ECE 498KL: eCrime and Internet Service Abuse

Basic Password Attacks and Defenses

Kirill Levchenko

November 13, 2018

ILLINOIS
Electrical & Computer Engineering
COLLEGE OF ENGINEERING

Passwords

- **Password:** (a) secret known to a user that is (b) used to authenticate the user to a system (c) by supplying it to the system when challenged
- Authenticating system needs to be able to verify that the supplied password is correct

ID	Name	Passw. routine	Accounts with passw.	Leak date
1	000webhost.com	`$p`	15 035 687	≈ Mar. 2015
2	17.media	`md5($p)`	3 824 575	≈ Sep. 2015
3	51cto.com	`md5(md5($p).$s), md5($p)`	3 923 449	≈ Dec. 2013
4	7k7k.com	`$p`	9 231 185	≈ Oct. 2011
5	aipai.com	`md5($p)`	4 529 928	≈ Apr. 2011
6	ashleymadison.com	`bcrypt($p)`	36 140 796	≈ July 2015
7	badoo.com	`md5($p)`	122 730 419	≈ June 2016
8	csdn.net	`$p`	6 425 905	≈ Oct. 2011
9	duduniu.cn	`$p`	14 192 866	≈ Aug. 2011
10	gawker.com	`des($p)`	487 292	≈ Dec. 2010
11	gmail.com	`$p`	4 925 994	≈ Sep. 2014
12	imesh.com	`md5(md5($p).$s)`	51 308 651	≈ Sep. 2013
13	ispeak.cn	`$p`	8 294 278	≈ Apr. 2011
14	linkedin.com	`sha1($p)`	112 275 414	≈ Feb. 2012
15	mail.ru	`$p`	5 269 103	≈ Sep. 2014
16	mate1.com	`$p`	27 402 581	≈ Feb. 2016
17	mpgh.net	`md5(md5($p).$s)`	3 119 180	≈ Oct. 2015
18	myspace.com	`sha1($p)`	358 986 419	≈ 2008
19	naughtyamerica.com	`md5($p)`	989 401	≈ Apr. 2016
20	nexusmods.com	`md5(md5($s).md5($p))`	5 918 540	≈ Dec. 2015
21	r2games.com	`md5(md5($p).$s), md5($p)`	11 758 232	≈ Oct. 2015
22	renren.com	`$p`	4 392 208	≈ Nov. 2011
23	sprashivai.ru	`$p`	3 472 645	≈ May 2015
24	taobao.com	`$p`	14 769 995	≈ Jul. 2015
25	tianya.cn	`$p`	29 642 564	≈ Nov. 2011
26	twitter.com	`$p`	26 121 984	≈ June 2016
27	vk.com	`$p`	92 144 526	≈ 2012
28	weibo.com	`$p`	4 529 994	≈ Dec. 2011
29	xiaomi.com	`md5(md5($p).$s)`	8 281 358	≈ May 2014
30	xsplit.com	`sha1($p)`	2 990 112	≈ Nov. 2013
31	yandex.ru	`$p`	1 186 565	≈ Sep. 2014
Total accounts with email addr.: 994 301 846 , **Total distinct email addr.:** 884 460 979				

Table 1: Analyzed identity leaks ($p - clear password, $s - salt)

```
Function bcrypt
   Input:
      cost:     Number (4..31)                    log2(Iterations). e.g. 12 ==> 2^12 = 4,096 iterations
      salt:     array of Bytes (16 bytes)         random salt
      password: array of Bytes (1..72 bytes)      UTF-8 encoded password
   Output:
      hash:     array of Bytes (24 bytes)

   //Initialize Blowfish state with expensive key setup algorithm
   state ← EksBlowfishSetup(cost, salt, password)

   //Repeatedly encrypt the text "OrpheanBeholderScryDoubt" 64 times
   ctext ← "OrpheanBeholderScryDoubt"  //24 bytes ==> three 64-bit blocks
   repeat (64)
      ctext ← EncryptECB(state, ctext) //encrypt using standard Blowfish in ECB mode

   //24-byte ctext is resulting password hash
   return Concatenate(cost, salt, ctext)
```

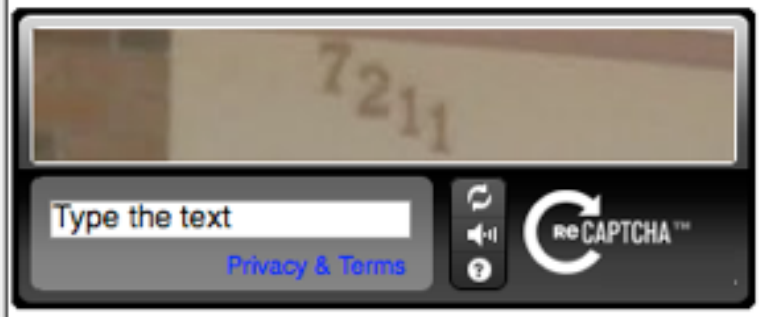
CrackStation

Defuse.ca · Twitter

CrackStation » Password Hashing Security » Defuse Security »

Free Password Hash Cracker

Enter up to 10 non-salted hashes:



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5), md5-half, sha1, sha1(sha1_bin()), sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+

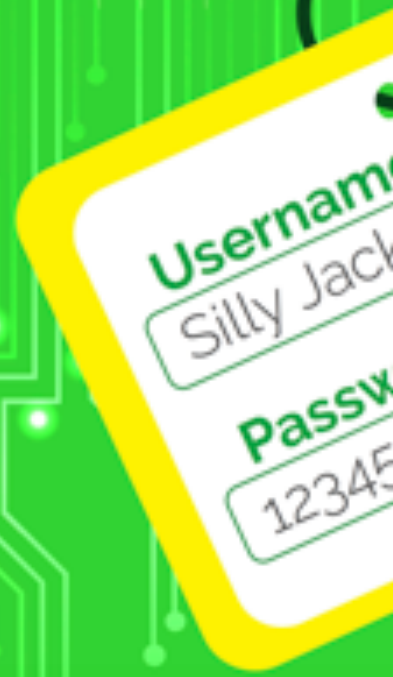
Download CrackStation's Wordlist

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our hashing security page.

Crackstation's lookup tables were created by extracting every word from the Wikipedia databases and adding with

WORST PASSWORDS OF 2015



SplashData releases its annual list in an effort to encourage the adoption of stronger passwords to improve Internet security. The passwords evaluated are mostly from North American and Western European users. The list shows many **people continue to put themselves at risk for hacking and identity theft** by using weak, easily guessable passwords.

RANK	PASSWORD	CHANGE FROM 2014
1	123456	Unchanged
2	password	Unchanged
3	12345678	1 ↗
4	qwerty	1 ↗
5	12345	2 ↘
6	123456789	Unchanged
7	football	3 ↗
8	1234	1 ↘
9	1234567	2 ↗
10	baseball	2 ↘
11	welcome	NEW
12	1234567890	NEW
13	abc123	1 ↗
14	111111	1 ↗
15	1qaz2wsx	NEW
16	dragon	7 ↘

"123456" and **"password"** once again reign supreme as the most commonly used passwords

Some longer passwords are so simple as to make their extra length virtually **worthless**