Analyzing rule efficiency in a hybrid dictionary attack

Passcape Software

1 Analyzing rule efficiency in a hybrid dictionary attack

1.1 Abstract

When it comes to password decryption, a hybrid dictionary attack is a very flexible and indispensable tool. Unlike a common dictionary attack, which uses a custom though, but nevertheless a fixed set of word mutation rules, a hybrid attack can be completely adjusted according to your needs.

Many users when decrypting passwords use their own password mutations rules, starting hybrid dictionary attack several times with different settings. For example, the first attack starts with the most frequently used and limited set of rules for searching popular passwords quickly, then there goes a hybrid attack rules to locate non-Latin passwords, and then a slow attack with a full set of rules for finding out all possible combinations, etc. By the way, a full collection of more than 600000 mutation rules is available at the following link: hybrid_all.ini

1.2 Password generation rules

Many advanced researchers when decrypting passwords use their own password mutations rules, starting hybrid dictionary attack several times with different settings. For example, the first attack starts with the most frequently used and limited set of rules for searching popular passwords quickly, then there goes a hybrid attack to locate non-Latin passwords, and then a slow attack with a full set of rules for finding out all possible combinations, etc. By the way, a full collection of more than 600000 mutation rules is available at the following link: hybrid_all.ini

What is the most effective rule-file in a given situation? The problem is, some sets of rules were made to search for passwords based on popular mutations, others for finding certain types of passwords. For example, like these ones:

overwrite.ini, insert.ini - search for passwords with replaced or inserted character
numbers.ini - search for passwords with numbers at the beginning or end of a word
simple_dates.ini - search for passwords with dates
nonenglish_words.ini - mutation of non-Latin passwords
l33t.ini - for passwords written in leet, like password -> p@$$w0rd
dotcom.ini - domains at the end of words

1.3 Rules efficiency

1.3.1 Overall efficiency

We tried to abstract from the fact that some sets are specific to a certain type of passwords, and tried to evaluate the effectiveness of each rule-file that comes with our Windows

Password Recovery tool. We made a list of **30819** NTLM hashes taken from the **Defcon Crack Me if You Can 2010** contest. Then sequentially started the hybrid attack with each (one) set to evaluate their effectiveness, check the working time, etc. The attack was carried out using the built-in wpr.pcd dictionary consisting of **5398185** words. Used hardware - Nvidia GTX 1060. The table below shows the list of the most effective sets, sorted by the maximum number of found items.

Rule-file name	Passwords found	% found	Number of rules	Work time (sec)	Found pwds per rule	Speed (found pwds/sec)
hybrid_all.ini	11248	36,5%	616716	3487	0,02	3,23
Hashcat_generated2.ini	7552	24,5%	304062	1775	0,02	4,25
nsa.ini	7270	23,6%	123286	552	0,06	13,17
english_words.ini	6758	21,9%	30055	208	0,22	32,49
Hashcat_d3ad0ne.ini	6492	21,1%	35323	189	0,18	34,35
InsidePro-HashManager.ini	5785	18,8%	6469	28	0,89	206,61
yurets.ini	5724	18,6%	74585	394	0,08	14,53
Hashcat_rockyou-30000.ini	5324	17,3%	29999	139	0,18	38,30
Hashcat_T0XlCv1.ini	5130	16,6%	11936	53	0,43	96,79
InsidePro-PasswordsPro.ini	4819	15,6%	3119	16	1,55	301,19
d3adhob0.ini	4764	15,5%	57536	308	0,08	15,47
Hashcat_generated.ini	4531	14,7%	14726	71	0,31	63,82
Hashcat_T0XlC.ini	3687	12,0%	4086	21	0,90	175,57
overwrite.ini	3239	10,5%	3	7	1079,67	462,71
insert.ini	3174	10,3%	3	7	1058,00	453,43
fasthash.ini	2112	6,9%	2943	16	0,72	132,00
Hashcat_T0XlC-insert_00-99_1950-2050_toprules_0_F.ini	2105	6,8%	4015	19	0,52	110,79
megatron-1.ini	1994	6,5%	450	2	4,43	997,00
megatron-2.ini	1804	5,9%	10664	50	0,17	36,08
Hashcat_best64.ini	1661	5,4%	77	1	21,57	1661,00

