I have the HASHCAT so I make the rules.

August 2014
Yiannis Chrysanthou
Information Security Advisor
@yiannistox

$ whoami

Name: ***Yiannis Chrysanthou***

Work and education

- Information Security Advisor at KPMG's UK Cyber Response Team
- MSc Information Security
  Thesis : ***Modern Password Cracking, a hands-on approach to creating an optimised and versatile attack***

Fun facts

- Member of Team Hashcat
- DEFCON competition - "***Crack me if you can***"
- Positive Hack Days competition "***Hashrunner***"
- Dan Gooding from Ars Technica , said that I cracked this password :
  "***Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn1.***"
- Full article : "***How the Bible and YouTube are fuelling the next frontier of password cracking***"
- Interviewed by Mark Ward on BBC news Technology

Agenda

1. Problem definition

we are here

" Cracking the last 10% of your hashes can be exponentially harder than the first 90%"

2. Attempting to solve the problem

" New, existing and improved tools and techniques to help break the 80% barrier"

2.1. Hashcat overview

2.2. Advanced wordlist generation

2.3. Advanced rule generation - *with Morph & Tmesis tools*

2.4. Count-words as input to advanced combinator attacks with Tmesis

2.5. Combining N-grams with Tmesis

Questions?

1. Problem definition

A password cracking attempt is successful if it cracks 92%+ of passwords every time.

Does this sound familiar?

- You already tried to brute force it
- You already used all wordlists (private/public)
- You already cracked 85-90% of a hashlist
- You need to crack the rest within the least time and with the least resources
- You're stuck with nothing else to try

I thought we're already cracking loads of passwords, do we need more?"
We always need MORE!!!

Agenda

1. Problem definition

“ Cracking the last 10% of your hashes can be exponentially harder than the first 90%“

2. Attempting to solve the problem ← we are here

“ New, existing and improved tools and techniques to help break the 90% barrier“

- 2.1. Hashcat overview
- 2.2. Advanced wordlist generation
- 2.3. Advanced rule generation - *with Morph & Tmesis tools*
- 2.4. Count-words as input to advanced combinator attacks with Tmesis
- 2.5. Combining N-grams with Tmesis

Questions?

2. Attempting to solve the problem

New, existing and improved tools and techniques to help break the 80% barrier

My 5 cents for today:

- I use Hashcat, it’s and awsome , really fast CPU and GPU based password cracking tool and with active community
- I make my own rules, and I constantly improve them
- I make my own wordlists and I constantly improve them
- I use and contribute towards the creation of Hashcat utils
- In today’s presentation I suggest the usage of Tmesis in combination with other existing utilities and methods

Agenda

2.1 Hashcat Suite Toolset overview

Tools within the suite

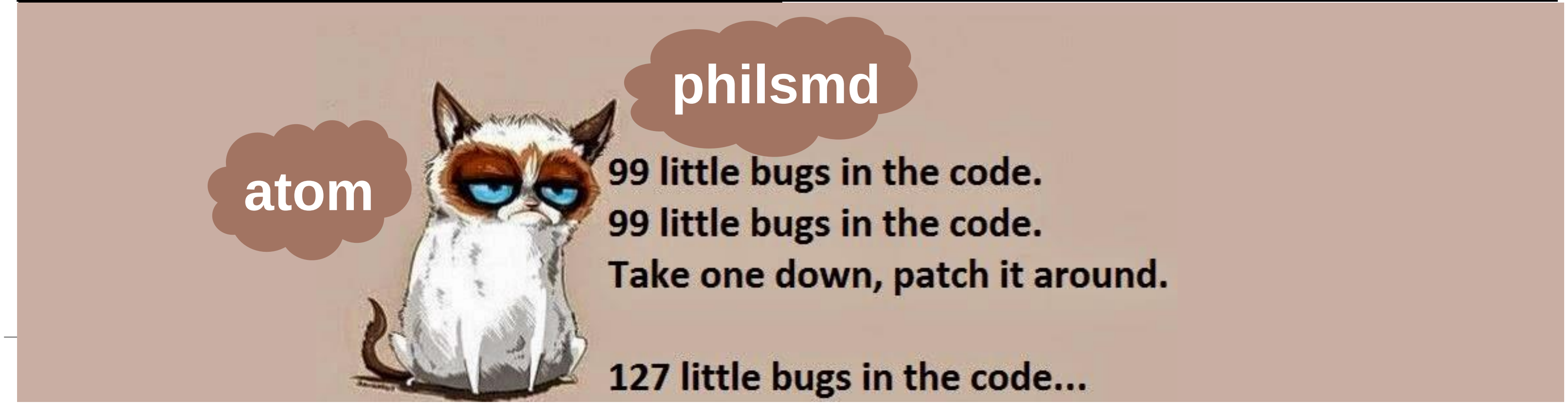
- hashcat-0.47.7z (CPU)
- oclHashcat-1.21.7z (GPU)
- hashcat-utils-1.0.7z
- maskprocessor-0.70.7z
- statsprocessor-0.083.7z

