Out of sight, out of mind with Windows Long File Names

zerosalarium.com/2025/08/pentest-trick-out-of-sight-out-of-mind-long-filename.html

Zero Salarium August 9, 2025

Pentest Trick: Out of sight, out of mind with Windows Long File Names

I. INTRODUCTION

One of the very important issues that red teamers and pentester always have to consider is how to keep their payloads low profile. You cannot carry out activities if your payloads are constantly being caught and blocked by antivirus software. In addition to avoiding the watchful eyes of AVs, keeping your payloads from being submitted to analysis sites on the Internet (and subsequently falling into the hands of malware analysts, blue teams, etc.) is also crucial.

In this article, I will present the idea of leveraging the long file name feature on Windows to help keep payloads from being collected by EDR's tools and scripts that gather samples.

Follow me on X to get the latest pentest and red team tricks that I've been researching: Two Seven One Three (@TwoSevenOneT) / X.

II. MAIN SECTION

1. Some basic information about Windows file names

Windows file names are designated in two styles: long and short.

Short filenames, also known as **8.3 filenames**, are used for older tools, DOS-based scripts, and certain installer packages.

Currently, we are using long filenames.

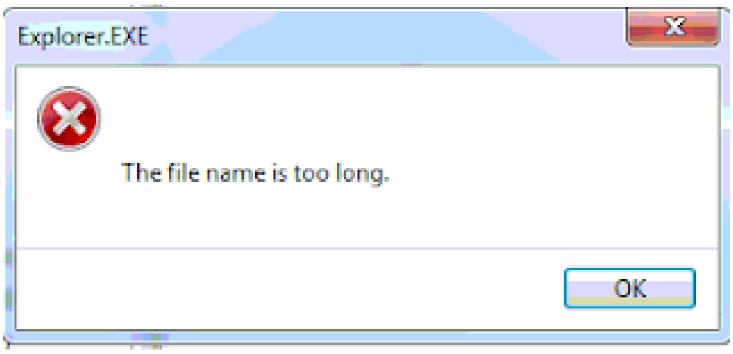
They can include spaces, mixed-case letters, Unicode characters, and most symbols, except reserved ones like <, >, :, ", /, \, |, ?, and *.

Long filenames have a maximum length of 260 characters, are case-insensitive, and support most Unicode characters, spaces, and many symbols. This includes the drive letter, colon, backslashes, and the terminating null character.

Filename Only: The maximum length for just the filename (e.g., mydocument.txt) is typically 255 characters, assuming it's in the root directory.

2. Curiosity about Process Explorer

Have you ever wondered what would happen if the file name you set exceeds the limit of 260 characters? Or have you encountered a warning like the one below?



Of course, Windows handles extremely long paths (beyond 260 characters) using the "**\\?**" prefix or group policy settings to lift the **MAX_PATH** (260 characters) limitation.

I wonder how EDR's monitoring tools will parse the file name if I use a file with a name length exceeding 260 characters.

