAC first before looking in the local directory and then will look at your configured file to see if you have redirected it but I thought the GAC took precedence over being local.

**Scott Hanselman:** So it's little tricky and you can take a look at a Richard Grimes excellent discussion of GAC, that he has got up at [shrinkster.com/dw0](http://shrinkster.com/dw0) and it depends on how you loaded the assembly. If you have asked for a fully qualified assembly name, is you said I want this assembly with this public key token in this version, it will come out of the GAC.

**Carl Franklin:** Okay.

**Scott Hanselman:** If you asked for it by the short name...

**Carl Franklin:** Right.

**Scott Hanselman:** Then it will come out of the local directory because the GAC doesn't know-- its just saying I want the foo you know assembly.load foo isn't enough information. You have to go step by step.

**Carl Franklin:** That's true you can have multiple versions in the GAC. That's true.

**Scott Hanselman:** He has got to—yeah so which one would it pick right?  
If it finds the local it will pick the local one.

**Carl Franklin:** Okay good point.



**Scott Hanselman:** Now all this is very complicated stuff, fusion debugging is a huge hassle but understanding the fusion debug stuff is actually a big part of known as basic debugging that we are talking about. I would like to keep fusion running all the time on my system. I've got a c:\fusionlogs. Travis Illig has got a great fusion log viewer's settings changer that you can get it at <http://shrinkster.com/dw1> in the pre.NET 2.0 world it's pretty complicated to set this kind of stuff up. He'll edit the registry and what happens is you will get in your fusion log's folder a list of all of the different attempts to load a .NET assembly and those failures, so if you are feeling like a DLL isn't getting loaded to a right version isn't getting loaded, you can go there and check that out but kind of back to the point of SysInternals tools. But kind of back to the point of this SysInternals tools as it relates to what got loaded, we talked in the past about .NET 2.0 applications loading up the wrong version of DLLs.

**Carl Franklin:** Right.

**Scott Hanselman:** When I loaded .NET 2.0 Outlook had some addins, that were .NET managed addins.

**Carl Franklin:** Right.

**Scott Hanselman:** And suddenly I was getting 2.0 loaded instead of 1.0, and things were breaking? I talked about that at [shrinkster.com/dvk](http://shrinkster.com/dvk). Process explorer again is a great tool where you can load up process explorer, look in the lower view, you say view, lower pane view and say, what DLLs are being loaded and from where? And if you take a look at that [shrinkster.com/dvk](http://shrinkster.com/dvk), you can say well I insist that this particular version of the STK, be loaded in memory the CLR rather. We talked about that little bit, yeah last week when we were mentioning shell extension only one version of the CLR process.

**Carl Franklin:** Right.

**Scott Hanselman:** So what's going to happen in the shell extensions world when there is two-three versions of the CLR out there, it's become a real problem.

**Carl Franklin:** Ok.

**Scott Hanselman:** File both FileMon and RegMon and TCPView those kind of three together, I am finding more and more useful

when I am having a situation where, what exactly is this application doing? Is it writing to the disc, is it looking in the registry? And both FileMon and RegMon at [shrinkster.com/dvg](http://shrinkster.com/dvg) and [shrinkster.com/dvt](http://shrinkster.com/dvt) respectively, will allow filtering so I can say, I want to see what the carlfranklin.exe is doing. And both the registry and to the file system at run time. So well, this might not be something you'd want to use to debug, DLLs are getting loaded although they would show up as being accessed on the file system. It will show you, is this application looking for a particular a key in the registry? And last week when I was having a heck of a time debugging my, my thinkpad's wireless card, I had the feeling that there was an application that was going into similar wireless registry settings and changing them. And I used RegMon to take a look at that, if you...

**Carl Franklin:** You had a feeling?

**Scott Hanselman:** Don't you just have a gut words like well where could it be getting this information? This is the thing I was talking about, a lot of this kind of root cause analysis when you, when you not interested in going at that really deep level of debugging like, that Russinovich level of debugging.

