

# Re: Is this secure

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-05/msg01775.html>

---

- *From:* Ray Booyesen <[rj\\_booyesen\\_NS@xxxxxxxxxx](mailto:rj_booyesen_NS@xxxxxxxxxx)>
  - *Date:* Thu, 11 May 2006 15:24:18 +0100
- 

Gef.Mongoose@xxxxxxxxxx wrote:

Ray Booyesen wrote:

Gef.Mongoose@xxxxxxxxxx wrote:

Ray Booyesen wrote:

Gef.Mongoose@xxxxxxxxxx wrote:

Ray Booyesen wrote:

Gef.Mongoose@xxxxxxxxxx  
wrote:

What  
would  
be  
considered  
a  
secure  
way  
to  
store  
passwords?

Paul

Hi Gef

I use SHA1  
to hash my  
passwords.  
When a user  
is created  
on my site,  
his

Re: Is this secure

password  
prefixed  
with a  
randomly  
generated  
salt and  
hashed with  
SHA1.  
Both the  
hashed  
password  
and salt are  
stored in the  
database.

When the  
user logs in,  
his  
password is  
sent to the  
SQL server  
in plain  
text through  
a stored  
proc and the  
stored  
procedure  
returns  
whether it  
is correct or  
not, the salt  
and hash  
never leave  
the database  
once there.

If the user  
changes  
their  
password a  
new salt is  
generated  
and stored  
again in the  
database.

Hope this  
helps.

Regards  
Ray

Re: Is this secure

Hi Ray,

I've created a class to create a random salt, use it with a password to create a salted hash and then put it and the salt into the db.

I'm curious as to what stored proc you use to validate a login password. When the user wishes to log in, they will supply their password, but then i'd need the salt to create a saltedhash to compare against the one in the database. Wouldn't this mean pulling the salt for the user out to create the saltedhash?

Paul

For the case of authenticating, the user name and password is passed to the database. In the stored proc, the password is salted with the stored salt and hashed. Then this hash is compared to the stored hashed password. If they are the same, you can pass back true or 1 or whatever you want.

Let me know if you need any other info! :)

Regards  
Ray

Hi Ray,

Thanks for the answers :). I'm currently using MySQL server with this project, and it doesn't contain any functionality for hashing etc (as far as I am aware anyway). So this means I would have to create the hash outside the DB. Would pulling the salt out to create the salted

Re: Is this secure

hash when a user tries to log in create any huge security risk?  
(I can't see another way of doing this with MySQL server – but i'm still learning so i might have missed something).

Paul

MySQL can create the hashes for you, the functions are built in. E.g. `SELECT MD5(FullName) FROM Customer` will create a hash for you. The SHA hash functions also exist.

I suppose unless you're using the latest version, you won't have access to stored procedures.

Its not really a security risk as the highest risk there would be the transfer of the salted hash and the salt from the database. All the processing will happen server side and shouldn't be too much of a problem. (I'm sure the more paranoid of the forum will quickly say differently. ;) )

Hope this helps.

Regards  
Ray

Thanks again Ray :).

I am using `MYSQ5`, and tried looking through the online manual for hash functions but couldn't seem to find any. I'll take another look.

Thanks again, you've been a huge help :)

Paul

Sure thing! :)

.