Sec Bug #79797 Use of freed hash key in the phar_parse_zipfile function

Submitted: 2020-07-06 03:38 UTC **Modified:** 2020-08-03 08:08 UTC
From: grigoritchy at gmail dot com **Assigned:** stas (profile)
Status: Closed **Package:** PHAR related
PHP Version: 7.2.32 **OS:** Linux
Private report: No **CVE-ID:** 2020-7068

[2020-07-06 03:38 UTC] grigoritchy at gmail dot com

```
Description:
------------
he phar_parse_zipfile function had use-after-free vulnerability because of mishandling of the actual_alias variable.

----- ext/phar/zip.c -----
int phar_parse_zipfile(php_stream *fp, char *fname, size_t fname_len, char *alias, size_t alias_len,
phar_archive_data** pphar, char **error) /* {{{ */
{
        ...

        mydata->alias = entry.is_persistent ? pestrndup(actual_alias, mydata->alias_len, 1) : actual_alias;

        if (entry.is_persistent) {
                efree(actual_alias);
        }

        zend_hash_str_add_ptr(&(PHAR_G(phar_alias_map)), actual_alias, mydata->alias_len, mydata);

        ...
---------------------------

`actual_alias` variable is allocated by estrndup function, which string is part of data of the zip file.

The above code snippet `mydata->alias` is assigned by `pestrndup(actual_alias, mydata->alias_len, 1)` if
entry.is_persistent is true. Or `mydata->alias` is assigned by `actual_alias` variable.
And if `entry.is_persistent` is true, `actual_alias` variable is freed by invoke efree function. `actual_alias`
variable is used invoke of zend_hash_str_add_ptr function as 2nd argument.

Problem is that `actual_alias` variable is freed if `entry.is_persistent` is true, the key of `phar_alias_map` will
use freed memory. `entry.is_persistent` is true if `phar.cache_list` fields is defined in php.ini file.

So if `phar.cache_list` is defined with target phar path so that `entry.is_persistent` is true, then it can be that
`phar_alias_map` hash key would use sensitive freed memory data such as heap addresses that addresses set via linked
list after invoke the efree function. You can see details debugging information at Actual result.


Test script:
---------------
download poc.phar file at http://149.248.44.196/08b54c110a83ff5f9499da1882dbeeb2/poc.phar

----- php.ini -----
+ phar.cache_list =/path/poc.phar
-------------------

After setting php.ini file, execute php cli.

Expected result:
----------------
"AAAAA" is `actual_alias` data.
zend_hash_str_add function use "AAAAA" string as hash key.


Actual result:
--------------
"AAAAA" is `actual_alias` data.
After invoke of efree function, `actual_alias` data is "0x00007ffff3a02008". Then zend_hash_str_add function use
"0x00007ffff3a02008" string as hash key.


# gdb `which php`
(gdb) b *phar_parse_zipfile+11046
(gdb) r
Starting program: /usr/local/bin/php
...
(gdb) x/i $pc
=> 0x5555557817a6 <phar_parse_zipfile+11046>:   callq  0x5555558cd0a0 <_efree>
(gdb) x/32gx $rdi
0x7ffff3a02000: 0x0000004141414141      0x00007ffff3a02010
0x7ffff3a02010: 0x00007ffff3a02018      0x00007ffff3a02020
0x7ffff3a02020: 0x00007ffff3a02028      0x00007ffff3a02030
0x7ffff3a02030: 0x00007ffff3a02038      0x00007ffff3a02040
0x7ffff3a02040: 0x00007ffff3a02048      0x00007ffff3a02050
0x7ffff3a02050: 0x00007ffff3a02058      0x00007ffff3a02060
0x7ffff3a02060: 0x00007ffff3a02068      0x00007ffff3a02070
0x7ffff3a02070: 0x00007ffff3a02078      0x00007ffff3a02080
0x7ffff3a02080: 0x00007ffff3a02088      0x00007ffff3a02090
0x7ffff3a02090: 0x00007ffff3a02098      0x00007ffff3a020a0
0x7ffff3a020a0: 0x00007ffff3a020a8      0x00007ffff3a020b0
0x7ffff3a020b0: 0x00007ffff3a020b8      0x00007ffff3a020c0
0x7ffff3a020c0: 0x00007ffff3a020c8      0x00007ffff3a020d0
0x7ffff3a020d0: 0x00007ffff3a020d8      0x00007ffff3a020e0
0x7ffff3a020e0: 0x00007ffff3a020e8      0x00007ffff3a020f0
0x7ffff3a020f0: 0x00007ffff3a020f8      0x00007ffff3a02100
(gdb) ni
...
(gdb) x/i $pc
