4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	1 sec	2 secs	4 secs
8	Instantly	Instantly	28 secs	2 mins	5 mins

Are Your Passwords in the Green?

Infographic | Research Apr 18 | Written By Corey Neskey

It's here! The **2023 update to the Hive Systems Password Table** that's been shared across the internet, the news, universities, and by thousands of organizations worldwide. So what's new, and how did we generate this eye catching table? **Keep reading below!**

Not a reader? Check out **our video review** of this year's table instead!

Looking for high resolution versions to download? Or the table in other languages?

DOWNLOAD NOW

Follow us - stay ahead.

Mar 1, 2024

Let's Talk About Cookies!

Cookies help enhance our browsing experience, but what are the risks? Learn more about how cookies

17 min read

Since 2020, we've conducted a lot of research to develop and present the Hive Systems Password Table. But for those of you that want to know about the "how" then you've come to the right place because we're going to walk you through our methodology. While the data fits nicely into the table above, things aren't as as simple as they look. So we'll talk through the data, our assumptions, and oh, you're going to see a LOT of variations of the password table.

- **Got a question or comment?** Leave it below or message us on your favorite social media platform.
- **Heard about the LastPass breach?** Check out the variations of our Password Table for that exact scenario and see how you may be impacted.

"So how'd you make the table"?"

In 2022, we shared our update to a colorful infographic table that showed the relative strength of a hashed password against a cracking attempt, based on the password's length, complexity, hashing algorithm used by the victim, and the hardware used by the attacker. The data was based on how long it would take a consumer-budget hacker to crack your password hash using a desktop computer with a top-tier graphics card and then how long an organized-crime-budget hacker would take leveraging cloud compute resources. We looked at big name providers like Amazon AWS and Microsoft Azure but also the growing non-corporate

from misuse.

Feb 23, 2024

Navigating the Dual Impact of AI in Cybersecurity

Artificial Intelligence (AI) is set to be the newest ally for many companies, but it's also set to be the newest threat.

Feb 12, 2024

SOC 2 Made Simple

Are you tired of filling out lengthy vendor questionnaires and are looking to pursue an SOC 2 examination report instead? Discover some basic practices to secure your customers' data and pave the way for SOC

This year we've updated our cracking hardware to the latest and greatest, including that of the internet darling ChatGPT! We also opted for a more realistic set of special characters in our testing. Most websites only accept these ^*%$!&@# and so we dropped the rest. This only impacts the right-most column of the password table.

First, let's get some key terms out of the way. We're going to talk about "hashing." In the context of passwords, a "hash" is a scrambled version of text that is reproducible if you know what hash software was used. In other words, if your friend hashes the word "password" using MD5 hashing software, the output hash will be *5f4dcc3b5aa765d61d8327deb882cf99*. Now if you hash the word "password" using MD5 hashing software, you'll also get *5f4dcc3b5aa765d61d8327deb882cf99*! You and your friend both secretly know the word "password" is the secret code, but anyone else watching you just sees *5f4dcc3b5aa765d61d8327deb882cf99*. Passwords are stored in servers as hashes like this instead of in plain text like "password." That way, if someone steals the database all they can see are these hashes but not the password that made them.

You can't do the reverse. A hash digest like *5f4dcc3b5aa765d61d8327deb882cf99* can't be computed to produce the word "password" that was used to make it. Hashing software is a one-way-street by design. The way that hackers solve this problem is by "cracking" the passwords instead. In this context, "**cracking**" means making a list of all combinations of characters on your keyboard and then hashing them. Then you look for matches between the list and a breached database of password hashes. You can do that with any

Graphics cards are those circuit boards that stick out of your computer's bigger green circuit board. Among other things, this special circuit board has a GPU on it. A **GPU** is the shiny square tile on your graphics card that says NVIDIA or AMD on it. GPU stands for graphical processing unit – they were built to make pictures load faster on your computer screen (and to play great video games). As it turns out, they're also great at calculating hashes too. A popular application for hashing is called **Hashcat**. Hashcat includes hashing software like MD5 and allows you to try not just MD5 but thousands of others and see how fast it was able to do so. We usually say "hash function" instead of "hash software."

