

---

Subject: \*SOLVED\* privvmpages exhaustion (DoS?)

Posted by [gm77](#) on Sun, 02 Jul 2006 04:39:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I've just found interesting behavior. A non-privileged user can perform a DoS attack on privvmpages. Perhaps it's a bug in OpenVZ, but I haven't investigated it.

I'd like to describe the steps which lead me to this situation. I was packaging Hardened-PHP 5.1.4 and executed 'make test' after successful build (the configure options doesn't make sense). Suddenly, the testsuit has failed with 'Cannot allocate additional ... bla-bla-bla'. I've checked /proc/user\_beancounters and I found that privvmpages were exhausted (I have softlimit set to 2147483647, i.e. maximum). I've checked process list and there were no unnecessary processes detected (just bare minimum), but the held value for privvmpages was still at its top. :( It didn't want to decrease, so I've restarted VPS -- with a timeout of around 10 seconds between shutdown and restart (otherwise the counter isn't decreased :() )

OK, I've investigated a little and found that when /tmp isn't writable to the user who runs tests this bug appears.

To reproduce:

0 (optional). create clean VPS environment

1. groupadd someuser
2. useradd -g someuser -m someuser
3. chgrp someuser /tmp
4. chmod 1707 /tmp
5. su - someuser
6. download php 5.1.4 from [www.php.net](http://www.php.net)
7. unpack, configure --with-pcre-regex, make, make test
8. you've got a problem :)

explanation of steps:

1 and 2 can be combined to single useradd, but some distributions don't create a dedicated group by default and we need the dedicated group for step 3.

3 and 4 - to restrict /tmp to one user only and to not mess with other accounts inside VPS.

5-8 - run-test.php (this script is called by make test) forks test and reads their output from the pipe, one of the tests tries to create a temporary file in /tmp, but gets 'Permission denied', then the following is happening:

```
25982 munmap(0x40018000, 4096)          = 0
25982 gettimeofday({1151812614, 303792}, NULL) = 0
25982 getpid()                          = 25982
25982 open("/tmp/fooWyX3Hb", O_RDWR|O_CREAT|O_EXCL, 0600) = -1 EACCES (Permissio
```

```

n denied)
25982 getpid() = 25982
25982 open("/tmp/foos483hQ", O_RDWR|O_CREAT|O_EXCL, 0600) = -1 EACCES (Permissio
n denied)
25982 write(1, "\nWarning: fwrite(): supplied arg"..., 156 <unfinished ...>
21744 <... select resumed> ) = 1 (in [5], left {59, 914000})
25982 <... write resumed> ) = 156
21744 read(5, "\nWarning: fwrite(): supplied arg"..., 8192) = 156
21744 select(8, [5 7], [], [], {60, 0} <unfinished ...>
25982 write(1, "\nWarning: fwrite(): supplied arg"..., 156) = 156
21744 <... select resumed> ) = 1 (in [5], left {60, 0})
21744 read(5, <unfinished ...>
25982 write(1, "\nWarning: fwrite(): supplied arg"..., 156 <unfinished ...>
21744 <... read resumed> "\nWarning: fwrite(): supplied arg"..., 8192) = 156
25982 <... write resumed> ) = 156
21744 select(8, [5 7], [], [], {60, 0}) = 1 (in [5], left {60, 0})

... and so on until privvmpages are exhausted. :)

```

P.S. Just performed all the step to check before saving the message:

```

builder!sources:~/rpm.d/BUILD/php-5.1.4$ ls -ld /tmp
drwx---rwt 3 root sources 4096 Jul  2 07:32 /tmp
builder!sources:~/rpm.d/BUILD/php-5.1.4$ id
uid=150(sources) gid=150(sources) groups=150(sources)
builder!sources:~/rpm.d/BUILD/php-5.1.4$ ulimit -v
131072
builder!sources:~/rpm.d/BUILD/php-5.1.4$ make test
[a lot of test messages]
[canceled with Ctrl-C since privvmpages are increasing rapidly]
builder!root:/# fgrep privvmpages /proc/user_beancounters
privvmpages 44473039 44485367 2147483647 2147483647 0
builder!root:/#

```

The problem is that once the privvmpages limit is exceeded - there is no way to restore normal operations without VPS re-execution. IMHO, once process has terminated all resources should be freed, but this doesn't happen.