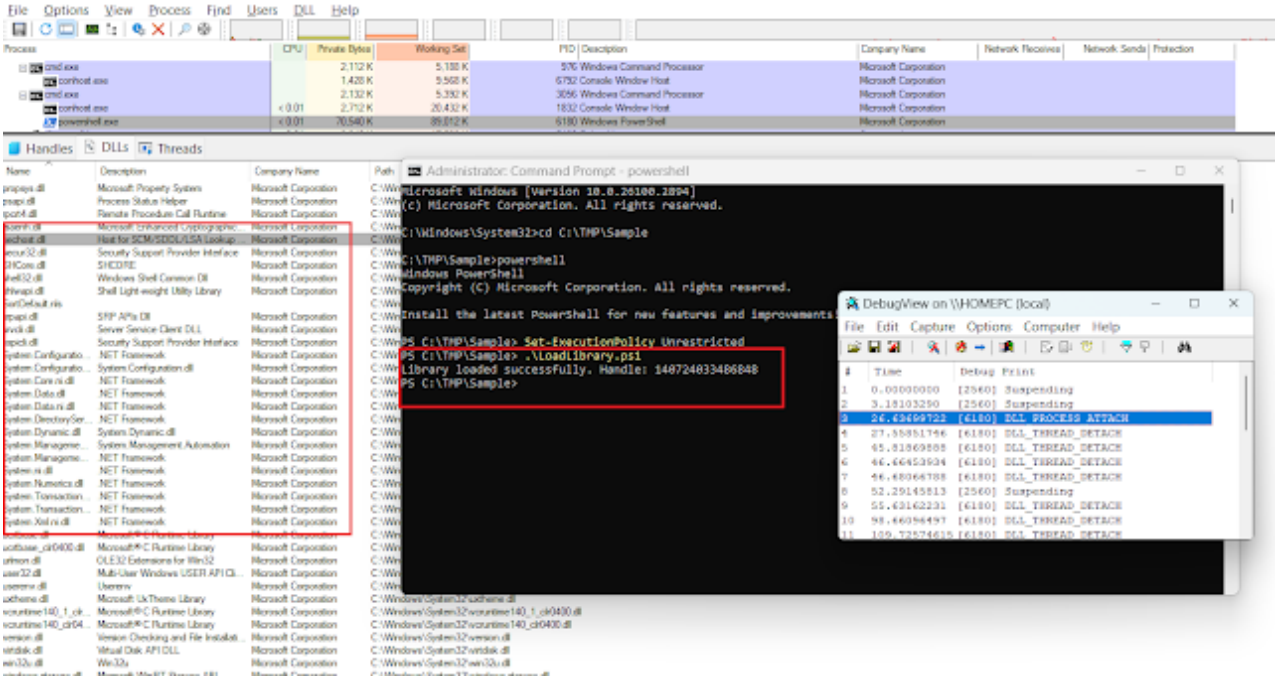
For instance, let's consider **Process Explorer**. I used the simple PowerShell script below to call the **LoadLibrary** API and load my DLL into the process.

```
Add-Type @"
using System;
using System.Runtime.InteropServices;
public class NativeMethods
{
    [DllImport("kernel32.dll", SetLastError = true)]
    public static extern IntPtr LoadLibrary(string lpFileName);
}
"@
$libraryPath = "\\?\Long-Path\SampleDLL.dll"
$handle = [NativeMethods]::LoadLibrary($libraryPath)
if ($handle -eq [IntPtr]::Zero) {
    $error = [System.Runtime.InteropServices.Marshal]::GetLastWin32Error()
    Write-Host "Failed to load library. Error code: $error"
} else {
    Write-Host "Library loaded successfully. Handle: $handle"
}
```

My DLL is located at the path:

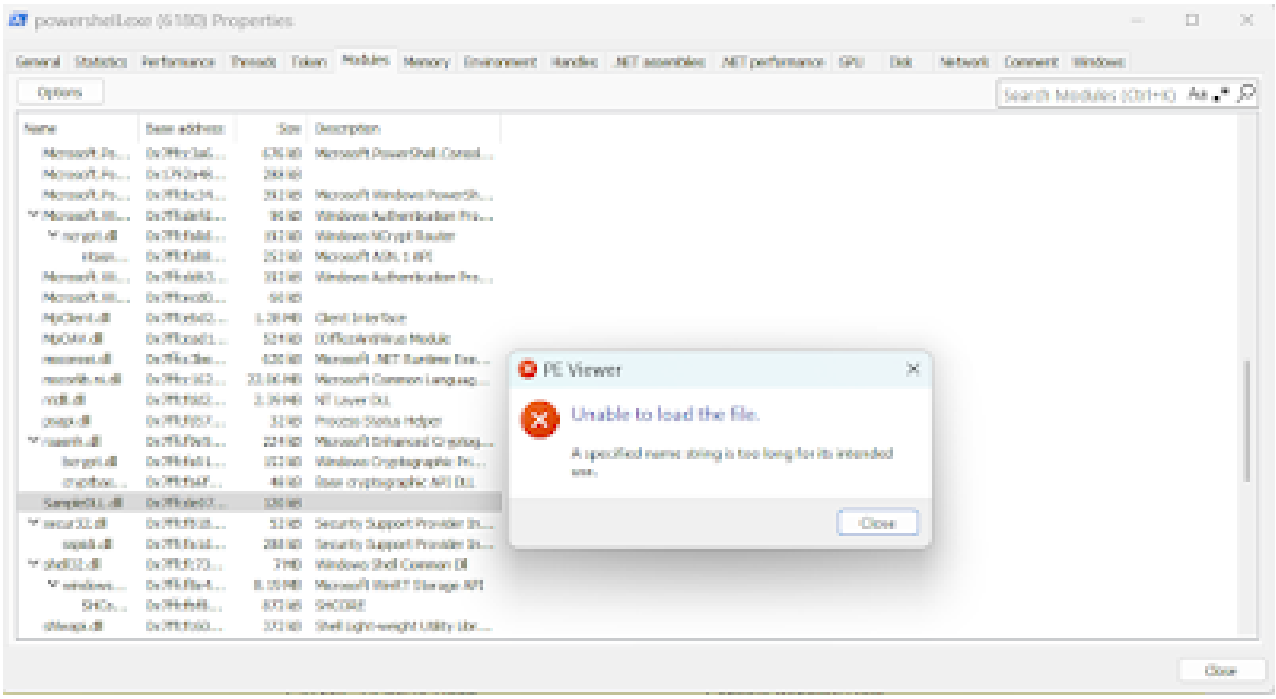
```
"\\?
\C:\TMP\Sample\AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB\CCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCC\SampleDLL.dll"
```

