**Problem**

The openssl\_random\_pseudo\_bytes function in ext/openssl/openssl.c in PHP before 5.4.44, 5.5.x before 5.5.28, and 5.6.x before 5.6.12 **incorrectly relies on the deprecated RAND\_pseudo\_bytes** function, which makes it easier for remote attackers to defeat cryptographic protection mechanisms via unspecified vectors.

It was discovered that the PHP openssl\_random\_pseudo\_bytes() function did not return cryptographically strong pseudo-random bytes.

* Fix - use **RAND\_bytes** instead of deprecated **RAND\_pseudo\_bytes**

<https://www.openwall.com/lists/oss-security/2016/04/24/1>

<https://github.com/php/php-src/blob/php-5.4.43/ext/openssl/openssl.c>

**openssl.c**

if ((strong\_result = **RAND\_pseudo\_bytes**(buffer, buffer\_length)) < 0) {

**Fix**

<https://linux.die.net/man/3/rand_pseudo_bytes>

RAND\_bytes() puts num cryptographically strong pseudo-random bytes into buf. An error occurs if the PRNG has not been seeded with enough randomness to ensure an unpredictable byte sequence.

RAND\_pseudo\_bytes() puts num pseudo-random bytes into buf. Pseudo-random byte sequences generated by RAND\_pseudo\_bytes() will be unique if they are of sufficient length, but are not necessarily unpredictable. They can be used for non-cryptographic purposes and for certain purposes in cryptographic protocols, but usually not for key generation etc.

<https://github.com/php/php-src/blob/php-5.4.44/ext/openssl/openssl.c>

**openssl.c**

if (**RAND\_bytes**(buffer, buffer\_length) <= 0) {