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 secs	22 mins	1 hour	8 hours
9	Instantly	2 mins	19 hours	3 days	2 weeks
10	Instantly	58 mins	1 month	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	61tn years	100tn years	7qd years

Our first password table in 2020 used data from 2018 based on MD5 hashed passwords cracked by an RTX 2080 GPU.

We based our first password table (above) and time estimates on 2018 GPU (RTX 2080 graphics card) and 2018 security practices (MD5 hashing). In fact, that appears to still be the assumption many "How

compares the two cards in terms of calculations per second and hashes per second.

When shopping for a graphics card or cloud GPU, you're given "calculations per second," usually in "floating point operations per second" (FLOPS). The FLOPS measure doesn't take into account the unique properties of hashing algorithms, password character composition, and the hardware "around" the graphics card like your motherboard, CPU, and RAM. Fortunately, hashcat made it easy for password recovery experts to automate testing their hardware on real hashing exercises and then log the results to share. The result is an ever-growing dataset of observed hashing performance using various hardware and hashing approaches called "benchmarks".

MD5	Calculations per second (FP32 aka float) FLOPS	Hashes per second (H/s)
RTX 2080	10,070,000,000,000	37,085,000,000
RTX 3090	35,580,000,000,000	69,379,700,000
RTX 4090	82,580,000,000,000	164,100,000,000

Comparison of an RTX 2080, 3090, and 4090 calculations per second and MD5 hashes per second.

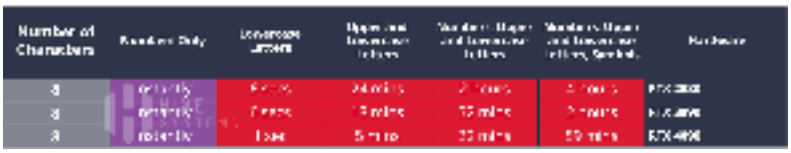
- In 2018 the RTX 2080 card cracked about 37 billion hashes per second (H/s). Sites hosting Mark Wales' HowSecureIsMyPassword (HSIMP) code rounded that up to 40 billion H/s. Many sites continue to assume that hardware and

about 70 billion H/s.

- In 2022 we saw the RTX 4090 crack about 164 billion H/s.

"So how much faster is that in terms of time?"

Assuming the 8-character password recommendation from NIST is used:



Max time required to crack 8-character MD5 password hashes of various complexity.

So, random, complex 8-character passwords that once took four hours to crack now only take one. If you leverage consumer cloud computing, minutes, if you leverage enterprise cloud computing instant.

Let's look at the tables side by side.

Number of Characters	Numbers Only	Lowercase Letters
4	Instantly	Instantly
5	Instantly	Instantly
6	Instantly	Instantly
7	Instantly	Instantly
8	Instantly	5 secs
9	Instantly	2 mins

Number of Characters	Numbers Only	Lowercase Letters
4	Instantly	Instantly
5	Instantly	Instantly
6	Instantly	Instantly
7	Instantly	Instantly
8	Instantly	3 secs
9	Instantly	1 mins

Number of Characters	Numbers Only	Lowercase Letters
4	Instantly	Instantly
5	Instantly	Instantly
6	Instantly	Instantly
7	Instantly	Instantly
8	Instantly	1 secs
9	Instantly	33 secs

Password tables comparing MD5 hashes cracked by the 2080, 3090, and 4090.

"This seems like a job for the cloud, right?"

Number of Characters	Numbers Only	Lowercase Letters	Number of Characters	Numbers Only	Lowercase Letters
4	Instantly	Instantly	4	Instantly	Instantly
5	Instantly	Instantly	5	Instantly	Instantly
6	Instantly	Instantly	6	Instantly	Instantly
7	Instantly	Instantly	7	Instantly	Instantly
8	Instantly	1 secs	8	Instantly	Instantly
9	Instantly	33 secs	9	Instantly	10 secs

Password tables comparing MD5 hashes cracked by one RTX 4090 against 8 A100 GPUs from Amazon AWS.