With a length exceeding the 260-character limit, when using the Process Explorer Lower Pane to list the DLLs loaded into the process, my DLL will not appear in the list.



It may be because, in the code that executes the listing of loaded modules in Process Explorer, the programmer used an array initialized with the constant MAX_PATH, which cannot handle content larger than that. This is just my speculation based on my programming experience; if you want to know more accurately, Ghidra (https://github.com/NationalSecurityAgency/ghidra) can assist you.

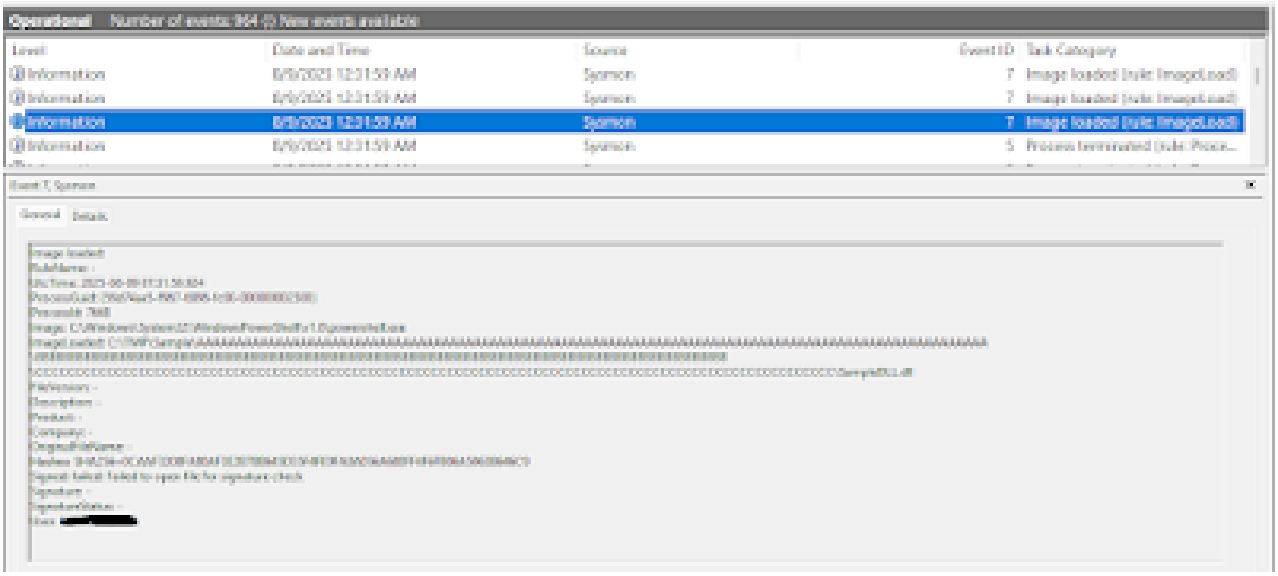
Let's try checking with System Informer:



At this point, System Informer can list the loaded DLLs, but when you double-click to view the details, it automatically removes the "**\\?**" prefix of path, preventing the PE Viewer tool from accessing the file.