**Carl Franklin:** Yeah.

**Scott Hanselman:** Is, there is only so many places that these things can be finding their information from.

**Carl Franklin:** Right.

**Scott Hanselman:** Sometimes I feel like people are debugging an application and they go, and then a miracle happens and you know and it's like how many (voice overlap)

**Carl Franklin:** The Gary Larson Methodology.

**Scott Hanselman:** Right exactly. Someone says well, I click here and then all this stuff that I am completely glossing over happens. We talked about that – talking about debugging the XML serializer.

**Carl Franklin:** Correct, yeah.

**Scott Hanselman:** Its-- Windows is not that much of a black box. You can see inside the processes, you can see what threads are running in and what DLLs and sort what process. You can see with simple servers a little bit more inside what is going on. There are, there are debuggers who provide you this value but even the most



basic stuff like, like this Trojan issue I had, where I needed to find out which process was talking to this particular server? I found that there was a Win logon process had a DLL loaded inside it and it was trying to send email from a Russian website.

**Carl Franklin:** No kidding.

**Scott Hanselman:** Well we had Ethereal, the open source packet sniffer on the wire, we could see that someone is trying to talk SMTP off to this Russian machine, but the guys that were doing this debugging forgot the you want to resolve that connection to a particular executable. So we just followed the path we said TCPView from again TCPView is [shrinkster.com/dvs](http://shrinkster.com/dvs), you can say resolve endpoints we could see that some mail.ru was being opened by Win logon. We said well Win logon that seems like a reasonable thing, it does not sound evil. We know we need Win logon that's the NT login manager. So we loaded up process explorer, we sort it by company name in this particular case somebody said wow, that's not you know, that company name. It was a random eight character generated DLL, ok that doesn't look right.

**Carl Franklin:** No.

**Scott Hanselman:** So when we went into the Windows folder found that DLL, we saw that it had been updated a few days before, so that DLL has appeared out of nowhere a couple of days ago. We looked inside the thing with the hex-viewer, we didn't see any nasty strings. But it clearly didn't look right. Searched for it on google didn't find any which meant eight character random generated thing.

**Carl Franklin:** Right.

**Scott Hanselman:** Did you search from google for eight generated, eight character generated, you know DLL, Win logon and found a bunch of Mal-ware applications that like to do that. So how does this DLL get loaded in memory? Well, where else could you look, so we look in the registry. We look in the registry we search for the name of that DLL we can see in the registry under the Win logon section deep in the bowels there is a place where all the different things that want to load in to Win logon put themselves. We could have seen that had we used autoruns the tool that lets you see all the things that happen at start up that SysInternals'. I just poked around the registry, removed that-- the whole thing is listed up on my blog. The point is though that you have to be able to follow that you know...

**Carl Franklin:** Follow the path.

**Scott Hanselman:** Follow the path, from the connection that was open, back to the process, and the process to the DLL because there are places that you can hide things. People have said my blog and they have informed me that that, you know that there are, now kernel mode, evil malware applications that don't show up in task manager at all these don't show up inside of things like process explorer and I think that we'll see at the Tech-Ed presentation that Mark will talk a lot about that kind of stuff. But for the most part it's hard to hide stuff.

**Carl Franklin:** Yeah.

**Scott Hanselman:** If you know, where to look you know?

**Carl Franklin:** Vista is going to make that much easier of course with security being enforced at the kernel level.

**Scott Hanselman:** Yeah, theoretically although I have put in the recent builds of Vista it constantly prompts you, is it ok to do this? Is it okay to do that? You are running a control panel so are you are ok with that? We found a really great website a lot of people recommended, a woman name Tess who works for Microsoft at [shrinkster.com/dvg](http://shrinkster.com/dvg), she is an expert debugger and she works for the PSS, Product Support Services group at Microsoft and she posts the most interesting got this crazy dump file and got -- and found out that it was this you know it was the garbage collector or it was the block thread or whatever, [shrinkster.com/dvg](http://shrinkster.com/dvg).