Oclhashcat attack modes

- Brute-Force attack
- Combinator attack
- Dictionary attack
- Fingerprint attack
- Hybrid attack
- Mask attack
- Permutation attack
- Rule-based attack
- Table-Lookup attack
- Toggle-Case attack

Hashcat utils

- expander
- len
- morph
- rli
- combinator
- cutb
- **tmesis.pl**
- maskprocessor
- StatsProcessor
- **Countwords.pl**


philsmd
atom
99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.
127 little bugs in the code...

Agenda

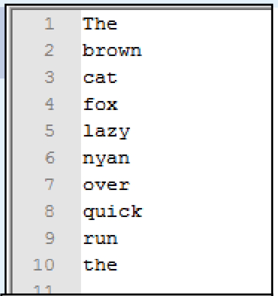
2.2 Advanced wordlist generation

You can create wordlists from <u>ANYTHING! Example sources:</u>

- *Website content*
- *Online bible*
- *Movie scripts*
- *Poems and song lyrics*
- *Ebooks (free) and whitepapers*
- **Previously cracked passwords**
- Wikipedia offers **free copies of all content in 283 languages** available to download!!!
- IRC logs are a great **source of slang words** keeps an archive of all logs since 2004!!!
- **Pastebin posts** are an amazing source for worldlists and pastebin keeps archives you can collect!!!

Command to make a wordlist from a bunch of text:

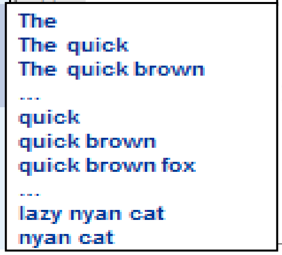
```
$ echo “The quick brown fox run
over the lazy nyan cat” | tr “ ” “\n” | sort –u
```



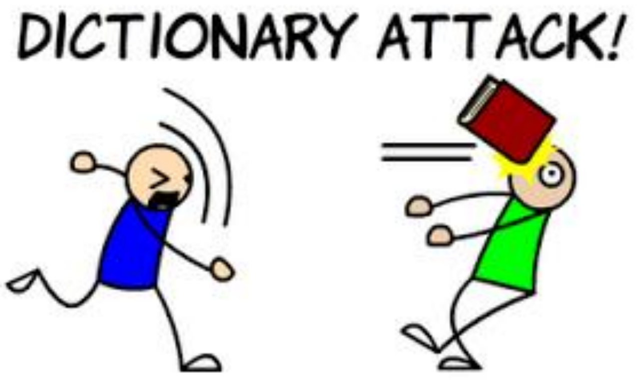
```
The
brown
cat
fox
lazy
nyan
over
quick
run
the
```

PERL scripts to make wordlists of phrases **CMIYC 2012**:

http://hashcat.net/CMIYC2012/phraser1.pl
http://hashcat.net/CMIYC2012/phraser2.pl



```
The
The quick
The quick brown
...
quick
quick brown
quick brown fox
...
lazy nyan cat
nyan cat
```