So, if EDR solutions tools monitor and log the name of my DLL, do they retain the record with the original name or "beautify" the file name by removing the prefix?

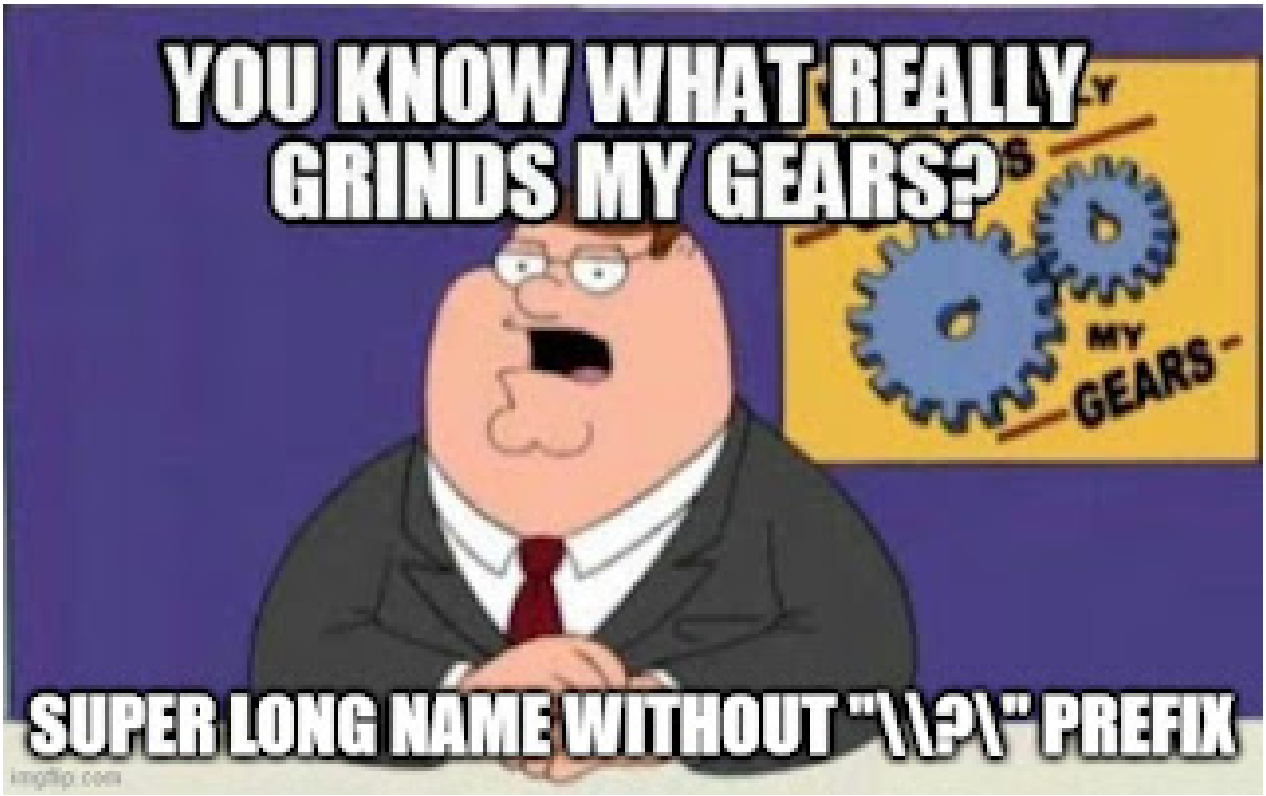
Experimenting with Sysmon:



As expected, Sysmon has removed the "\\?\" prefix from the path to the file.

