

Re: hashed password and UsernameTokenManager

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.webservices.enhancements/2006-01/>

- *From:* "Sami Vaaraniemi" <samivanospam@xxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 Jan 2006 17:44:24 +0200
-

Yes, I agree. It does give a level of protection as users tend to use the same or similar passwords on multiple sites.

Regards,
Sami

"Phil Lee" <phil.lee@xxxxxxxxxxxxxxxxxx> wrote in message
<news:%23nVKiZqFGHA.1288@xxxxxxxxxxxxxxxxxxxxxxxxxx>

> Sami,

>

> As you say an MD5 hashed password in the UsernameToken is a password
> itself (or password equivalent).

>

> The reason for further hashing and salting the already hashed password is
> to make offline dictionary attacks more difficult if someone manages to
> steal the database. See

> <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwse/html/secusernameToken.asp>

>

> There are dictionaries of pre-hashed common passwords available which
> means discovering passwords from non-salted password hashes is easy if
> you have the database. This is why salting and iterative hashing is used.

>

> If someone has stolen your database why bother about them getting the
> passwords? Because people tend to use the same password on more than one
> site, so you are preventing the hacker getting easy access to more sites.

>

> The reason for hashing the password on the client is to provide a simple
> level of protection if someone hacks inside your secure channel (which is
> encrypted of course) and can get access to the unencrypted data. If this
> happens then your pretty stuffed though.

>

> Phil

>

>

> "Sami Vaaraniemi" <samivanospam@xxxxxxxxxxxxxxxxxx> wrote in message
> <news:OfRKzqFGHA.3100@xxxxxxxxxxxxxxxxxxxxxxxxxx>

>>

Re: hashed password and UsernameTokenManager

>> "Steven Cheng[MSFT]" <stcheng@xxxxxxxxxxxxxxxxxxxx> wrote in message
>> news:BERz8onEGHA.3764@xxxxxxxxxxxxxxxxxxxx
>>> Thanks for your response Phil,
>>>
>>> Yes, storing password hash is the best practice... However, the
>>> cleartext
>>> mentioned here is just what the value which is used to construct the
>>> UsernameToken at clientside... Because when it used usernametoken or
>>> derived token to encrypt or sign the message, we'll need the original
>>> cleartext value to construct the token key and decrypt the message....
>>> Thus, if your custom security database stores only password hash, you
>>> need
>>> to also use hashed password text to construct the username token...
>>
>> Note that if you hash the password in the client by giving the hashed
>> password to UsernameToken, then the hashed password *becomes the
>> password*. If you do this, then IMO it is pointless to store the password
>> hash in the database.
>>
>> Regards,
>> Sami
>>
>>>
>>> Thanks,
>>>
>>> Steven Cheng
>>> Microsoft Online Support
>>
>>
>
>

• **References:**

- ◆ **hashed password and UsernameTokenManager**
 ◇ From: Phil Lee
- ◆ **Re: hashed password and UsernameTokenManager**
 ◇ From: Pablo Cibraro
- ◆ **Re: hashed password and UsernameTokenManager**
 ◇ From: Steven Cheng[MSFT]
- ◆ **Re: hashed password and UsernameTokenManager**
 ◇ From: Phil Lee
- ◆ **Re: hashed password and UsernameTokenManager**
 ◇ From: Steven Cheng[MSFT]
- ◆ **Re: hashed password and UsernameTokenManager**
 ◇ From: Sami Vaaraniemi
- ◆ **Re: hashed password and UsernameTokenManager**

Re: hashed password and UsernameTokenManager

◇ *From:* Phil Lee

- Prev by Date: [**Re: WSE2 to WSE3, what to do with the pipeline ?**](#)
- Next by Date: [**Re: WSE2 to WSE3, what to do with the pipeline ?**](#)
- Previous by thread: [**Re: hashed password and UsernameTokenManager**](#)
- Next by thread: [**Re: How to decrypt soap envelop at the client side**](#)
- Index(es):
 - ◆ [**Date**](#)
 - ◆ [**Thread**](#)