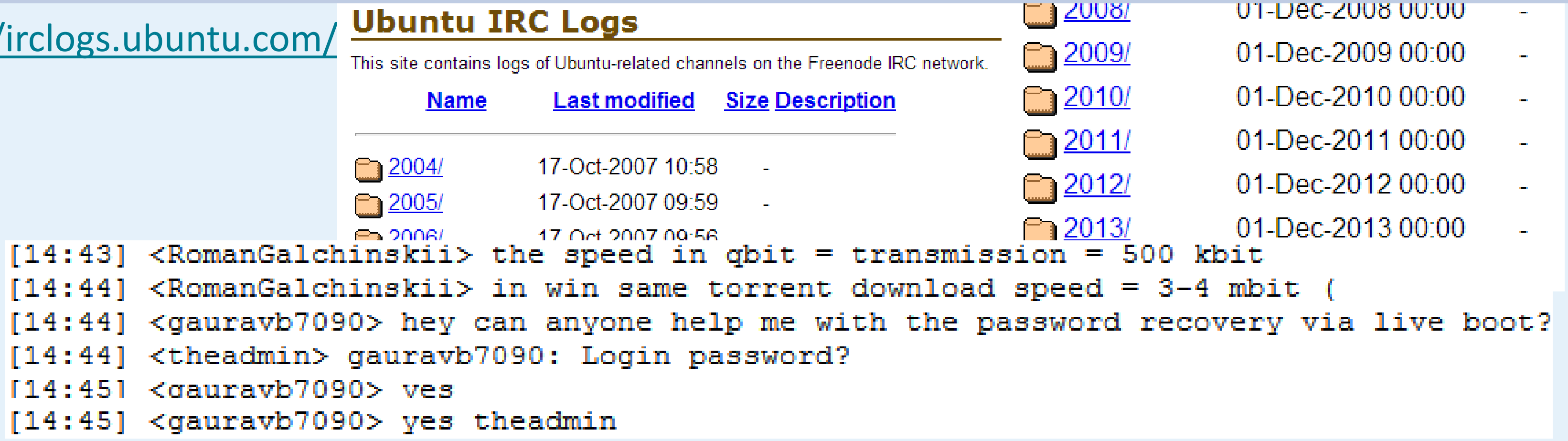
2.2 Advanced wordlist generation - continued

Wikipedia content download in 283 languages

http://en.wikipedia.org/wiki/Wikipedia:Database_download

IRC chat logs download since 2004

http://irclogs.ubuntu.com/



Ubuntu IRC Logs

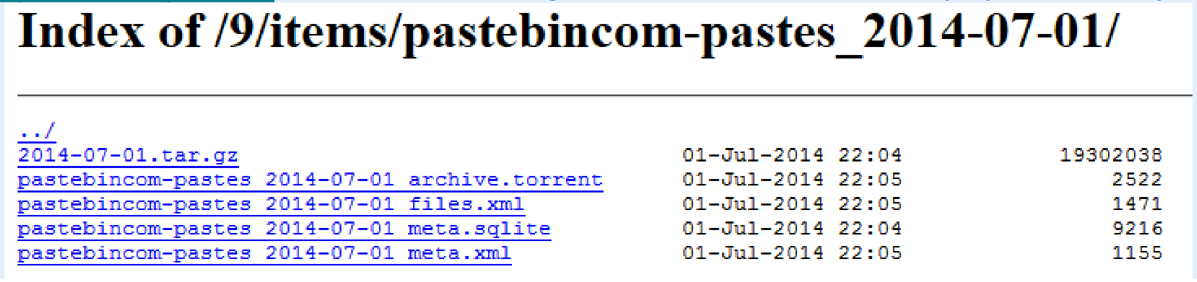
This site contains logs of Ubuntu-related channels on the Freenode IRC network.

```
Name        Last modified       Size Description
2004/       17-Oct-2007 10:58    -
2005/       17-Oct-2007 09:59    -
2006/       17-Oct-2007 09:56
2008/       01-Dec-2008 00:00    -
2009/       01-Dec-2009 00:00    -
2010/       01-Dec-2010 00:00    -
2011/       01-Dec-2011 00:00    -
2012/       01-Dec-2012 00:00    -
2013/       01-Dec-2013 00:00    -
```

```
[14:43] <RomanGalchinskii> the speed in qbit = transmission = 500 kbit
[14:44] <RomanGalchinskii> in win same torrent download speed = 3-4 mbit (
[14:44] <gauravb7090> hey can anyone help me with the password recovery via live boot?
[14:44] <theadmin> gauravb7090: Login password?
[14:45] <gauravb7090> yes
[14:45] <gauravb7090> yes theadmin
```