Rule-file name	Passwords found	% found	Number of rules	Work time (sec)	Found pwds per rule	Speed (found pwds/sec)
numbers.ini	1457	4,7%	650	5	2,24	291,40
simple_dates.ini	1300	4,2%	4828	32	0,27	40,63
Hashcat_combinator.ini	1287	4,2%	39	2	33,00	643,50
Hashcat_toggles5.ini	1215	3,9%	4943	26	0,25	46,73
Hashcat_toggles4.ini	1178	3,8%	1940	10	0,61	117,80
Hashcat_toggles3.ini	1153	3,7%	575	3	2,01	384,33
Hashcat_toggles2.ini	1136	3,7%	120	1	9,47	1136,00
Hashcat_toggles1.ini	1107	3,6%	15	2	73,80	553,50
Hashcat_T0XlC-insert_space_and_special_0_F.ini	1023	3,3%	480	2	2,13	511,50
Hashcat_specific.ini	997	3,2%	176	1	5,66	997,00
Hashcat_Incisize_leetspeak.ini	869	2,8%	15487	120	0,06	7,24
Hashcat_T0XlC-insert_top_100_passwords_1_G.ini	856	2,8%	1600	10	0,54	85,60
nonenglish_words.ini	824	2,7%	4448	82	0,19	10,05
hashcat_ninja_leetspeak.ini	733	2,4%	2047	15	0,36	48,87
Hashcat_leetspeak.ini	704	2,3%	17	2	41,41	352,00
l33t.ini	583	1,9%	1046	11	0,56	53,00
Hashcat_oscommerce.ini	135	0,4%	256	2	0,53	67,50
dotcom.ini	127	0,4%	40	2	3,18	63,50

The first column shows the number of passwords found by the given set of rules. The second displays the number of found passwords in percentage. The third column shows the total number of rules for the file. In the fourth, the time took to verify all the rules. The next two columns are a little bit more interesting and output the most effective rules. More specifically,

the number of found passwords for a single rule and the number of found passwords per second.

Quite funny to see the significant superiority of some sets (for example, *insert.ini* and *overwrite.ini*) and the failure of others, such as *Hashcat_Incisize_leetspeak.ini*. But this is understandable, taking into account that they were mainly designed to search for passwords of certain types. In general, the results can be useful for those who face strict time frames when recovering passwords. For instance, a good balanced rule set is *InsidePro-HashManager.ini*, which has found **5785** passwords in only **28** seconds. For comparison, the set of all mutation rules found **11248** password for ~ **1 hour** using the above-mentioned hardware.

1.3.2 Top 10 best rule-sets

Top 10 best rule-sets (sorted by the number of found passwords)

Rule-file name	Passwords found	% found
hybrid_all.ini	11248	36,5%
Hashcat_generated2.ini	7552	24,5%
nsa.ini	7270	23,6%
english_words.ini	6758	21,9%
Hashcat_d3ad0ne.ini	6492	21,1%
InsidePro-HashManager.ini	5785	18,8%
yurets.ini	5724	18,6%
Hashcat_rockyou-30000.ini	5324	17,3%
Hashcat_T0XlCv1.ini	5130	16,6%
InsidePro-PasswordsPro.ini	4819	15,6%

1.3.3 Top 10 biggest rule-sets

Top 10 biggest rule-sets (by the number of rules)

Rule-file name	Number of rules
hybrid_all.ini	616716
Hashcat_generated2.ini	304062
nsa.ini	123286
yurets.ini	74585
d3adhob0.ini	57536
Hashcat_d3ad0ne.ini	35323
english_words.ini	30055
Hashcat_rockyou-30000.ini	29999
Hashcat_Incisize_leetspeak.ini	15487
Hashcat_generated.ini	14726

1.3.4 Top 10 most efficient rule-sets

Top 10 most efficient rule-sets (by the number of found passwords per rule)

Rule-file name	Found passwords per rule
overwrite.ini	1079,67
insert.ini	1058,00
Hashcat_toggles1.ini	73,80
Hashcat_leetspeak.ini	41,41
Hashcat_combinator.ini	33,00
Hashcat_best64.ini	21,57
Hashcat_toggles2.ini	9,47
Hashcat_specific.ini	5,66
megatron-1.ini	4,43
dotcom.ini	3,18

1.3.5 Top 10 fastest rule-sets

Top 10 fastest rule-sets (by the number of found passwords in second)

Rule-file name	Found passwords in second
Hashcat_best64.ini	1661,00
Hashcat_toggles2.ini	1136,00
Hashcat_specific.ini	997,00
megatron-1.ini	997,00
Hashcat_combinator.ini	643,50
Hashcat_toggles1.ini	553,50
Hashcat_T0XlC-insert_space_and_special_0_F.ini	511,50
overwrite.ini	462,71
insert.ini	453,43
Hashcat_toggles3.ini	384,33