

Forum:[Hack Forums](#)

Thread: [\[Tut\] How to identify the type of hashes \[Tut\]](#)

0m3r - May 14, 2012, 5:12 am

I see alot people here on hf asking what type of this hash is etc etc.. i'll share some things what i know i m not good at decrypting-encoding-etc.. lets start ..

In this tutorial show how to define some basic types of hashes and there examples: Base64, Megan-35, Feron-74, AER-256, Armon-64, ATOM-128, ESAB-46, EZIP-64, GILA7, OKTO3, Tigo-3FX, Trip-5, ZARA-128, ZONG22, HAZZ-15, MD5 , sha-256(unix) , sha-512(unix)

Base64

Ends with == or = mostly

for example.

Code c21hc2hwdW1wa2lucw==

Megan 35

ends with 5

for example.

Code mwwXmwXqngvqkw1Vmq55

feron 74

ends with 4

for example.

Code CmlxCmxisuliPmygCi44

AER-256

It starts with 2

The second number is never higher than 6

Sequence with a periods and commas etc..

for example.

Code 2475686.270836455, 2412016.365752809, 33.673757802746564

Armon-64

Usually has a + in the middle

There is a sequence point point point

It starts with 2

Only the numbers except + in the middle

for example.

Code 25448157090.709496+94249848.213684

ATOM-128

Often contains +

Mainly letters

It starts with the letter

Often ends with CC

for example.

Code nlhqLjQ5MiT6MI63+/CC

ESAB-46

Ends with / /

It contains letters and numbers

for example.

Code 8mlR8mRg92lg3mVe8g//EZIP-64

A large sequence of numbers

A common sequence. /. /

It usually starts with a

for example.

Code 15623.730897750893/14124.447874419286/15213.338713579566/-126.04199709670604

GILA7

It usually ends with the capital letter

It usually ends with the G

It can contain +, =

for example.

Code qrF1qr1538F5DrOmQ5GG

OKTO3

listing the numbers

for example.

Code 74806369382479140

747938510478572200

761035954145620400

696080542373685800

-515497671532164.4

Tigo-3FX

Usually contains =

Mostly end up with 'FXX'

for example.

Code 7If9=8fpBltnBkHY

TRIPO-5

Mostly ends with 'TT'

It usually contains /

for example.

Code Vb9KdExM/zpMWwTT

ZARA-128

It looks like this:

Code 427 421 409 427 416 424 429 421 424 419 417 422 427

ZONG22

It usually ends with the two

Can keep =, +

Code dfNbYvNhoop2

HAZZ-15

It usually ends with the 55

It usually starts with a 'q'

for example.

Code qmlGwujeD8JFDoFsEH55

MD5

It contains 32 digits

Includes numbers 1-9 and letters a-f

for example.

Code ff1f51cac8e1ad0bf7453e308b237a92

SHA-256(Unix)

has 55 characters in hash

for example.

Code \$5\$12345678\$5B8vYYiY.CVt1RITTf8KbXBH3hsxY/GNooZaBBGWec5

SHA-512(Unix)

has 98 characters in hash

for example.

Code

\$6\$FFH2inEE\$bDig7qKFq0m3wZY6BahqWjTzmcFma9rAfPPocCvyXKyGqrErw5s1daWDotarbiUk76Sb8NO/ZhJ8WzeM6RW.q0for decryption use : <http://www.crypo.com/>

Enjoy if you like it please comment below i would love it

binary_strider - May 14, 2012, 5:28 am

Shit man.. looks really good..

Bookmarked the page for later use :D

0m3r - May 14, 2012, 5:35 am

thank you both of you for appreciating....

K?ala - May 14, 2012, 5:35 am

Very nice job OP. Very useful guide.

Sinofield - May 14, 2012, 5:46 am

base64 and half the things on that list aren't hashes. They're encoding schemes. I can decode a base64 message, for example the one in your post is "smashpumpkins". MD5, SHA-256, SHA-512 are hashes. You can't get the original text back unless you bruteforce it. It can't be "decoded".

The real way to figure out what a hash is, is either count the number of bits in the hash, for example, MD5 is usually represented as hex, so count the number of letters in the hash, 32 for MD5, half that, so 16 bytes, and multiply by 8, which gives you 128-bit hash.

<http://google.com/search?q=128+bit+hash>

And you can see the first result is MD5.

For the UNIX crypt hashes, you just look at what's in the between the first two dollar signs:

\$1\$

Paul Hennig-Kamp's MD5 scheme.

\$2a\$

Niels Provos' Blowfish scheme.

\$5\$

Ulrich Drepper's SHA-256 scheme.

\$6\$

Ulrich Drepper's SHA-512 scheme.

\$apr1\$

Identical to \$1\$.

{SHA}

SHA-1 hash, also used by the Apache Portable Runtime library.

\$P\$

MD5-based hash used by PHPass

\$H\$

Same as above, but used by PHPbb because, well, they're PHP developers.

\$2x\$

"compatibility" option for OpenWall's bcrypt implementation.

\$2y\$

"force correct algorithm" option for OpenWall's bcrypt implementation.

So \$5\$12345678\$5B8vYYiY.CVt1RITTf8KbXBH3hsxY/GNooZaBBGWec5

Is an SHA-256 hash. Note that "regular" SHA-256 hashes look like this:

3f209d3a614607ba72dbcc4f5aee118ca134cce93aba7a87c8c144f1abd03f4

The UNIX crypt hashes are not compatible with the "regular" form, the schemes are different, they're made to be more computationally expensive.