**Carl Franklin:** I love the name of her blog, "if broken it is, fix it you should."

**Scott Hanselman:** Yeah.

**Carl Franklin:** That's great.

**Scott Hanselman:** I mean a professional debugger it's -- definitely a kind of tails from the debugger's side and I am enjoying that and I am in a need to take a debugger class because last time I did this kind of lower level debugging was really back in the SoftIce just when we were switching into protective mode, so I am out of practice.

**Carl Franklin:** Wow!

**Scott Hanselman:** There is this-- a number of tools that you can use SysInternals are just kind



of some of them. The same thing applies to ASP.Net kind of level debugging we've talked before about tools like Fidler, [shrikster.com/dvf](http://shrikster.com/dvf), and IEAHTTP headers at [shrinkster.com/dve](http://shrinkster.com/dve). These are tools that let us see what is going on at the HTTP level. Frankly I find that ASP.Net kind of debugging to be the easiest because you can really see what is going on, and I mean there is no secret discussion happening between your browser and the server.

**Carl Franklin:** Right.

**Scott Hanselman:** It's all right there; whether it be AJAX or cookies you can see all that traffic.

**Carl Franklin:** So is there an end to your Trojan Horse story or you know how did it-- how did it end?

**Scott Hanselman:** Oh we dug on a little bit more and there also was a browser helper object like a tool bar that was hiding inside of IE and it would go back and forth with this Win logon and apparently it was emailing itself around so the win logon stuff was the bit of the Trojan that would email itself to other people, via these Russian mail relayers and then the browser helper object would listen to IE and occasionally pop up porn ads while surfing. We think it was a Trojan.Vundo, variant but not quite sure. I tried a number of different malware tools and no one would detect it so, there probably is one that would detect it but I think that it just was some variant that this stuff, the new stuff hadn't figured out.

**Carl Franklin:** okay

**Scott Hanselman:** A number of good comments happened though on that particular post. Folks recommended tools like spywaredoctor to try to look at it but you know you can't even trust the spyware tools anymore.

**Carl Franklin:** Yeah

**Scott Hanselman:** Because sometimes they themselves are spyware. Myself, I use spybot. I trust them.

**Carl Franklin:** Yep

**Scott Hanselman:** Interestingly Windows defender, I don't think found this one and I'm not quite sure what I'm thinking about the windows defender tool quite yet.

**Carl Franklin:** So it that Microsoft anti-spyware, the sequel?

**Scott Hanselman:** Right. Windows defender is the next generation of Windows anti-spyware, the stuff that they bought from the giant cooperation and windows defender will be also built into Vista.

**Carl Franklin:** OK good.

**Scott Hanselman:** And actually speaking of defending type things, you know that the Google toolbar was revved this week.

**Carl Franklin:** No I didn't know that.

**Scott Hanselman:** Interestingly if you go to tools extensions and say 'find updates' for some reason it's not showing up yet as an official update on Fire Fox but if you go to [toolbar.google.com](http://toolbar.google.com) or [shrinkster.com/dvl](http://shrinkster.com/dvl) it includes anti-phishing. Which is very cool cause I've got my parents using Fire Fox and I'm sure all of the listeners have their family and friends using Fire Fox.

**Carl Franklin:** Yeah I do.

**Scott Hanselman:** But I still worry about fishing and this will actually bring up a nice fat pop up saying that you know I am suspicious of this website think twice before giving your pay pal password away. And you had seen something that was it Project IP.com?

**Carl Franklin:** Yeah Project IP.com, this isn't really phishing or anything, its just open my eyes to something we knew was there but it can be exploited. This is the clipboard. It can be read; the default setting in IE6 is that a server can read the contents of your clipboard. And usually you know you don't really think about that to much but a lot of people copy their credit card numbers from some file they have somewhere safe and sound on their computer and then paste it into you know amazon.com or wherever they are ordering stuff from when they need to order. I'm not one of those people who has my credit card memorized so, and you know if you do that and then you don't immediately copy something else into the clipboard there's a chance that whoever's looking at your-- whoever's processing your order can or something else-- not just that, cross site script for example some java script code could read the contents of your clipboard. So you know you go to another site after that its still in your clipboard, they can read it.

