hashing sensitive data

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 873 times

0

I need to scramble the names and logins of all the users in a UAT database we have. (because of the data protection act)

However, there is a catch.

The testers still need to be able to login using the hashed login names

so if a user login is "Jesse.J.James" then the hash should be something like

Ypois.X.Qasdf

i.e. approximately the same length, with the dots in the same place

so MD5, sha1 etc would not be suitable as they would create very long strings and also add their own special characters such as + and = which are not allowed by the validation regex.

So I'm looking for some suggestions as to how to achieve this

I guess I need to rollmy own hashing algorith

anyone done anything similar?

I am using c# but I guess that is not so important to the algorithm

thanks alot

ADDED -

Thanks for all the answers. I think I am responsible for the confusion by using the word "Hash" when that is not what needed to be done

c# encryption hash

Share

Improve this question

Follow

edited Oct 6, 2008 at 23:39

asked Oct 6, 2008 at 1:13

ChrisCa

11k ● 22 ● 85 ● 119

Why on earth would they need to be able to login with their hash? And why aren't + and = not allowed? – Joe Van Dyk Oct 6, 2008 at 1:17

the real data is only allowed on the production system + and = are not allowed bacause there are various validation rules governing acceptable login names (alphanumerics plus full stops and apostrophe – ChrisCa Oct 6, 2008 at 1:19

You aren't hashing, you are randomizing. Very different thing. – Tim Howland Oct 6, 2008 at 1:35

@Tim Howland: put that in a comment, so that I can vote it up. – Thilo Oct 6, 2008 at 1:40

@Christo Fur- You should remove the accepted answer mark from my answer (if you can) to bring this question back into visibility. It's not done cooking yet. :) Thanks. – Jeffrey L Whitledge Oct 6, 2008 at 5:16

9 Answers

Sorted by: Highest score (default)

10

Testers should NOT be logging in as legitimate users. That would clearly violate the non-repudiation requirement of whatever data protection act you're working under.

The system should not allow anyone to log in using the hashed value. That defeats the whole purpose of hashing!

I'm sorry I am not answering your specific question, but I really think your whole testing system should be reevaluated.

ADDED:

The comments below by JPLemme shed a lot of light on what you are doing, and I'm afraid that I completely misunderstood (as did those who voted for me, presumably).

Part of the confusion is based on the fact that hashes are typically used to scramble passwords so that no one can discover what another person's password is, including those working on the system. That is, evidently, the wrong context (and now I understand why you are hashing usernames instead of just passwords). As JPLemme has pointed out, you are actually working with a completely separate parrallel system into which live data has been copied and anonymized, and the secure login process that uses hashed (and salted!) passwords will not be molested.

In that case, WW's answer below is more relevant, and I recommend everyone to give your up votes to him/her instead. I'm sorry I misunderstood.

Share Improve this answer Follow

edited Oct 6, 2008 at 4:59

answered Oct 6, 2008 at 1:20

Jeffrey L Whitledge

59.4k ● 9 ● 74 ● 100

8

You do not need to hash the data. You should just randomize it so it has no relation to the original data.

For example, update all the login names, and replace each letter with another random letter.

Share Improve this answer

answered Oct 6, 2008 at 1:25

Follow

WW.
24.3k ● 15 ● 97 ● 124

4

I think you are taking the wrong approach here. The idea of a hash is that it is one-way, noone should be able to use that hash to access the system (and if they can then you are likely still in violation of the data protection act. Also, testers should not be using real accounts unless those accounts are their own.

You should have the testers using mock accounts in a separated environment. By using mock accounts in a separate environment there is no danger in giving the testers the account information.

Share Improve this answer

Follow

answered Oct 6, 2008 at 1:17

Dr8k
1,096 ● 5 ● 11

1

Generally speaking, it is ill advised to roll your own encryption/hashing algorithms. The existing algorithms do what they do for a reason.

Would it really be so bad to either give the testers an access path that hashed the user names for them or just have them copy/paste SHA-1 hashes?

Share Improve this answer

answered Oct 6, 2008 at 1:20

Follow

Aaron Maenpaa
123k ● 11 ● 97 ● 108

> Someone needs to make SO automatically show a warning about rolling your own algorithm when a question is tagged encryption. It'd remove 1/3 of the answers. :-P No offense to you. – PhirePhly Oct 6, 2008 at 3:54

1

Hashes are one-way, by definition.

If all you are trying to protect from is casual perusal of the data (so the encryption level is low), do something simple like a transposition cypher (a 1-1 mapping of different characters to one another -- A becomes J, B becomes '-', etc). Or even just shift everything by one (IBM becomes HAL).

But do recognize that this is by no means a guarantee of privacy or security. If those are qualities you are looking for, you can't have testers impersonating real users, by definition.

Share Improve this answer

Follow

answered Oct 6, 2008 at 1:29

SquareCog
19.6k ● 8 ● 51 ● 63

1

Did this recommendation go through your organization's auditing department? You might want to talk to them if not, it's not at all clear the scheme you're using protects your organization from liability.

Share Improve this answer

Follow

edited Oct 6, 2008 at 1:54

answered Oct 6, 2008 at 1:48

Adam Bellaire

110k ● 19 ● 152 ● 165

hi - no it was an idea that was being kicked around. based on responses here and discussions with others we arent going to do this. we will generate test data to test the system

– ChrisCa Oct 6, 2008 at 2:26

0

Why not use a test data generator for the data that could identify an individual?

Creating test data in a database

Share Improve this answer

Follow

edited May 23, 2017 at 12:13

Community Bot

1 ● 1

answered Oct 6, 2008 at 1:19

Kev

120k ● 53 ● 305 ● 391

0

To give you some more information:

I need to test a DTS package that imports all the users of the system from a text file into our database. I will be given the live data.

However, once the data is in the database it must be scrambled so that it doesnt make sense to the casual reader but allows testers to log in to the system

Share Improve this answer

Follow

answered Oct 6, 2008 at 1:22

ChrisCa

11k • 22 • 85 • 119

0

thanks for all the answers. I think you are almost certainly right about our test strategy being wrong.

I'll see if I can change the minds of the powers that be

Share Improve this answer

Follow

answered Oct 6, 2008 at 1:37

ChrisCa

11k • 22 • 85 • 119