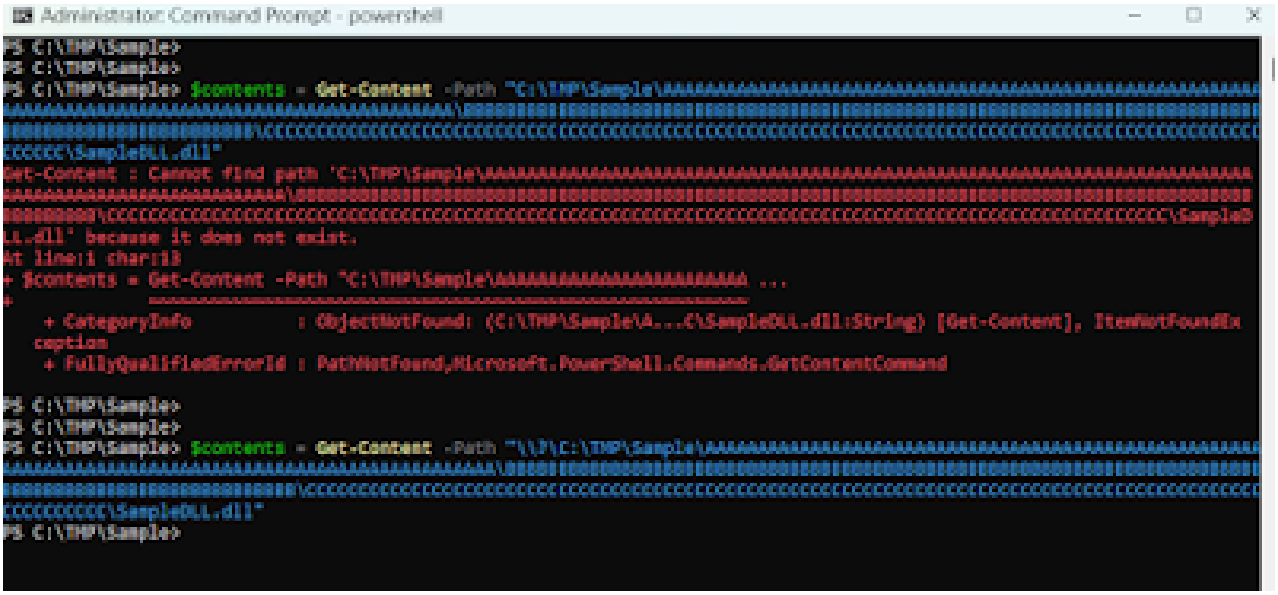
3. How to leverage long filenames to evade detection by security tools

Assuming an EDR uses a specific log parsing set to analyze results from Sysmon, which automatically or manually retrieves suspicious files from clients for further analysis, or simply uploads files to VirusTotal, what issues may arise when the path has been "beautified"?



With the DLL example I mentioned earlier, if a script is used to collect files back to the EDR's server, the script will fail with the message "**... does not exist.**" Based on this result, there is a high likelihood that you will avoid collected by the EDR and prevent analysis of the payload sample being used.

Because at this moment, the access result to the file returned is non-existent. The SysAdmin may think that the file has been deleted or moved to another location. This misunderstanding is even more likely to occur when the tool/script interacts with multiple clients simultaneously.



Of course, the paths you create should appear somewhat natural, for example:

```
C:\Users\Doe\AppData\Local\Packages\Microsoft.ApplicationCompatibilityEnhancements_8wekyb3d8bbwe\Settings\RoamingState\ConfigurationPlugin\ ...\Payload.dll
```

III. CLOSURE

Windows simultaneously uses two types of names: Short and Long Filenames.

The purpose of the Short Filename is to maintain compatibility with older operating system components.

The file names we commonly use on Windows are Long Filenames, with a maximum length of 260 characters.

If you want to use a name longer than 260 characters, you can prepend the prefix “\\?\” to the beginning of the path.

For pentesters and red teamers, if you hide your payload in paths longer than 260 characters, the likelihood is that collection tools and scripts will be unable to access it, resulting in a "file does not exist" message. This is because these tools typically do not automatically add the “\\?\” prefix to the path.

For SysAdmins and blue teamers, it is advisable to check the length of the filename before accessing it and to add the bypass prefix when the filename exceeds **MAX_PATH**. This will ensure that no files are missed during collection.

Author of the article: Two Seven One Three