

Forum:[Hack Forums](#)

Thread: [Create Your Own MD5 Hash Cracker](#)

Hackers - January 9, 2024, 8:02 am

Hi all - this tutorial is setup for MD5 hash cracking, but you can use similar code for other algorithms. I chose MD5 because it's a bit easier to work with and show the functionality of a hash cracker. Please keep in mind that if you are cracking a hash of a long string, it could take days, weeks, months, or years to crack it. For short strings, it's not going to take more than a few seconds - such as with a, abc, or other very short strings. Even at 5-6 characters in length, the cracker takes a lot longer to run, and obviously for a 10+ string that can have letters, numbers, or symbols, it can take ages.

I hope this tutorial helps you understand how a cracker works.

The Basics

Understanding the principles of cracking is important.

String: any form of text, using letters, symbols, numbers, etc.

Examples: Password, a sentence, your username, your email - all strings.

Hash: a value (in our case a normal string) that has been encrypted.

Example: the MD5 hash of "abc" is "900150983cd24fb0d6963f7d28e17f72"

It's not secure to store a user's password in "plaintext" (unencrypted format), so we encrypt it with an algorithm (such as MD5) so the encrypted version is stored instead. What if you forget your own password and only have the hash? That's when you can use a hash cracker to try to retrieve your old password!

PHP:

I am including the PHP example because of how many people use PHP and because MD5 is commonly used in PHP systems more than it's used by other languages (in my experience). Please note that PHP cracking is not going to be nearly as fast as Python or lower level languages.

Spoiler ([Click to View](#))

How to get an MD5 hash

Getting an MD5 in PHP is super easy because it's a default function:

PHP Code

```
<?php
```

```
md5("abc")
```

This will return our hash of: 900150983cd24fb0d6963f7d28e17f72

If we only knew the hash and not "abc" then we could test strings until we find a match.

For example, here is essentially what a cracker does... it tests strings until it finds the hash:

PHP Code

```
<?php
md5("a")
md5("b")
md5("c")
md5("ab")
md5("ac")
md5("abc")
```

So we could do a comparison:

PHP Code

```
<?php
if(md5("abc") === "900150983cd24fb0d6963f7d28e17f72"){
    echo "We found the matching hash!";
}
```

Let's make it automatic! For this next part, you only need to copy over the md5Crack function and then use the \$result code in same file to start cracking! Just give it a hash \$target and let it run!

First, here is our main structure for our PHP script which will start our functionality:

PHP Code

```
<?php
$target = '900150983cd24fb0d6963f7d28e17f72'; // MD5 hash of 'abc'

$result = md5Crack($target, 1, 4);
if ($result) {
    echo "Password found: $result\n";
} else {
    echo "Password not found within the specified range and character set.\n";
}
```

The function md5Crack is not a default PHP function, it's made by me. It takes a char set (ex: A-Z uppercase, a-z lowercase, 0-9 numbers, and common symbols) and a min & max length for our string we're trying to find (ex: 1 char to 16 chars in length).

PHP Code

```
<?php
function md5Crack($targetHash, $minLength, $maxLength) {
    $charset = array_merge(range('a', 'z'), range('A', 'Z'), range('0', '9'), str_split(
'!@#$%^&*()_-=+[]{}|;:,.<>?'));
    $maxCharsetIndex = count($charset) - 1;
    $stack = array_fill(0, $maxLength, 0);

    while (true) {
        for ($length = $minLength; $length <= $maxLength; $length++) {
            $attempt = "";
            for ($i = 0; $i < $length; $i++) {
                $attempt .= $charset[$stack[$i]];
            }

            if (md5($attempt) == $targetHash) {
                return $attempt;
            }
        }
    }
}
```

```

}

// Increment stack
$stack[0]++;
for ($i = 0; $i < $maxLength - 1; $i++) {
    if ($stack[$i] > $maxCharsetIndex) {
        $stack[$i] = 0;
        $stack[$i + 1]++;
    }
}

// Check if all combinations are tested
if ($stack[$maxLength - 1] > $maxCharsetIndex) {
    return null;
}
}
}

```

Python:

I am including Python because it's faster than PHP at a lot of things, and it's also a very commonly used language.

Spoiler ([Click to View](#))

How to get an MD5 hash in Python:

I'm sure there are multiple ways, but in my case I've chosen to use hashlib to be the core of my hash cracker.

```

Code hashlib.md5("a".encode()).hexdigest()
hashlib.md5("b".encode()).hexdigest()
hashlib.md5("c".encode()).hexdigest()
hashlib.md5("ab".encode()).hexdigest()
hashlib.md5("ac".encode()).hexdigest()
hashlib.md5("abc".encode()).hexdigest()

```

So we could do a comparison:

```

Code if hashlib.md5("a".encode()).hexdigest() == "900150983cd24fb0d6963f7d28e17f72"){
    print("We found the matching hash!")
}

```

Let's make it automatic! For this next part, we'll be doing the same as we did in our PHP function - looping through our full min/max range (length of string) and using the char set (A-Z, a-z, 0-9, symbols) to test different possible values. The md5_crack function is not default, and all it's really doing is facilitating the loop logic to wrap the hashlib md5 function until it finds the matching hash.

Full Source Code:

Unfortunately HF firewall won't let me post this code, so here is a pastebin:

<https://pastebin.com/Zr3fBJdL>

and if pastebin goes down, here is an image:

(Jan 9, 2024 10:21 AM)johnbelton446 Wrote: [\[CITE\]](#) interested in this please let me know
All the source code is in the spoilers, enjoy! :)

Kaushal@67 - May 7, 2025, 9:27 am

Nice, cracking md5 hash

SIM-5600039 - May 21, 2025, 2:07 pm

Very interesting information I will try it out.

Hackers - March 16, 2026, 8:22 pm

(May 21, 2025 02:07 PM)SIM-5600039 Wrote: [\[CITE\]](#) Very interesting information I will try it out.
Thanks, hope you did. Useful for learning