Моя программа

Пароль	SHA-1 (секунд)	MD5 (секунд)	bcrypt (часов)	Argon2 (часов)
легкий	~1200	~2500	>1 часа	>1 часа
средний	0.5	1.25	>1 часа	>1 часа
сложный	5.02	11.71	>1 часа	>1 часа
очень сложный	~1200	~2500	>1 часа	>1 часа

Hashcat

.\hashcat.exe -a 3 -d 2 -m 100 'hash' -O --increment -w3

Пароль	SHA-1 (секунд)	MD5 (секунд)	bcrypt (часов)	Argon2 (часов)
легкий	2.92	3.14	>1 часа	>1 часа
средний	3.03	3.10	0.205	0.712
сложный	2.36	3	>1 часа	>1 часа
очень сложный	2.91	2.99	>1 часа	>1 часа s

- Легкий - 123456
- Средний - apple
- Сложный - pow12

- Очень сложный - Create

SHA-1 и MD5 имеют примерно одинаковую устойчивость, атака с помощью hashcat заняла одинаковое время, с помощью моей программы разница всего в 2 раза. bcrypt и argon2 имеет очень высокую устойчивость к перебору и требуют очень много времени для атаки как с помощью моей программы, так и с помощью hashcat.

Время подбора при использовании одинакового набора символов зависит от длины самого пароля и насколько близко символы пароля расположены друг к другу. Поэтому “средний“ и “сложный” пароли подбираются быстрее чем легкий, так как они короче.

Профессиональная утилита hashcat очевидно оказалась намного эффективнее моей программы, так как выполняет хэширование напрямую на gpu с множеством оптимизаций.