And for comparison, here's how things keep stacking up:

Max time required to crack 8-character MD5 password hashes of various complexity.

If you have the cash, you can rent the ungodly power of Amazon's high performance computing clusters. At the moment, Amazon offers renting 8 NVIDIA A100 Tensor Core GPUs through their EC2 P4d offering called **"p4d.24xlarge"** and advertised as the "highest performance for ML training and HPC applications in the cloud," at just $32.77 per hour. Not fast enough? Buy more instances! Note that these are the maximum amounts of time it would take to crack a password, so you'd most likely be spending less. The more instances you have running in parallel, the less time it will take.

MD5	Calculations per second	Hashes per

8 x A100s	2,500,000,000,000,000	523,500,000,000

Comparison of an RTX 2080 GPU and 8 x A100 GPUs calculations per second and hashes per second.

Going back to our consumer-grade hacker point of reference, let's assume we want to spend less money and avoid the big corporate cloud. Sites like vast.ai enable regular people to rent out their computer hardware through their residential internet connection. At the time of writing, the top performing rental was not one, but *twelve* RTX 4090s for the low, low price of $6 per hour! We don't know anything about the security or business practices of vast.ai, so tread carefully.

MD5	Calculations per second (FP32 aka float) FLOPS	Hashes per second (H/s)
RTX 2080	10,070,000,000,000	37,085,000,000
RTX 3090	35,580,000,000,000	69,379,700,000
RTX 4090	82,580,000,000,000	164,100,000,000
8 x A100s	155,920,000,000,000	517,742,464,000
8 x RTX 4090s	660,000,000,000,000	1,312,800,000,000
12 x RTX 4090s	1,237,200,000,000,000	1,939,500,000,000

"But what about my favorite chatbot ChatGPT?"

In the spirit of ~~creative grand conspiracy theories~~ sophisticated counterfactual reasoning:

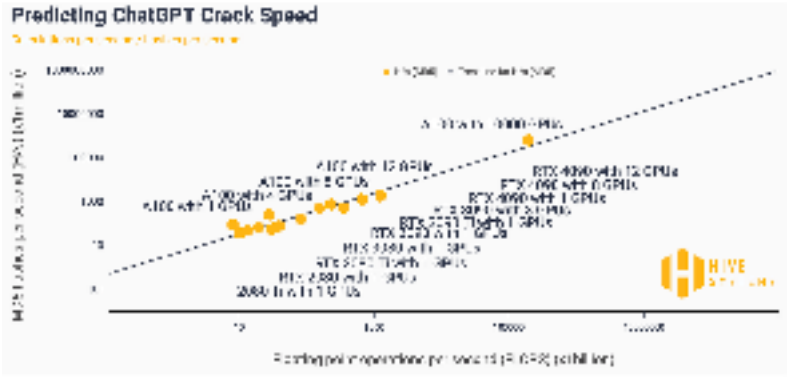
Suppose you trick venture capitalists into funding your AI project but then use that power for evil to crack passwords. ChatGPT was trained on a Microsoft Azure supercomputing offering consisting of 10,000 NVIDIA A100 GPUs.

What would the password table look like under the influence of that kind of hardware?

MD5	Calculations per second (FP32 aka float) FLOPS	Hashes per second (H/s)
8 x A100s	155,920,000,000,000	517,742,464,000
12 x A100s	233,880,000,000,000	776,613,696,000
10,000 x A100s	?????	?????

