

Forum:[Hack Forums](#)

Thread: [What is the most safe hashing/crypting algorithm?](#)

MattiasR - March 21, 2013, 9:15 am

Hey! I have tried Googling, but i cant find anything.

So basically i want to know what hashing algorithm is the most safe one to store passwords.

ScreamX - March 21, 2013, 9:46 am

Any algorithm can be safe if you use it correctly. I personally use SHA-256, with a salt. But, there are literally hundreds of algorithms.

Also, make sure your use a one-way hash.

Zher0 - March 21, 2013, 9:58 am

The "most safe" algo, would probably be one you created yourself.

MattiasR - March 21, 2013, 10:08 am

(Mar 21, 2013 09:46 AM)ScreamX Wrote: [\[CITE\]](#) Any algorithm can be safe if you use it correctly. I personally use SHA-256, with a salt. But, there are literally hundreds of algorithms.

Also, make sure your use a one-way hash.

Md5 can't be safe xD

But yeah. I was thinking about either SHA256 or SHA512

elitr - March 21, 2013, 10:14 am

SHA2 or bcrypt are the way to go.

For bcrypt, for a simple hash, this would be enough:

```
crypt ( 'Passwort', 'SALTFORTHEPASSWORD );
```

But you should add some other measures to it like a pseudo-random salt.

MattiasR - March 21, 2013, 11:05 am

(Mar 21, 2013 10:14 AM)elitr Wrote: [\[CITE\]](#) SHA2 or bcrypt are the way to go.

For bcrypt, for a simple hash, this would be enough:

```
crypt ( 'Passwort', 'SALTFORTHEPASSWORD );
```

But you should add some other measures to it like a pseudo-random salt.

Okay, thanks! I'll look it up. ?

SIM-1419016 - March 21, 2013, 11:05 am

```
hash("sha512", $salt . $pass . $salt);
```

MattiasR - March 21, 2013, 11:08 am

(Mar 21, 2013 11:05 AM)Timmarus Wrote: [\[CITE\]](#) hash("sha512", \$salt . \$pass . \$salt);

Is SHA512 more secure than SHA256?

Thanks for helping!

SIM-1419016 - March 21, 2013, 11:09 am

(Mar 21, 2013 11:08 AM)MattiasR Wrote: [\[CITE\]](#) Is SHA512 more secure than SHA256?

Thanks for helping!

Yes, yes it is.

MattiasR - March 21, 2013, 11:12 am

(Mar 21, 2013 11:09 AM)Timmarus Wrote: [\[CITE\]](#) Yes, yes it is.

Okay cool!

By the way, I am trying to make a website which is skid proof.

21redallday12 - March 21, 2013, 11:31 am

I'd just create basic one yourself. eg a = 2 ,b =6 etc

Matt39 - March 21, 2013, 12:51 pm

Don'ts

Don't limit what characters users can enter for passwords. Only idiots do this.

Don't limit the length of a password. If your users want a sentence with supercalifragilisticexpialidocious in it, don't prevent them from using it.

Never hash passwords with SHA1 or MD5! Modern crackers can exceed 60 and 180 billion hashes/second (respectively).

Do's

Use bcrypt when you can; bcrypt if you cannot.

Use PBKDF2 if you cannot use either bcrypt or bcrypt, with SHA2 hashes.

Implement a reasonable 8-10 character minimum length, plus require at least 1 upper case letter, 1 lower case letter, a number, and a symbol. This will improve the entropy of the password, in turn making it harder to crack. (See the "What makes a good password?" section for some debate.)

Source: <http://stackoverflow.com/questions/401656/secure-hash-and-salt-for-php-passwords>

J?mes - March 21, 2013, 1:38 pm

(Mar 21, 2013 12:51 PM)Matt39 Wrote: [\[CITE\]](#) Don'ts

Don't limit what characters users can enter for passwords. Only idiots do this.

Don't limit the length of a password. If your users want a sentence with supercalifragilisticexpialidocious in it, don't prevent them from using it.

Never hash passwords with SHA1 or MD5! Modern crackers can exceed 60 and 180 billion hashes/second (respectively).

Do's

Use bcrypt when you can; bcrypt if you cannot.

Use PBKDF2 if you cannot use either bcrypt or bcrypt, with SHA2 hashes.

Implement a reasonable 8-10 character minimum length, plus require at least 1 upper case letter, 1 lower case letter, a number, and a symbol. This will improve the entropy of the password, in turn making it harder to crack. (See the "What makes a good password?" section for some debate.)

Source: <http://stackoverflow.com/questions/401656/secure-hash-and-salt-for-php-passwords>

I would agree with most of that except that nobody has actually achieved 180 billion SHA / MD5 a second. Also, I personally think making people have one uppercase and a number makes hashes easier to crack. Almost everyone will use their uppercase letter on the first character - you already

know where it's going to be. The symbol will most likely be at the end or a substitute like \$=S, @=A, !=1, etc. You would also generally have numbers on the end before symbols. So technically, you're just telling hash crackers what to aim the attacks at. For example, you would be seeing more of this:

Code Happy24!
H@ppy24
Uw0tm8?
Rather than random combos like

Code jf2ah#rn5
9J&hnf^mk
Nfk5asf&Fv

So I personally wouldn't put a specified amount of specific characters as I find those easier to crack.

(Mar 21, 2013 09:58 AM)Ace - Wrote: [\[CITE\]](#) The "most safe" algo, would probably be one you created yourself.

I personally would go with this. An algorithm that you created yourself that had a low hash rate / sec, e.g. edit up bcrypt a bit.

(Mar 21, 2013 11:05 AM)Timmarus Wrote: [\[CITE\]](#) hash("sha512", \$salt . \$pass . \$salt);
I personally wouldn't recommend doing this. It has a pretty high hash rate / sec.

(Mar 21, 2013 11:08 AM)MattiasR Wrote: [\[CITE\]](#) Is SHA512 more secure than SHA256?
Thanks for helping!
Neither are really worth using but yep.

(Mar 21, 2013 11:31 AM)21redallday12 Wrote: [\[CITE\]](#) I'd just create basic one yourself. eg a = 2 ,b =6 etc
Encoding for passwords? VERY BAD IDEA

MattiasR - March 21, 2013, 3:43 pm

(Mar 21, 2013 12:51 PM)Matt39 Wrote: [\[CITE\]](#) Don'ts

Don't limit what characters users can enter for passwords. Only idiots do this.
Don't limit the length of a password. If your users want a sentence with supercalifragilisticexpialidocious in it, don't prevent them from using it.
Never hash passwords with SHA1 or MD5! Modern crackers can exceed 60 and 180 billion hashes/second (respectively).

Do's

Use script when you can; bcrypt if you cannot.
Use PBKDF2 if you cannot use either bcrypt or script, with SHA2 hashes.
Implement a reasonable 8-10 character minimum length, plus require at least 1 upper case letter, 1 lower case letter, a number, and a symbol. This will improve the entropy of the password, in turn making it harder to crack. (See the "What makes a good password?" section for some debate.)

Source: <http://stackoverflow.com/questions/401656/secure-hash-and-salt-for-php-passwords>
I will use either bcrypt or sha512.