MD5Online

Best source of MD5 information

What is MD5 Salt and How to Use It?

You probably already now that MD5 hashing is not a secure way to store passwords
If you know our services, you probably know that we have a giant database with a lot of words, that can be decrypted in a few seconds
By using salt, you could protect yourself a bit more against this kind of database, but not so much …

What is MD5 Salt and How to Use It?
In cryptography, salt is a random string that you add to an input word, to generate a different hash that with the word alone
MD5 doesn't really offer this feature in the cryptographic algorithm, but you can concatenate two strings to get the same result

In this post I'll explain you what is a salt in the MD5 algorithm, how to use it in your code, and why do you need to use it

Table of Contents

What is an MD5 salt?

I already give you the answer in the introduction, but I'll give you here an example

A database without MD5 salt

Let's say you want to store user passwords in a database
As often, you may want to use MD5 to store the password

Let's take two users from your database, it will look like this:

username	password
b.king	5f4dcc3b5aa765d61d8327deb882cf99
m.donald	ab4f63f9ac65152575886860dde480a1

This is what the table looks like when you use a MD5 function to store the password
If you use the MD5 decryption tool on MD5Online, you'll find in a second what these passwords are

Adding salt

If you start using salt, you'll need to concatenate a string to the use password
To do this, you have two choices

Use a static salt for any users

You can choose to add a static salt like "randomstringforsalt" before any password
So if the m.donald pa

It's a first step for m
It will be as if the m.d

lizengo
Originele software onmiddelijk na aankoop downloaden en gebruiken.
OPENEN

And guess what?
At the time I write these lines, the corresponding MD5 is not in the MD5Online database ☺

Use a dynamic salt

If you always use the same salt, an attacker can find it, and then make his job easier
If he knows that he needs to add “randomstringforsalt” before each password, your salt is no longer useful

To avoid him to understand that, you may use dynamic salt
For example, you can use the account creation date as salt: “azerty20190512”
Or even better, a MD5 hash of the account creation date, like this: “azertyd003a3d8626f9a78abc9ce900b217819”

It’s a basic example, you have to found a better salt, that looks like a random value but that you can find easily to generate password hash

Database with salt

The database with salt looks like exactly the same as previously
And that’s the strength to do that, the attacker will not now directly if you are using salt or not
So he will try without, and maybe never find your passwords

Here is an example with the same password and the static salt:

username	password
b.king	81345f0d478885f72dd51c07cc3ab146
m.donald	244b7f46f6aa268fc862e73d81cfc832

Why do you need to use salt with MD5?

How to decrypt a MD5?

The main weakness of the MD5 algorithm is its speed
You can encrypt a lot of words in a few amounts of time
So it’s possible to make a lot of tries each second, to find the password behind an MD5 hash

To decrypt a password, a hacker will use two different methods:

- Brute force attacks: make the maximum tries each second until he finds the word
- Using a database: look up for the word in a database

If you want to learn more about this MD5 decryption methods, click on the link to check my other post on the subject

Why do you need to use salt?

With both methods, the password length is an issue for him to find the decrypted hash value

In brute force mode, the attacker will probably start with most common passwords, and then start the alphabet list (a, b, … aa, ab, …)
I don’t have the current time needed for each password length, but the more characters you have the best it is. So if you add 32 characters with your salt, no matter the password size, you’re almost safe

And it’s the same problem by using an MD5 hash database
If we don’t consider special characters, there is 62 possibilities for each password letter:

- Alphabet lower case: 26
- Alphabet upper case: 26
- Digits: 10

So the number of total hash to generate and store increase fast:

- 3 characters password: 140,608
- 4 characters password: 7,311,616
- 5 characters password: 380,204,032 possibilities
- 6 characters password: 19,770,609,664 possibilities
- etc …

This is an exponential function
You could guess that a 6 characters password is two times safer than a 3 characters password
But no, it adds a lot more possibilities for each additional character

You understand that with a password of 30 characters or more there is probably no database that will have the information, except for basic phrases
That’s why using salt, or at least asking for long passwords is a good practice

How to add salt into a MD5?

Now that you are convinced that this is important, here's how to do it

In fact, it's easy, you just need to concatenate the two strings together

In PHP for example, you need to use the point symbol to concatenate two strings:

```
<?php
$salt = "randomstringforsalt";
$password = $_POST['password'];

$md5 = md5($salt.$password);
?>
```

It's that simple
Don't forget to use a long salt, or even better a dynamic salt
If your salt is "1234" it's like you didn't use one (azerty or azerty1234 are two weak passwords)

Conclusion

That's it, you now know how to use MD5 salt in your code, and why it's so important if you want to stay with an MD5 encryption method

But if you can, it's probably a best option to choose another algorithm
In PHP you have the password_hash() function, or you can also use bcrypt or scrypt to get safer password in your database

But it's a good thing to have learned what is salt and how to use it, as you can use salt with any cryptographic algorithm

report this ad

© 2020 MD5Online | Powered by WordPress & Customizable Blogily