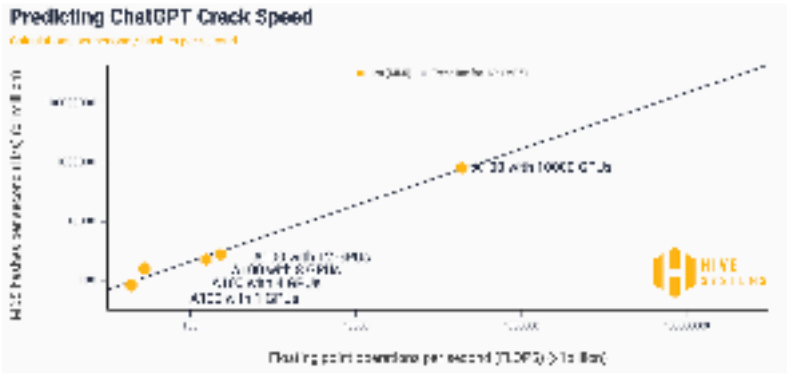
Comparison of an A100 x8 x 12 and x10,000 GPUs calculations per second and hashes per second.

We couldn't get our hands on 10,000 A100s to run a test but we can infer based on how FLOPS scale linearly with Hashes. FLOPS are the advertised "calculations per second" in general that GPU manufacturers write on the box. Hashes per second



Comparison of several GPUs calculations per second and MD5 hashes per second with a trend line for H/s.

If we zoom into just the A100s we can see there is still a trend:



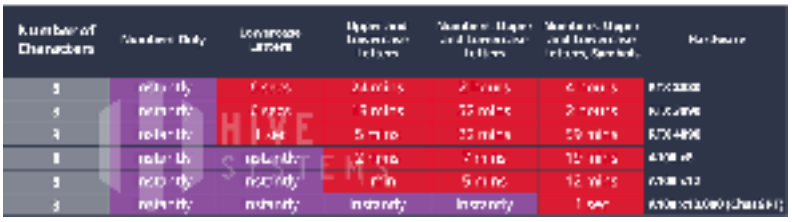
Comparison of A100 x1,4,8,12, and 10,000 GPUs calculations per second and MD5 hashes per second with a trend line for H/s.

If we multiply the speed for one A100 to 10,000 and add the same degradation factor we saw for the others of (about 5.5%) that allows us to fill in the last row of our table as:

8 x A100s	155,920,000,000,000	517,742,464,000
12 x A100s	233,880,000,000,000	776,613,696,000
10,000 x A100s	194,900,000,000,000,000	647,178,080,000,000

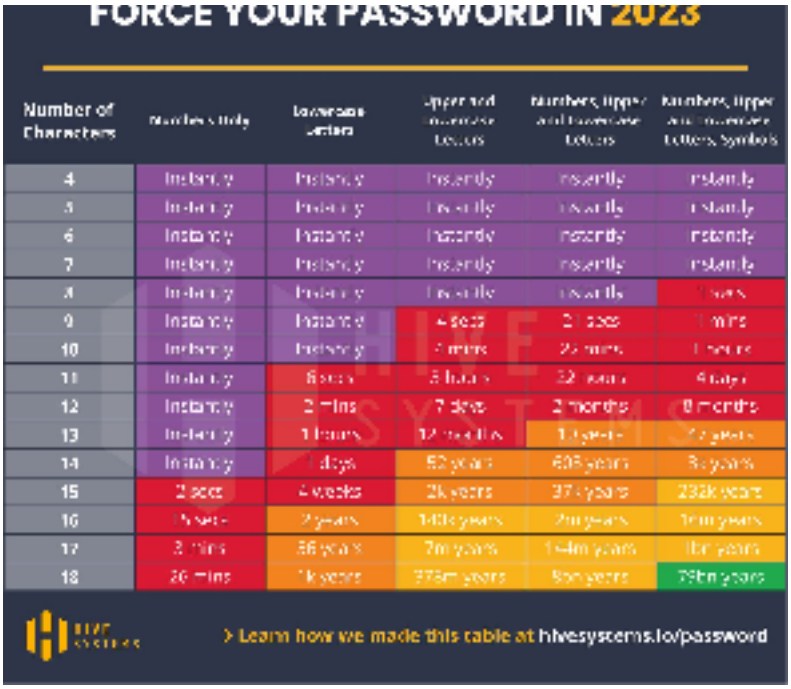
Comparison of an A100 x8 x12 and x10,000 GPUs calculations per second and hashes per second.

For comparison, take a look at how all of these are stacking up NOW:



Max time required to crack 8-character MD5 password hashes of various complexity.