=> 0x55555578176e <phar_parse_zipfile+10990>:   callq  0x555555900f10 <zend_hash_str_add>
(gdb) x/32gx $rsi
0x7ffff3a02000: 0x00007ffff3a02008      0x00007ffff3a02010
```

```
0x7ffff3a02010: 0x00007ffff3a02018      0x00007ffff3a02020
0x7ffff3a02020: 0x00007ffff3a02028      0x00007ffff3a02030
0x7ffff3a02030: 0x00007ffff3a02038      0x00007ffff3a02040
0x7ffff3a02040: 0x00007ffff3a02048      0x00007ffff3a02050
0x7ffff3a02050: 0x00007ffff3a02058      0x00007ffff3a02060
0x7ffff3a02060: 0x00007ffff3a02068      0x00007ffff3a02070
0x7ffff3a02070: 0x00007ffff3a02078      0x00007ffff3a02080
0x7ffff3a02080: 0x00007ffff3a02088      0x00007ffff3a02090
0x7ffff3a02090: 0x00007ffff3a02098      0x00007ffff3a020a0
0x7ffff3a020a0: 0x00007ffff3a020a8      0x00007ffff3a020b0
0x7ffff3a020b0: 0x00007ffff3a020b8      0x00007ffff3a020c0
0x7ffff3a020c0: 0x00007ffff3a020c8      0x00007ffff3a020d0
0x7ffff3a020d0: 0x00007ffff3a020d8      0x00007ffff3a020e0
0x7ffff3a020e0: 0x00007ffff3a020e8      0x00007ffff3a020f0
0x7ffff3a020f0: 0x00007ffff3a020f8      0x00007ffff3a02100

(gdb) bt
#0  0x000055555578176e in zend_hash_str_add_ptr (pData=0x55555647da30, len=5, str=0x7ffff3a02000 "\b
\240\363\377\177", ht=<optimized out>) at /home/ubuntu/php-7.4.7/Zend/zend_hash.h:619
#1  phar_parse_zipfile (fp=fp@entry=0x7ffff3a07000, fname=fname@entry=0x7ffff3a0b018 "/home/ubuntu/poc.phar",
fname_len=fname_len@entry=21, alias=alias@entry=0x0, alias_len=alias_len@entry=0, pphar=pphar@entry=0x7fffffffcde8,
    error=<optimized out>) at /home/ubuntu/php-7.4.7/ext/phar/zip.c:715
#2  0x000055555578c72d in phar_open_from_fp (fp=0x7ffff3a07000, fname=fname@entry=0x7ffff3a0b018
"/home/ubuntu/poc.phar", fname_len=fname_len@entry=21, alias=alias@entry=0x0, alias_len=alias_len@entry=0,
    pphar=pphar@entry=0x7fffffffcde8, is_data=0, error=0x0, options=0) at /home/ubuntu/php-7.4.7/ext/phar/phar.c:1720
#3  0x000055555578efd7 in phar_open_from_filename (fname=0x7ffff3a0b018 "/home/ubuntu/poc.phar",
fname@entry=0x7ffff3a03018 "/home/ubuntu/poc.phar", fname_len=21, alias=alias@entry=0x0, alias_len=alias_len@entry=0,
    options=options@entry=0, pphar=pphar@entry=0x7fffffffcde8, error=0x0) at /home/ubuntu/php-
7.4.7/ext/phar/phar.c:1537
#4  0x000055555578f25e in phar_split_cache_list () at /home/ubuntu/php-7.4.7/ext/phar/phar.c:141
#5  phar_ini_cache_list (entry=<optimized out>, new_value=<optimized out>, mh_arg1=<optimized out>, mh_arg2=<optimized
out>, mh_arg3=<optimized out>, stage=<optimized out>) at /home/ubuntu/php-7.4.7/ext/phar/phar.c:183
#6  0x000055555590d423 in zend_register_ini_entries (ini_entry=0x55555634b640 <ini_entries+128>,
ini_entry@entry=0x55555634b5c0 <ini_entries>, module_number=18) at /home/ubuntu/php-7.4.7/Zend/zend_ini.c:252
#7  0x000055555578b6e0 in zm_startup_phar (type=<optimized out>, module_number=<optimized out>) at /home/ubuntu/php-
7.4.7/ext/phar/phar.c:3399
#8  0x00005555558f64a3 in zend_startup_module_ex (module=0x5555563bfdd0) at /home/ubuntu/php-
7.4.7/Zend/zend_API.c:1860
#9  0x00005555558f654c in zend_startup_module_zval (zv=<optimized out>) at /home/ubuntu/php-7.4.7/Zend/zend_API.c:1875
#10 0x0000555555903f32 in zend_hash_apply (ht=ht@entry=0x555556373900 <module_registry>,
apply_func=apply_func@entry=0x5555558f6540 <zend_startup_module_zval>) at /home/ubuntu/php-7.4.7/Zend/zend_hash.c:1812
#11 0x00005555558f683a in zend_startup_modules () at /home/ubuntu/php-7.4.7/Zend/zend_API.c:1986
#12 0x00005555558913d3 in php_module_startup (sf=<optimized out>, additional_modules=<optimized out>,
num_additional_modules=<optimized out>) at /home/ubuntu/php-7.4.7/main/main.c:2332
#13 0x000055555597f2ed in php_cli_startup (sapi_module=<optimized out>) at /home/ubuntu/php-
7.4.7/sapi/cli/php_cli.c:407
#14 0x00005555556623c7 in main (argc=1, argv=0x5555563899d0) at /home/ubuntu/php-7.4.7/sapi/cli/php_cli.c:1323
```

