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:

08/02/2019

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 an attacker to execute arbitrary code. 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. Successfully exploiting 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. Failed exploitation could result in a denial-of-service condition.

THREAT INTELLIGENCE:

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

SYSTEMS AFFECTED:

- PHP 7.1 prior to 7.1.31
- PHP 7.2 prior to 7.2.21
- PHP 7.3 prior to 7.3.8

RISK:

Government:

- Large and medium government entities: **High**
- Small government: **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 7.1.31

- Bug #77919 (Potential UAF in Phar RSHUTDOWN).
- Bug #78222 (heap-buffer-overflow on exif_scan_thumbnail). (CVE-2019-11041)
- Bug #78256 (heap-buffer-overflow on exif_process_user_comment). (CVE-2019-11042)

Version 7.2.21

- Bug #69044 (discrepency between time and microtime).
- Bug #76058 (After "POST data can't be buffered" using php://input makes huge tmp files).
- Bug #77124 (FTP with SSL memory leak).
- Bug #77919 (Potential UAF in Phar RSHUTDOWN).
- Bug #78173 (XML-RPC mutates immutable objects during encoding).
- Bug #78183 (finfo_file shows wrong mime-type for .tga file).
- Bug #78189 (file cache strips last character of uname hash).
- Bug #78192 (SegFault when reuse statement after schema has changed).
- Bug #78202 (Opcache stats for cache hits are capped at 32bit NUM).
- Bug #78222 (heap-buffer-overflow on exif_scan_thumbnail). (CVE-2019-11041)
- Bug #78231 (Segmentation fault upon stream_socket_accept of exported socket-to-stream).
- Bug #78241 (touch() does not handle dates after 2038 in PHP 64-bit).
- Bug #78256 (heap-buffer-overflow on exif_process_user_comment). (CVE-2019-11042)
- Bug #78269 (password_hash uses weak options for argon2).
- Bug #78279 (libxml_disable_entity_loader settings is shared between requests (cgi-fcgi)).
- Bug #78291 (opcache_get_configuration doesn't list all directives).

- Bug #78297 (Include unexistent file memory leak).

Version 7.3.8

- Bug #78212 (Segfault in built-in webserver).
- Bug #69044 (discrepency between time and microtime).
- Bug #78256 (heap-buffer-overflow on exif_process_user_comment). (CVE-2019-11042)
- Bug #78222 (heap-buffer-overflow on exif_scan_thumbnail). (CVE-2019-11041)
- Bug #78039 (FTP with SSL memory leak).
- Bug #78279 (libxml_disable_entity_loader settings is shared between requests (cgi-fcgi)).
- Bug #76058 (After "POST data can't be buffered", using php://input makes huge tmp files).
- Bug #78231 (Segmentation fault upon stream_socket_accept of exported socket-to-stream).
- Bug #78341 (Failure to detect smart branch in DFA pass).
- Bug #78189 (file cache strips last character of uname hash).
- Bug #78202 (Opcache stats for cache hits are capped at 32bit NUM).
- Bug #78271 (Invalid result of if-else).
- Bug #78291 (opcache_get_configuration doesn't list all directives).
- Bug #78338 (Array cross-border reading in PCRE).
- Bug #78197 (PCRE2 version check in configure fails for "##.##-xxx" version strings).
- Bug #78192 (SegFault when reuse statement after schema has changed).
- Bug #77919 (Potential UAF in Phar RSHUTDOWN).
- Bug #78297 (Include unexistent file memory leak).
- Bug #78241 (touch() does not handle dates after 2038 in PHP 64-bit).
- Bug #78269 (password_hash uses weak options for argon2).

Successfully exploiting 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. Failed exploitation could result in a denial-of-service condition.

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-7.php#7.1.31

https://www.php.net/ChangeLog-7.php#7.2.21

https://www.php.net/ChangeLog-7.php#7.3.8

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/

Chris Watts
Security Operations Analyst
MS Department of Information Technology Services
601-432-8201 | www.its.ms.gov