The following security alert was issued by the Information Security Division of the Mississippi Department of ITS and is intended for State government entities. The information may or may not be applicable to the general public and accordingly, the State does not warrant its use for any specific purposes.

TLP: WHITE
Disclosure is not limited. Subject to standard copyright rules, TLP: WHITE information may be distributed without restriction.
http://www.us-cert.gov/tlp/

DATE(S) ISSUED:
01/08/2021

SUBJECT:
Multiple Vulnerabilities in PHP Could Allow for Arbitrary Code Execution

OVERVIEW:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow for arbitrary code execution. PHP is a programming language originally designed for use in web-based applications with HTML content. PHP supports a wide variety of platforms and is used by numerous web-based software applications. Successful exploitation of the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.

THREAT INTELLIGENCE:
There are currently no reports of these vulnerabilities being exploited in the wild.

SYSTEMS AFFECTED:
- PHP 8.0 prior to version 8.0.1
- PHP 7.3 prior to version 7.3.26
- PHP 7.4 prior to version 7.4.14

RISK:
Government:
- Large and medium government entities: **High**
- Small government entities: **High**

Businesses:
- Large and medium business entities: **High**
- Small business entities: **High**

Home users: Low

TECHNICAL SUMMARY:
Multiple vulnerabilities have been discovered in PHP, the most severe of which could allow an attacker to execute arbitrary code. Details of these vulnerabilities are as below:

Version 8.0
- Fixed bug #80345 (PHPIZE configuration has outdated PHP_RELEASE_VERSION).

- Fixed bug #72964 (White space not unfolded for CC/Bcc headers).
- Fixed bug #80391 (Iterable not covariant to mixed).
- Fixed bug #80393 (Build of PHP extension fails due to configuration gap with libtool).
- Fixed bug #77069 (stream filter loses final block of data).
- Fixed bug #77961 (finfo_open crafted magic parsing SIGABRT).
- Fixed bug #69625 (FPM returns 200 status on request without SCRIPT_FILENAME env).
- Fixed bug #80438 (imap_msgno() incorrectly warns and return false on valid UIDs in PHP 8).
- Fix a regression with valid UIDs in imap_savebody().
- Make warnings for invalid message numbers/UIDs between functions consistent.
- Fixed bug #80425 (MessageFormatAdapter::getArgTypeList redefined).
- Fixed bug #80404 (Incorrect range inference result when division results in float).
- Fixed bug #80377 (Opcache misses executor_globals).
- Fixed bug #80433 (Unable to disable the use of the AVX command when using JIT).
- Fixed bug #80447 (Strange out of memory error when running with JIT).
- Fixed bug #80480 (Segmentation fault with JIT enabled).
- Fixed bug #80506 (Immediate SIGSEGV upon ini_set("opcache.jit_debug", 1)).
- Fixed bug #80368 (OpenSSL extension fails to build against LibreSSL due to lack of OCB support).
- Fixed bug #80458 (PDOStatement::fetchAll() throws for upsert queries).
- Fixed bug #63185 (nextRowset() ignores MySQL errors with native prepared statements).
- Fixed bug #78152 (PDO::exec() - Bad error handling with multiple commands).
- Fixed bug #66878 (Multiple rowsets not returned unless PDO statement object is unset()).
- Fixed bug #70066 (Unexpected "Cannot execute queries while other unbuffered queries").
- Fixed bug #71145 (Multiple statements in init command triggers unbuffered query error).
- Fixed bug #76815 (PDOStatement cannot be GCed/closeCursor-ed when a PROCEDURE resultset SIGNAL).
- Fixed bug #79872 (Can't execute query with pending result sets).
- Fixed bug #79131 (PDO does not throw an exception when parameter values are missing).
- Fixed bug #72368 (PdoStatement->execute() fails but does not throw an exception).
- Fixed bug #62889 (LOAD DATA INFILE broken).
- Fixed bug #67004 (Executing PDOStatement::fetch() more than once prevents releasing resultset).
- Fixed bug #79132 (PDO re-uses parameter values from earlier calls to execute()).
- Fixed bug #73809 (Phar Zip parse crash - mmap fail).
- Fixed bug #75102 (`PharData` says invalid checksum for valid tar).
- Fixed bug #77322 (PharData::addEmptyDir('/') Possible integer overflow).
- Fixed bug #76813 (Access violation near NULL on source operand).
- Fixed bug #62004 (SplFileObject: fgets after seek returns wrong line).
- Fixed bug #80366 (Return Value of zend_fstat() not Checked).
- Fixed bug #77423 (FILTER_VALIDATE_URL accepts URLs with invalid userinfo). (CVE-2020-7071)
- Fixed bug #77594 (ob_tidyhandler is never reset).

