**Problem**

**Fix mcrypt\_create\_iv(..., MCRYPT\_RAND) to auto-seed RNG**

Summary: Without seeding the random number generator,

we'll always get the same IV, and that reduces the security

of this function.

Fortunately, f\_rand() has all of that logic for auto-seeding

and selection of a suitable initial seed built in.

MCRYPT\_RAND should be removed from PHP 7.0 so that developers are discouraged from using it in production systems or in frameworks used by other developers. MCRYPT\_RAND is a constant that instructs mcrypt\_create\_iv() to use a non-cryptographically secure entropy source. While IVs do not have the same secrecy requirements as a private key, they should be unpredictable. Using MCRYPT\_RAND is therefore unsuitable for cryptographic applications.

**ext\_mcrypt.cpp**

iv[--size] = (char)(255.0 \* **rand()** / RAND\_MAX);

**Fix**

**ext\_mcrypt.cpp**

**#include "hphp/runtime/ext/ext\_math.h"**

// Use userspace rand() function because it handles **auto-seeding**

iv[--size] = (char)**f\_rand**(0, 255);

Définition de f\_rand dans ext\_math.h :

int64\_t f\_rand(int64\_t min /\* = 0 \*/, int64\_t max /\* = RAND\_MAX \*/) {

if (!s\_rand\_is\_seeded) {

s\_rand\_is\_seeded = true;

srand(math\_generate\_seed());

}

int64\_t number = rand();

if (min != 0 || max != RAND\_MAX) {

RAND\_RANGE(number, min, max, RAND\_MAX);

}

return number;

}

srand(int $seed = 0, int $mode = MT\_RAND\_MT19937): void