And if you've been wondering about how bad ChatGPT will be on the password cracking industry, look no further than this grim table:



Time it takes the same hardware for ChatGPT to crack passwords.

"Wait... who still uses MD5 to hash their passwords in 2023?"

Good point! GPUs can generate MD5 hashes very quickly, while other hashing functions make the process slow by design. Though there are a surprising number of sites still using MD5, let's give people the benefit of the doubt and assume they hash all their passwords with bcrypt instead. bcrypt also has salting built-in and ends up being stronger than salting MD5 (and many other hash implementations).

Password tables comparing bcrypt hashes cracked by the RTX 3090 against 8 A100 GPUs from Amazon AWS

bcrypt	Calculations per second (FP32 aka float) FLOPS	Hashes per second (H/s)
RTX 4090	82,580,000,000,000	184,000
8 x A100s	155,920,000,000,000	1,081,800

Comparison of RTX 4090 and 8 A100s calculations per second and hashes per second.

In bcrypt terms, the consumer GPU hardware can only handle 184,000 hashes per second, while the EC2 instance with the 8 A100 GPUs handles 1,081,800.

"So how did you pick just one of these to be 'the 2023 Password Table'?"

We reviewed password data breaches from 2007 to present, reported through HaveIBeenPwned, to see what attackers have actually been trying to crack and whether that changed over time. Generally speaking, website logins that people probably care less about, like forums and restaurants, used and continue to use MD5 and SHA-1. That is a pretty big deal assuming people reuse the same passwords on more sensitive sites like banking, government, private messaging, email, and social media.

Password storage solutions like LastPass, 1Password, and Bitwarden use a hashing approach called PBKDF2 with a strong hash alternative to MD5 called SHA-256. Even NIST recommends PBKDF2 SHA-256. But we also found that things look different "in the wild." Breached password hashes

like PBKDF2. Bcrypt also seems to be the more secure option in terms of resources required to crack it.

Until we see more PBKDF2 or bcrypt implementations we figure it is best to stick with MD5 for this year's password table. If the site you are wondering about discloses which hashing implementation they use then see the respective table for that hash.

As a result, the 2023 Hive Systems Password Table is based on the power of the RTX 4090 with 12 GPUs against MD5. We hope in future years we stop seeing the use of MD5 and push the purple back to the top!

"What about the elephant in the room: what if my password has been previously stolen, uses simple words, or I reuse it between sites?"

Our password table focuses on the idea that the hacker is working in a "black box" situation and is having to start from scratch to hack your hash. Through the use of rainbow tables, dictionary attacks, and previously stolen hashes, your password table may (unsurprisingly) look like this:

5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	Instantly	Instantly	Instantly
8	Instantly	Instantly	Instantly	Instantly	Instantly
9	Instantly	Instantly	Instantly	Instantly	Instantly
10	Instantly	Instantly	Instantly	Instantly	Instantly
11	Instantly	Instantly	Instantly	Instantly	Instantly
12	Instantly	Instantly	Instantly	Instantly	Instantly
13	Instantly	Instantly	Instantly	Instantly	Instantly
14	Instantly	Instantly	Instantly	Instantly	Instantly
15	Instantly	Instantly	Instantly	Instantly	Instantly
16	Instantly	Instantly	Instantly	Instantly	Instantly
17	Instantly	Instantly	Instantly	Instantly	Instantly
18	Instantly	Instantly	Instantly	Instantly	Instantly

Password table if your password has been previously stolen, uses simple words, or if you reuse it between websites.

Limitations of Our Work

- Cracking passwords this way assumes that the attacker has acquired a hash digest of one or more passwords, such as those found in password data breaches on HaveIBeenPwned or more recently LastPass!
- The implied attack assumes that MFA is not used or has been bypassed. If you can get access to download the encrypted database, like what happened with LastPass, you don't need to deal with MFA when making attempts thereafter.
- These metrics assume that passwords are randomly generated. Non-randomly generated passwords are much easier and faster to crack because humans are fairly predictable. As such, the time frames in these tables serve as a "best case" reference point. Passwords that have not been randomly generated would be cracked significantly faster.

breached passwords before bothering to crack new ones.

