

Reflected XSS

High Reflected XSS Source

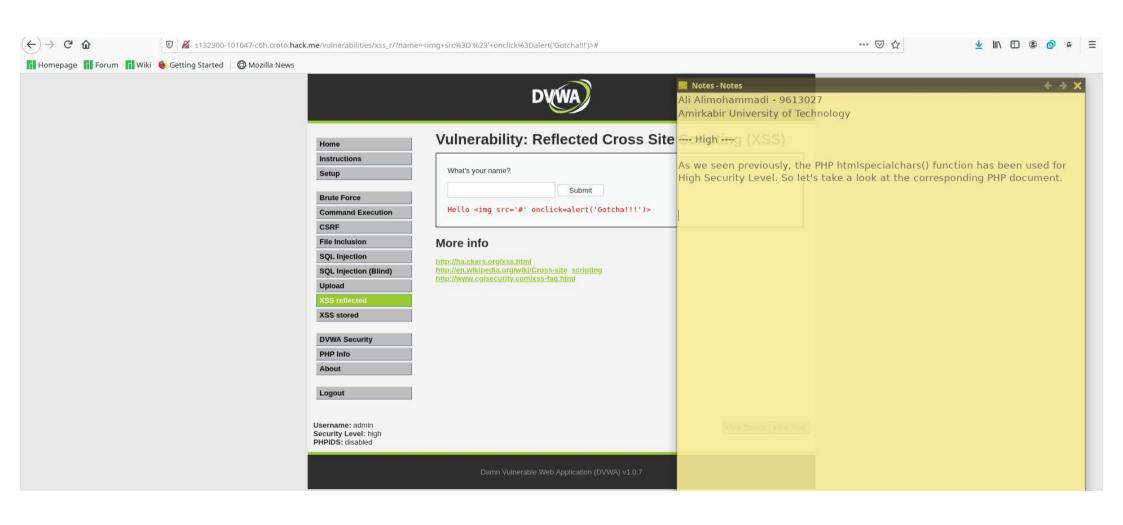
```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . htmlspecialchars($_GET['name']);
    echo '';
}
```

Medium Reflected XSS Source

```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . str_replace('<script>', '', $_GET['name']);
    echo '';
}
?>
```

Low Reflected XSS Source

```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . $_GET['name'];
    echo '';
}
```





htmlspecialchars

(PHP 4, PHP 5, PHP 7)

htmlspecialchars - Convert special characters to HTML entities

Description

```
htmlspecialchars ( string $string [, int $flags = ENT COMPAT | ENT HTML401 [, string $encoding =
ini_get("default_charset") [, bool $double_encode = TRUE ]]] ) : string
```

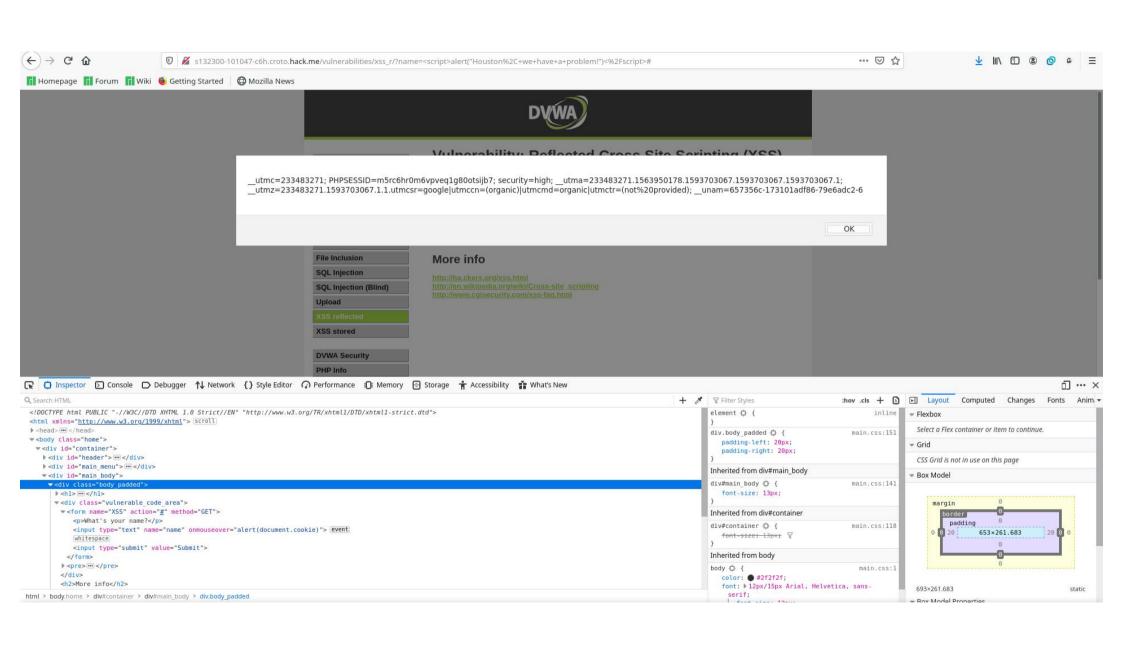
Certain characters have special significance in HTML, and should be represented by HTML entities if they are to preserve their meanings. This function returns a string with these conversions made. If you require all input substrings that have associated named entities to be translated, use htmlentities() instead.

If the input string passed to this function and the final document share the same character set, this function is sufficient to prepare input for inclusion in most contexts of an HTML document. If, however, the input can represent characters that are not coded in the final document character set and you wish to retain those characters (as numeric or named entities), both this function and htmlentities() (which only encodes substrings that have named entity equivalents) may be insufficient. You may have to use mb_encode_numericentity() instead.

Performed translations

Character	Replacement
& (ampersand)	&
" (double quote)	", unless ENT_NOQUOTES is set
' (single quote)	' (for ENT_HTML401) or ' (for ENT_XML1, ENT_XHTML or ENT_HTML5), but only when ENT_QUOTES is set
< (less than)	<
> (greater than)	>

addcslashes addslashes bin2hex convert_uudecode convert_uuencode count chars get_html_translation_table hebrevo html_entity_decode htmlentities htmlspecialchars_decode implode md5_file



Ali Alimohammadi - 9613027

Amirkabir University of Technology

---- High ----

As we seen previously, the PHP htmlspecialchars() function has been used for High Security Level. So let's take a look at the corresponding PHP document.

Next, you'd want to close the tag and start a new one (e.g. <script>) but htmlspecialchars() doesn't let you (as it escapes > and <). So instead, you have to use an event handler that works for <input> tags. The onload event handler doesn't apply to input boxes but you can use others, e.g.onmouseover. (The side effect is that you will need minimal user interaction to trigger the XSS, or need to take advantage of additional attributes like autofocus to have it triggered immediately on page load.)

So I added onmouseover=alert(document.cookie) to the input.