- Fixed bug #80462 (Nullsafe operator tokenize with TOKEN_PARSE flag fails).
- XmlParser opaque object renamed to XMLParser for consistency with other XML objects.
- Fixed bug #48725 (Support for flushing in zlib stream).

Version 7.4

- Fixed bug #74558 (Can't rebind closure returned by Closure::fromCallable()).
- Fixed bug #80345 (PHPIZE configuration has outdated PHP_RELEASE_VERSION).
- Fixed bug #72964 (White space not unfolded for CC/Bcc headers).
- Fixed bug #80362 (Running dtrace scripts can cause php to crash).
- Fixed bug #80393 (Build of PHP extension fails due to configuration gap with libtool).
- Fixed bug #80402 (configure filtering out -lpthread).
- Fixed bug #77069 (stream filter loses final block of data).
- Fixed bug #77961 (finfo_open crafted magic parsing SIGABRT).
- Fixed bug #69625 (FPM returns 200 status on request without SCRIPT_FILENAME env).
- Fixed bug #80425 (MessageFormatAdapter::getArgTypeList redefined).
- Fixed bug #80368 (OpenSSL extension fails to build against LibreSSL due to lack of OCB support).
- Fixed bug #73809 (Phar Zip parse crash - mmap fail).
- Fixed bug #75102 (`PharData` says invalid checksum for valid tar).
- Fixed bug #77322 (PharData::addEmptyDir('/') Possible integer overflow).
- Fixed bug #80458 (PDOStatement::fetchAll() throws for upsert queries).
- Fixed bug #63185 (nextRowset() ignores MySQL errors with native prepared statements).
- Fixed bug #78152 (PDO::exec() - Bad error handling with multiple commands).
- Fixed bug #70066 (Unexpected "Cannot execute queries while other unbuffered queries").
- Fixed bug #71145 (Multiple statements in init command triggers unbuffered query error).
- Fixed bug #76815 (PDOStatement cannot be GCed/closeCursor-ed when a PROCEDURE resultset SIGNAL).
- Fixed bug #77423 (FILTER_VALIDATE_URL accepts URLs with invalid userinfo). (CVE-2020-7071)
- Fixed bug #80366 (Return Value of zend_fstat() not Checked).
- Fixed bug #80411 (References to null-serialized object break serialize()).
- Fixed bug #77594 (ob_tidyhandler is never reset).
- Fixed #48725 (Support for flushing in zlib stream).

Version 7.3

- Fixed bug #77423 (FILTER_VALIDATE_URL accepts URLs with invalid userinfo). (CVE-2020-7071)
- Fixed bug #80457 (stream_get_contents() fails with maxlength=-1 or default).

Successful exploitation of the most severe of these vulnerabilities could allow for arbitrary code execution in the context of the affected application. Depending on the privileges associated with the application, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights.

RECOMMENDATIONS:

The following actions should be taken:

- Upgrade to the latest version of PHP immediately, after appropriate testing.
- Verify no unauthorized system modifications have occurred on system before applying patch.
- Apply the principle of Least Privilege to all systems and services.
- Remind users not to visit websites or follow links provided by unknown or untrusted sources.

REFERENCES:

PHP:

https://www.php.net/ChangeLog-8.php#PHP_8_0
https://www.php.net/ChangeLog-7.php#PHP_7_4
https://www.php.net/ChangeLog-7.php#PHP_7_3

CVE:

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-7071