- Think of how vast the LastPass breach was. 30 million customers' secrets were stolen. But as of publishing this article, you won't find the LastPass breach represented in HaveIBeenPwned. Why? Because the trove of passwords hasn't surfaced in public yet! Imagine how many stolen secrets and vulnerabilities never reach the light of day or even the dark web. I'd speculate keeping secrets secret gives more leverage and power to criminals than releasing them.
- Hashcat defaults to 999 iterations for PBKDF2 SHA-256 but that doesn't represent what people use. NIST recommends a minimum of 10,000 iterations and sites like LastPass (now) use 600,000, and 1password 650,000 iterations.
- Hashing a bunch of character combos is only one step to "cracking." The second step is looking for matches between the hashed strings and the breached hashed password dataset. We assume that this lookup requires a trivial amount of additional computation and time.
- Last time we included all QWERTY keyboard symbols but this year we stuck with the set commonly accepted on most websites and generated by most password generators ^*%$! &@# . That choice only impacts the last column of our tables.

Encoding	Alias	Character	Characters

~~ASCII~~	~~Uppercase~~	~~A-Z~~	~~26~~
ASCII	Numbers	0-9	10
ASCII	Symbols A	^*%$!&@#	8
~~ASCII~~	~~Symbols B~~	~~\s?/.>,<`~|;:]}[{"\"< /td>~~	~~19~~
~~Unicode~~	~~Latin Set~~	~~u00A1-u00A2, u00A4-u00FF~~	~~93~~
~~Unicode~~	~~Latin Ext A~~	~~u0100-u017F~~	~~128~~
~~Unicode~~	~~Latin Ext B~~	~~u0180-u024F~~	~~208~~
~~Unicode~~	~~Latin Ext C~~	~~u2C60-u2C7F~~	~~32~~
~~Unicode~~	~~Latin Ext D~~	~~uA720-uA7FF~~	~~29~~
~~Unicode~~	~~Cyrillic Uppercase~~	~~u0410-u042F~~	~~32~~
~~Unicode~~	~~Cyrillic Uppercase~~	~~u0430-u044F~~	~~32~~

Acknowledgements

- Thank you everyone who commented on last year's Password Table here on the site, on Reddit, Twitter, YouTube, via email and everywhere else!
- Thank you **@Chick3nman512** for answering our questions and sanity checking our hashcat results!
- Thank you **Roger K** for checking our math and helping us realize we forgot to put in the right table for this year!

References

- Hashes per second (H/s) benchmarks were either generated by Hive Systems using hashcat, or were collected from Github repo/

- We obtained GPU hardware specs from the manufacturer or www.techpowerup.com/gpu-specs.

Want to see tables from past years?

- 2022 Hive Systems Password Table
- 2020 Hive Systems Password Table

Corey Neskey

https://www.hivesystems.com/corey-neskey

Comments (20)

Most Liked

Valentin 11 months ago · 1 Like

Why should we strive for the green, when some orange times would have to have been started in the stone age to finish today ? Is it an attempt to account for the rise in processing power over time ?

Alex Nette 11 months ago · 1 Like

Correct! Processing power is increasing at an incredible pace. What was orange this year, may be red or purple just next year.

It would be great to see the results when using pass phrases. Three unrelated words, with a number and special character. Any plans on that as passphrases become more widely used?

Alex Nette 11 months ago · 0 Likes

Great question Jonathan! The good news is that in bruteforcing, the process is likely to cross every conceivable combination possible - including passphrases. So the Password Table does cover them too!

Is it 123? 11 months ago · 1 Like

Fantastic piece of work - the generation of 'the table' and this detailed explanation of what went into it. Thanks!

question 6 months ago · 0 Likes

if i cannot reuse passwords between sites, then can i just add single symbols from reused password like '1234' and '$1234'?