Patches

Add a Patch

Pull Requests

Add a Pull Request

History

All | Comments | Changes | Git/SVN commits | Related reports

[2020-07-14 15:07 UTC] cmb@php.net

```
-Assigned To:
+Assigned To: stas
```

[2020-07-14 15:07 UTC] cmb@php.net

Suggested fix including regression test:
<https://gist.github.com/cmb69/dec5400fbdf619195015a696c3ed136d>

Stas, could you please handle this?

[2020-08-03 06:44 UTC] stas@php.net

```
-CVE-ID:
+CVE-ID: 2020-7068
```

[2020-08-03 06:45 UTC] stas@php.net

```
-PHP Version: 7.4.7
+PHP Version: 7.2.32
```

[2020-08-03 08:09 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=7355ab81763a3d6a04ac11660e6a16d58838d187
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 08:09 UTC] stas@php.net

```
-Status: Assigned
+Status: Closed
```

[2020-08-03 08:09 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=7355ab81763a3d6a04ac11660e6a16d58838d187
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 08:09 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=9c3171f019d07b4271c5929478dddba0861e92af
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 08:10 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=775385df0e954f8cf9b5046bebc8e40ce26e601b
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 09:05 UTC] cmb@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=f57a99937967ed010c6c57b339b703a3fff5eaa6
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 09:16 UTC] cmb@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=7355ab81763a3d6a04ac11660e6a16d58838d187
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function

[2020-08-03 09:16 UTC] cmb@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=9c3171f019d07b4271c5929478dddba0861e92af
Log: Fix #79797: Use of freed hash key in the phar_parse_zipfile function