



**State of Alaska Cyber Security &  
Critical Infrastructure  
Cyber Advisory**

**April 30, 2015**

*The following cyber advisory was issued by the State of Alaska and was 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.*

**ADVISORY NUMBER:  
SA2015-051**

**DATE(S) ISSUED:  
04/30/2015**

**SUBJECT:**

**Vulnerabilities in PHP 'unserialize()' Function Could Allow Remote Code Execution**

**OVERVIEW:**

Multiple vulnerabilities have been discovered in the PHP programming language's 'unserialize()' function which could allow for remote code execution and information disclosure.

Successful exploitation may allow an attacker to execute arbitrary code in the context of the user running the affected application or result in denial-of-service conditions. Depending on the privileges associated with the user, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights. Failed exploit attempts will likely cause a denial-of-service condition.

**THREAT INTELLIGENCE:**

There are currently no reports of these vulnerabilities being exploited in the wild, however proof of concept exploit code is publically available from [Seclist.org](http://Seclist.org).

**SYSTEMS AFFECTED:**

- PHP 5.4 prior to 5.4.40
- PHP 5.5 prior to 5.5.24
- PHP 5.6 prior to 5.6.8

**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 has been discovered in PHP versions prior to 5.4.40, 5.5.24, and 5.6.8 which could lead to remote code execution and information disclosure.

These vulnerabilities occur due to a confusion error in the 'unserialize()' function:

The remote code execution vulnerability can be triggered because 'memcpy()' function's third parameter is an unsigned integer. An attacker can exploit this issue by supplying negative value through a fake string-type ZVAL and assigning a value to val which is larger than real allocated memory.

The information disclosure vulnerability can be triggered because the 'Z\_ARRVAL\_P' macro points to a fake ZVAL in memory through a fake HashTable and a fake Bucket. An attacker can exploit this issue by supplying a fake string-type ZVAL and lookup arbitrary memory address through the Z\_STRVAL\_PP macro to disclose sensitive information. Exploiting this further may cause the application to crash.

Successful exploitation may allow an attacker to execute arbitrary code in the context of the user running the affected application or result in denial-of-service conditions. Depending on the privileges associated with the user, an attacker could install programs; view, change, or delete data; or create new accounts with full user rights. Failed exploit attempts will likely cause a denial-of-service condition.

**RECOMMENDATIONS:**

We recommend the following actions be taken:

- Apply appropriate patches provided through php.net to vulnerable systems immediately after appropriate testing.
- Run all software as a non-privileged user (one without administrative privileges) to diminish the effects of a successful attack.
- Remind users not to visit un-trusted websites or follow links provided by unknown or un-trusted sources.
- Inform and educate users regarding the threats posed by hypertext links contained in emails or attachments especially from un-trusted sources.

**REFERENCES:**

Seclists.org Security Mailing List:  
<http://seclists.org/fulldisclosure/2015/Apr/105>

Security Focus:  
<http://www.securityfocus.com/bid/74413>