



RARLab's WinRAR Local Stack Overflow

04-July-2006

Summary

WinRAR is an archive manager that supports a diverse range of formats. This list of formats includes RAR, ZIP, CAB, ARJ, LZH, TAR, GZ, BZ2, ACE, UUE, JAR, ISO, 7Zip, and Z.

While processing LHA files, WinRAR concatenates the directory-name and the filename portions of an archive. Due to a lack of constraints while copying data, two stack-buffer overflows can result.

Impact

These vulnerabilities are present by default in WinRAR. An attacker would need to convince a WinRAR user to open a specially crafted file. This file can have any extension as long as WinRAR is configured to process it. Successful exploitation of these vulnerabilities results in code execution with the full privileges of the current user. Since these exploits are stack based, and due to specific code constructs, exploitation can be made reliable.

Affected software

WinRAR – At least versions less-than 3.60 beta 7 and greater-than 3.0, although others may be affected as well

Credit

These vulnerabilities were researched by Ryan Smith.

Contact

advisories@hustlelabs.com





The code in the following image is responsible for processing LHA extended-headers of the directory name type. The code will copy a user-supplied buffer up to 0x3FC bytes in length into the szDName variable.

```
016B2B95 141C Digest_DNAME: cmp     esi, 400h      ; CODE XREF: Process
016B2B96 141C          jl      short loc_16B2BA2
016B2B97 141C          mov    esi, 3FFh
016B2BA2 141C loc_16B2BA2: xor    eax, eax      ; CODE XREF: Process
016B2BA3 141C          jmp    short loc_16B2BB7
016B2BA6 141C loc_16B2BA6: mov    edx, [edi]    ; CODE XREF: Process
016B2BA7 141C          mov    cl, [edx]
016B2BA8 141C          and    cl, 0FFh
016B2BA9 141C          mov    [esp+eax+140Ch+szDName], cl
016B2BAE 141C          inc   dword ptr [edi]
016B2BAF 141C          inc   eax
016B2BB7 141C loc_16B2BB7: lea   edx, [esi-3]  ; CODE XREF: Process
016B2BB8 141C          cmp   eax, edx
016B2BB9 141C          jl    short loc_16B2BA6
016B2BBA 141C          mov   [esp+esi+140Ch+ucaFileData+0FFCh], 0
016B2BBB 141C          lea   eax, [esp+140Ch+szDName] ; string
016B2BBC 141C          mov   dl, '\0'      ; value
016B2BBD 141C          call ReplaceChars
016B2BBE 141C          add   esi, 0FFFFFFDh
016B2BBF 141C          mov   [esp+140Ch+szDName], esi
016B2BC0 141C          jmp   short try_next_hdr
```

The next image is the vulnerable portion of the code. The program takes the two user-supplied values, one up to 0x3FC bytes in length, the other up to 0xFC bytes in length, and concatenates them into a buffer that is 0x400 bytes in length. The code then copies the resultant buffer, up to 0x4F8 bytes in length, to the szFileName buffer that is only 0xFF bytes in length. This buffer mismanagement results in two stack based overflows.

```
016B2C41 141C          lea   esi, [ebx+Fi_LzHdr.szFileName]
016B2C44 141C          push  esi           ; src
016B2C45 1420          lea   eax, [esp+1410h+szDName]
016B2C4C 1420          push  eax           ; dest
016B2C4D 1424          call  _strcat
016B2C52 1424          add   esp, 8
016B2C55 141C          lea   edx, [esp+140Ch+szDName]
016B2C5C 141C          push  edx           ; src
016B2C5D 1420          push  esi           ; dest
016B2C5E 1424          call  _strcpy
016B2C63 1424          add   esp, 8
016B2C66 141C          add   ebp, [esp+140Ch+szDName]
```





Remediation

The code should either truncate the strings, or allocate more space for the strings.

Version 3.6 Beta 7 corrects the issue mentioned in this document. This version should be installed in order to mitigate the vulnerability. If the old version must be used, it may be possible to copy the lzh.fmt file from the new installer into the current directory. As well, if LHA compression is not needed, the file lzh.fmt may be removed from the installation directory.



Timeline of Events

- 04-July-2006 – Advisory draft date
- 11-July-2006 – Vendor notification
- 12-July-2006 – Vendor created a patch
- 13-July-2006 – Vendor released patched version
- 18-July-2006 – Advisory made public



Attributions

The images of The Muppet Show's Beaker and Dr. Bunsen were taken from <http://www.getbert.com>, <http://www.forskning.no> and <http://newsimg.bbc.co.uk>.

Code and cross-reference screenshots captured using IDA (<http://www.datarescue.com>).

Flawed code obtained from RARLabs (<http://www.rarlab.com>).

The Creative Commons license-notification image borrowed from <http://www.creativecommons.org>.

License

This work is licensed under the Creative Commons Attribution 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Attribution should be provided both in the form of a link or reference to <http://www.hustlelabs.com> and a copy of the researchers' names listed under the *Credit* section of this document.

All other trademarks and copyrights referenced in this document are the property of their respective owners.

