

PHP functions to avoid

The following is a list of PHP functions that are known to be vulnerable and exploitable. Avoid using these functions in your code.

eval, serialize/unserialize, md5, srand, mt_srand

Reference: devdocs.magento.com/guides/v2.1/extension-dev-guide/security/non-secure-functions.html

or

Magento website → Documentation & Resources → Magento Developer Documentation (Magento 2.1) → Development (PHP Developer Guide) → Security (Non-secure functions)

Date
01/04/2017

Debugging JavaScript with Google Chrome

Developer Tools:

We can debug our javascript code through Google Chrome developer tools very efficiently. Here are some terms to understand:

ctrl + r → (In browser) This ensures that reload the javascript file rather than using the cached version.

Avoid to use Global variable in javascript. so that subsequent included javascript file can't override the variable to produce some error.

Don't use ==, != (compares only value)

Always use ===, !== (compares value & type both)

"Source" Tab ⇒ Breakpoints, Conditional breakpoints, Scope (Local & Global),

Watch (Add variable to watch),

Call Stack,

Control moves from one breakpoint to another breakpoint.

Snippets

⏪ → Resumes script execution

↶ → Step over (Control moves line by line within the function)

↓ → Step in (Jumps from function, wages to fun definition)

↑ → Step out

"Console" Tab ⇒ Live edit javascript,

console.log(), console.clear(),

↑ like var_dump() in PHP

console.assert(expression, object),

console.table(),

-102-

multiline commands → shift + ⏏

(to write 2nd line of code)

(You can assume that these some

line of javascript code would be kept within <script> tag at last line of current browser created HTML page to execute)

Date
01/04/2017

"Elements" Tab \Rightarrow Live edit HTML

"Network" Tab \Rightarrow Page load performance & Ajax

Reference Videos: <http://blog.agindha.com/javascript/debugging-javascript-with-google-chrome-developer-tools/>

Date
03/09/2017

Ideal Page load time in Page Performance

The ideal page load time/speed for our website's HTML to be less than 1.5 seconds.

1. If the HTML page load time is less than 1.5 seconds (top 20% of pages on the web), there are some performance tool, shows/indicates no errors with loading the page.
2. If the HTML page load time is between 1.5-5 seconds or longer than 5 seconds there will be a warning message indicating that the page is not loading at an ideal speed.

Note: Google provides Google page speed test/insight tool to test performance of a webpage.

Date
05/04/2017

Reference: <http://blog.afindia.com/php/xdebug-php-debugging/>

Xdebug Installation

The Xdebug extension helps you debugging your PHP script by providing a lot of valuable debug information i.e. Xdebug is a PHP extension which provides debugging and profiling capabilities.

To install Xdebug, go to URL:

<https://xdebug.org/download.php>

Under RELEASES heading, there is a link 'custom installation instructions', click on it.

Here, paste your phpinfo.html view source code & click on button 'Analyse my phpinfo() output' and now follow the instructions, specially designed for you.

→ mandatory*, rest are optional

[XDebug]

```
zend_extension = D:\xampp\php\ext\php_xdebug-2.5.1-7.0-VC14.dll
```

```
;xdebug.profiler_enable=0
```

```
xdebug.profiler_output_dir="D:\xampp\tmp"
```

```
;xdebug.profiler_output_name="cachegrind.out.%t-%s"
```

```
xdebug.remote_host="127.0.0.1"
```

```
xdebug.remote_log="D:\xampp\tmp\xdebug.txt"
```

```
xdebug.trace_output_dir="D:\xampp\tmp"
```

```
; 3600 (1 hour), 36000=10h
```

```
;xdebug.remote_cookie_expire_time=36000
```

```
xdebug.remote_enable=on
```

```
xdebug.remote_handler="dbgp"
```

make output_buffering=off
in php.ini

IDE Key & remote_port
must be same in php.ini
& in Netbeans configuration

IDE Key=netbeans-xdebug
xdebug.remote_port=9000