Blue_x33 - May 14, 2012, 5:49 am

Great thread but there is a Tool for it (idk the name right now because i am on my phone) but it's something like Hash analyzer

0m3r - May 14, 2012, 5:49 am

(May 14, 2012 05:46 AM)Sinofield Wrote: [\[CITE\]](#) base64 and half the things on that list aren't hashes. They're encoding schemes. I can decode a base64 message, for example the one in your post is "smashpumpkins". MD5, SHA-256, SHA-512 are hashes. You can't get the original text back unless you bruteforce it. It can't be "decoded".

The real way to figure out what a hash is, is either count the number of bits in the hash, for example, MD5 is usually represented as hex, so count the number of letters in the hash, 32 for MD5, half that, so 16 bytes, and multiply by 8, which gives you 128-bit hash.
etc etc

i just post the basic which i told you guys in starting of thread and thanx for more information

and thank you to those who like this thread...

Kyle FYI™ - May 14, 2012, 6:15 am

Its always good to know this shit :)

.Pestilence. - May 14, 2012, 6:19 am

Thank you for posting this, definitely getting added to my bookmarks.

0m3r - May 14, 2012, 6:21 am

thank you boht of you guys :) enjoy

PawKy - May 14, 2012, 6:25 am

thanks you man! i will learn that soon!

kurosakinaruto - May 14, 2012, 6:47 am

gr8 stuff man!!

also nice sinofield!!

:D

bookmarked this

:D

0m3r - May 15, 2012, 3:58 am

thank you both of you guys i really appreciate your reply's ...

Keeper - May 15, 2012, 5:14 am

Woo, nice share. Thanks, I actually learnt a few new things!

fabiosuper - May 15, 2012, 5:54 am

very nice job. it is useful for me thanks

??g????!? - May 15, 2012, 7:40 am

Very helpfull to new users and pro.

Another bookmark everyone must have.

0m3r - May 19, 2012, 5:14 pm

thank you all of you for replying your feeds in this thread

Vuln - May 24, 2012, 3:14 pm

Good job OP.

A Base64 string doesn't have to end with == though. It could either end with an equals sign ('='), a double equals sign ('=='), or even without any equals signs ('').

0m3r - May 24, 2012, 4:50 pm

(May 24, 2012 03:14 PM)[He]isenberg Wrote: [\[CITE\]](#) Good job OP.

A Base64 string doesn't have to end with == though. It could either end with an equals sign ('='), a double equals sign ('=='), or even without any equals signs ('').

i mention on the top that its is a basic way mostly base64 ends with == and =

And thanx of appreciate and i edit the original post and add (=).

Bush - July 19, 2012, 2:20 pm

Thank you for this. It makes it easy to identify these hashes :)

Error2012 - August 23, 2012, 12:13 am

+1Pts

Hades10 - August 23, 2012, 12:31 am

Vey nice Tutorial....

Thank you very much!

(Bookmarking the page)

kivik - August 23, 2012, 10:11 am

I find it very useful

thanks for sharing

aloksaini - August 23, 2012, 10:17 am

BookMarked....

a quality post....

i didn't know that...

thanks...

Subpar - August 23, 2012, 10:26 am

Bookmarked, very HQ post here.

Thanks so much for this man, very very awesome!

LevelUp - September 25, 2012, 6:48 am

Really great TuTorial man, thanks alot for the info!

d4rk41g0r1thm - November 3, 2012, 4:01 am

Wow very good Tutorial helped me alot thank you :)

The Ion - November 3, 2012, 5:42 am

thanks for the thread there is a hash identifier but i prefer manual checking

stuvtro - November 3, 2012, 2:05 pm

Base64 isn't a hashing function.

Base64 doesn't necessarily require one or two '=' signs appended.

A good page: http://en.wikipedia.org/wiki/List_of_hash_functions

herrdeutschland - November 18, 2012, 4:34 am

nice work dude ,it looks just the same as my tut but here it s more clea

thanks anyway

Gift - November 19, 2012, 7:59 pm

I have to post on here and say thank you! This actually helped! I can at least visualize what I should do.

J?mes - November 19, 2012, 8:01 pm

May I also add that <http://www.insidepro.com/hashtypes.php> has a wide range of hashing types and will most probably help you find the hash you're looking for :).

Toasties - November 19, 2012, 8:50 pm

nice job op i like it. will keep for resources

Gustaf42 - November 20, 2012, 4:09 am

Some awesome shit man! Thanks for that, thumbs up!

//Gustaf42

0m3r - December 5, 2012, 3:02 am

thank you all of you guys i really appreciate your reply's

DEVIL X - January 21, 2013, 7:44 am

nice post mate, i think you should add a link to www.crypo.org

Pixelstealer - January 21, 2013, 8:35 am

Thanks for the share 0m3r =)!!!!

~Pixelstealer~

J?mes - January 21, 2013, 2:52 pm

I just re-read this thread and noticed that half of them are not even hashing algorithms. Only the last 3 examples are actually examples of hashes and the others are encryption methods.

vico100 - August 15, 2013, 1:54 pm

nice tutorial man . well done :) keep it up

Stellan - December 30, 2014, 9:51 pm

Old thread -.- but another thing to note is that with base64, the number of bytes will always be divisible by 3. Meaning that the = signs will or will not be there depending on how large the string being encoded is. For example "Encoded" is "RW5jb2RIZA==" as being a 7 byte string, it needed 2 more bytes to be divisible by 3.

LynS - *December 31, 2014, 8:42 am*

Thank you for the tutorial bro !

Really like it