**Scott Hanselman:** Yeah, that's definitely something to watch out for and actually speaking of security you were talking about some people keep information in text files, maybe they



shouldn't. One thing you might want to take a look at is if you are running a tool like Windows search, desktop search or Google desktop. Type in your social security number in your desktop search. Type in your credit card number, see if that exists somewhere on your system that you may have forgotten about.

**Carl Franklin:** That's a great idea.

**Scott Hanselman:** I found our social security numbers in a PDF that I had created of our taxes, years ago and forgotten about, and then used the search tool to find that bit of text and basically wiped our system of social security numbers, credit card numbers things like that.

**Carl Franklin:** Great idea if you can upgrade a computer and toss out a hard drive, right?

**Scott Hanselman:** Well if you're going to toss out a hard drive you need to burn the hard drive.

**Carl Franklin:** Yeah, hardly nobody does that of course and I have all my old hard drives and I don't know what to do with them but other than subjecting them to severe electromagnetic impulses I'm not sure what, how to do that. So, I mean there are companies out there that make these hard drive shredders we talked about them on Monday sort of as a gag because you know there are \$40-\$50 thousand dollars. Some of them are just nothing more than magnets but there are also things-- machines that will turn your hard drives into a pile of gravel.

**Scott Hanselman:** Well if your going to be donating your hard drive or giving it to a friend or if you've just got hard drives lying around, now that its so easy to plug a hard drive into a system or plug it into a USP disk temporarily, there's a tool called Darik's Boot and Nuke' at shrinkster.com/dw3. You can get a floppy version for older systems and a CDR version for newer ones. It's very dangerous but a self-contained boot system that wipes the hard drive of your machine.

**Carl Franklin:** Wow! That's great.

**Scott Hanselman:** So, you just boot off of it and there's even one for the Mac. It will do SATA drives, if your tossing a disk just make one of these CD's, label it very clearly because you would hate someone boot off this and make a mistake.

**Carl Franklin:** Well now, does this go down as far as the sectors and remove the binary information from the sectors of the drive because

I know that's the -- you can use on track software that kind of stuff to retrieve files that have been lost, by reading the underlying data.

**Scott Hanselman:** If you could take a look at the FAQ on Darik's Boot and Nuke site, he talks about secured deletion of data from magnetic memory.

**Carl Franklin:** OK

**Scott Hanselman:** Talking about what they have to do to flip the bits the different methods. You can read the FAQ and decide whether you think that this is...

**Carl Franklin:** OK

**Scott Hanselman:** Something you'd want to do.

**Carl Franklin:** So he's doing some smart things messing around with a fat table and...

**Scott Hanselman:** No he's wiping it he's actually going alright lets lay down one's and let's lay down zero's.

**Carl Franklin:** OK, cool.

**Scott Hanselman:** We'll do it a hundred times but he says in there are you sure it works great. No, but it is very well thought of and is certainly cheaper than buying one.

**Carl Franklin:** OK, good deal. You know I think we may have a offshoot of this maybe with Camtasia called 'Stumped Scott'. Where you could sort of remote assist into somebody's computer and try to uncover...

**Scott Hanselman:**-It might be fun, we could do live debugging of someone's problem.

**Carl Franklin:** Right, try to uncover a problem and see how long it takes you and if you can be stumped.

**Scott Hanselman:** Well I'm very sure I can be stumped.

I'm only as smart as the sum of my blog readers and I consistently am impressed with the level of folks that are goodly enough to read my blog and I consistently learn from the commenters particularly whoever comments first. You know whoever is the first guy is to stand up and go, "BS, it works like this."

**Carl Franklin:** Yes, right.



**Scott Hanselman:** And I appreciate that just like in this debugging issue, I had completely forgotten about symbol servers and was -- immediately configured my system. That will be the first thing I'll do now when I start using a dev-system, is get that symbol server started up.

**Carl Franklin:** Absolutely, your ego is not your friend folks. Destroy it.

**Scott Hanselman:** Ego less is programming is the way to go.

**Carl Franklin:** Exactly, all right until next week Scott. Thanks for sharing your thoughts and your expertise. We'll see you next week on Hanselminutes.

(Music)