

[UNIX] Zip Long Path Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0029.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/09/04

To: list@securiteam.com

Date: 9 Nov 2004 19:12:00 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Zip Long Path Buffer Overflow

SUMMARY

<<http://www.info-zip.org/Zip.html>> Zip is a compression and file packaging/archive utility. Although highly compatible both with PKWARE's PKZIP and PKUNZIP utilities for MS-DOS and with Info-ZIP's own UnZip. Zip is installed by default with many Linux distributions.

A vulnerability in Zip allows a malicious attacker to run arbitrary code with a specially crafted zip file.

DETAILS

Vulnerable Systems:

* Zip Version 2.3 which comes as "zip" package with Debian Linux.
Possibly all earlier Info-Zip versions are vulnerable.

When zip performs recursive folder compression, it does not check for the length of resulting path. If the path is too long, a buffer overflow occurs leading to stack corruption and segmentation fault. It is possible to exploit this vulnerability by embedding a shellcode in directory or file name. While the issue is not of primary concern for regular users, it can be critical for environments where zip archives are re-compressed automatically using Info-Zip application.

Securiteam: [UNIX] Zip Long Path Buffer Overflow

Example:

The issue can be reproduced by following these steps:

1. Create an 8-level directory structure, where each directory name is 256 characters long (we used 256 'a' characters).
2. run "zip -r file.zip *". The application will crash with "segmentation fault"
3. run "gdb -core core `which zip`" (assuming core dump is enabled)
4. type "where" and hit Enter. Here is what you'll see:

Program terminated with signal 11, Segmentation fault.

[garbage truncated]

#0 0x0805108e in error ()

#1 0x61616161 in ?? ()

#2 0x61616161 in ?? ()

#3 0x61616161 in ?? ()

ADDITIONAL INFORMATION

The information has been provided by <mailto:vuln@hexview.com> HexView.

The original article can be found at:

<<http://www.hexview.com/docs/20041103-1.txt>>

<http://www.hexview.com/docs/20041103-1.txt>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.