

Forum:[Hack Forums](#)

Thread: [PDO with htmlspecialchars | SQL Injection](#)

TRH4CK - February 21, 2014, 8:13 pm

Hey !

What up buddys ;) Hope you guys are all doing great!

I had a small question about SQL Injection and PDO.

I currently have this code:

PHP Code

```
<?php
$hash = htmlspecialchars($_GET['h']);
$query1 = $dbh->prepare("SELECT * FROM `uslist` WHERE `uqh` = ?");
$query1->bindValue(1, $hash);
$query1->execute();
$q1f = $query1->fetch(PDO::FETCH_ASSOC);
```

Actually I use codes like these everywhere on my site.

But the question is: Is it possible to SQL Inject in that code ? And is it possible to do any other kind of injection to it ?

Thanks in advance !! :)

SirBuck777 - February 21, 2014, 8:30 pm

I wouldn't recommend storing HTML entities that are already escaped. You should use htmlentities() when displaying user-inputted data, but for storing it, leave it as is.

As for your protection against SQL injection: Looks fine. But you should write a function that really cuts down on the work of having to prepare and bind every time. Let your code do that for you! :)

TRH4CK - February 21, 2014, 8:40 pm

(Feb 21, 2014 09:30 PM)SirBuck777 Wrote: [\[CITE\]](#) I wouldn't recommend storing HTML entities that are already escaped. You should use htmlentities() when displaying user-inputted data, but for storing it, leave it as is.

As for your protection against SQL injection: Looks fine. But you should write a function that really cuts down on the work of having to prepare and bind every time. Let your code do that for you! :)

Alright, i'll check about htmlentities.

Aha thanks, I was afraid I wasn't secure :D. Well I coded (actually recoded because my script was using mysql_) around 5months ago, and I was new at PDO, so I had to write all these prepare, bind and execute :D.

Phizo - February 22, 2014, 12:58 am

No, not since you prepared the statement. Though, keep in mind that htmlspecialchars() without ENT_QUOTES (second argument) allows interpolation of single quotes. That being said, if the

expected comparison value were a string, an attacker could still perform an injection, using alternatives that don't invoke the character translation offered by the function. Also, this is needless for numerical values, as you would simply cast the data to its respective data-type, thus preventing non-numerical values being accepted for interpolation.

MicroCoil - February 23, 2014, 6:00 am

PHP Code

```
<?php
$hash = htmlentities($_GET['h']);
$query1 = $dbh->prepare("SELECT * FROM `uslist` WHERE `uqh` = ?");
$query1->bindValue(1, $hash, PDO::PARAM_STR);
$query1->execute();
$q1f = $query1->fetch(PDO::FETCH_ASSOC);
```

There note when passing a variable to bindValue to PDO its recommended to use a flag, PDO::PARAM_STR or PDO::PARAM_INT