Alex Nette 6 months ago · 0 Likes

Hi there - generally that's not a great idea because you've created an "algorithm" of sorts that a hacker could use to guess your other passwords. For example, if you make your password for Facebook "ilikec4tzfacebook" then a hacker can probably guess that your bank account password is "ilikec4tzbankname" - not good!

I see you using “k” for Kilo, as in “17k years”, which I like.
But why aren't you consistently using then
“M” (capital) for Mega,
“G” for Giga (which is your american “bn”), and
“T” for Tera (which is your american “tn”)?

Alex Nette 6 months ago · 0 Likes

Hi Cha0waev - "k" in this case represents "thousands" on our table. Why Americans do it this way instead of using metric notations is probably on a Wikipedia page somewhere and beyond our knowledge!

Mike 9 months ago · 0 Likes

Great stuff! In the section on alternative hashes such as 'PBKDF2', you state:

"Until we see more PBKDF2 or bcrypt implementations we figure it is best to stick with MD5 for this year’s password table. If the site you are wondering about discloses which hashing implementation they use then see the respective table for that hash."

The last sentence in the paragraph is, "If the site you are wondering about discloses which hashing implementation they use then see the respective table for that hash."

Where are those other tables? I entered my contact info and looked at the contents of the password table zip file you provided via the download link you sent, but I don't see those other tables that are based on other hashes. I'm looking for the PBKDF2 version of your table (since that's what LP uses), and I'm interested in seeing what those numbers are for their PBKDF2 based passwords".

Please let me know. Thank you!

Hi Mike - check out our latest ACT post about this! https://www.hivesystems.io/blog/examining-the-lastpass-breach-through-our-password-table

---

Varinder Kumar 10 months ago · 0 Likes

Awesome work - no doubt why MFA's are must and as we progress further , Passwordless mechanisms would be the key factor for sites - specifically for ecommerce sites.

Jean-Luc GARNIER 11 months ago · 0 Likes

That's soo great! Congrats for this brilliant piece of work, that both reveals the weakness of most our (family 😉) passwords but also educates on the methods under the hood! 👏

Charles 11 months ago · 0 Likes

Any websites worth their cryptographic salt (ha! cryptography joke!) don't use MD5 anymore. Is it time to move on and base these tables off SHA 256 so people get a more realistic picture of risk? I know we want people to be security conscious, but at this point, this is almost misleading. You updated everything else for 2023 except the hashing method, which ought also to be updated. Make sure you're promoting 2FA. My two cents.

Alex Nette 11 months ago · 0 Likes

That's just the problem Charles - we're still seeing data breaches where the passwords use MD5. And with ~70% of people still reusing passwords, that

Marty 11 months ago · 0 Likes

Hi great infographic here thank you. What is unclear is if this is for cracking passwords to get into a stolen laptop for example or encrypted usb key, where you can have infinite attempts? or is this relevant to hackers trying to login to web based services - where you would usually have a small number of attempts before being kicked out? thanks, Marty

Tess de Wollaston 11 months ago · 0 Likes

Hello -- thank you so much for your article. I'm working on a screenplay, and would be very grateful to ask your advice on how long it would take to crack a specific password; I clicked on 'meet with an expert', but was unable to contact you because I have a gmail address. I'd really appreciate your help -- is there any way I could speak to you privately to discuss my questions? Thanks very much.

Alex Nette 11 months ago · 0 Likes

Hi Tess - yes please feel free to email us at information[@]hivesystems[.]io!

John Smith 11 months ago · 0 Likes

1) This whole article about the methodology of the chart...still doesn't describe the methodology of the specific chart being used. Which combinations of assumptions is used for the main chart at the top of the page? None of the charts shown in intermediate steps in the methodology match the main chart.

2) Why does no part of this guide go into the color coding choices used? 85k years to crack a password is still orange, not yellow or green? 6 BILLION years is still yellow? How do these colors make any sense?

Hi "John" - great points and feedback! For 1, we've added an additional sentence above to capture that. For 2, the colors are magnitudes of 100,000. Worryingly we saw an 8x decrease in times from just last year. So what was orange in 2023, may be red or purple in 2024. Just a good reminder to future proof your passwords!