Pastebin posts since 2011

https://archive.org/details/pastebinpastes **No more IP blocking / PROXY / TORSOCKS to scrape pastebin ! Yay !**



Index of /9/items/pastebincom-pastes_2014-07-01/

```
../
2014-07-01.tar.gz                               01-Jul-2014 22:04          19302038
pastebincom-pastes_2014-07-01_archive.torrent   01-Jul-2014 22:05              2522
pastebincom-pastes_2014-07-01_files.xml         01-Jul-2014 22:05              1471
pastebincom-pastes_2014-07-01_meta.sqlite       01-Jul-2014 22:04              9216
pastebincom-pastes_2014-07-01_meta.xml          01-Jul-2014 22:05              1155
```

2.2 Advanced wordlist generation - continued

By looking at passwords that were cracked using wordlists made of sources such as IRC, Wikipedia and Pastebin, the following conclusions were made:

- People ❤ passphrases
- People reuse their passwords and/or password creation patterns
- Someone’s username is someone else’s password* (HT epixoip)
- People are resourceful with passphrases and often make them personal
- **The source of the hashes becomes obvious once you start cracking the first passwords.**

A lot of users use pass-phrases ! Examples:

4772625698c6c1ed00afd0848fd2b57b:	Password must be at least 8 characters
ce950a8d7d367b5ce038e636893b49dc:	Yellow fruit that is popular among monkeys
4cee2c84f6de6d89a4db4f2894d14e38:	**this is not my real admin login**
02f26cba22e2fa9e07008d65782437de:	**look at my horse my horse is amazing**
9e2261652addceb69aca13e7e16331f9:	ComplexComplexComplex
00000eb6875515d8be2c055876ed4915eacd9141:	**iampasswordprotected**

2.2 Advanced wordlist generation - continued

A lot of users use pass-phrases ! Even more examples ☺
4fee893423be6b007094e3294692d961:alapdanceissomuchbetterwhenthestripperiscrying
7bd89bc713a2cdc74f9c12560fc58d43:**LyingIsTheMostFunAGirlCanHaveWithoutTakingHerClothesOff**
00000336feda5262d01afe6be86eef6069e5fd77:fuckedin.com
9eedfd733a63806076fbb639e99277161ba13fec:iwanttomeetmyhusband
2af3d0afeb0db07b8683dcd45d478671:iliketochangemypassword
3bc6ef741fe71f3f549b54dac165ab1d:**darlingiwanttodestroyyou**
2fd84862ee1da894d211114e056205f7:**iwanttodivorcemyhusband!**
d37542e426e66998c76dd6657bf141b0:**ifucked5goats**
9c89f38b7213d4ffe86d62f1bc451a09:**beautifulnakedwomenin3d**
00000ec247015b4be8961075506afae3447a4ee0:dancansuckmycock

Agenda

1. Problem definition

“ Cracking the last 10% of your hashes can be exponentially harder than the first 90%“

2. Attempting to solve the problem

“ New, existing and improved tools and techniques to help break the 80% barrier“

2.1. Hashcat overview

2.2. Advanced wordlist generation

2.3. Advanced rule generation - *with Morph & Tmesis tools* **we are here**

2.4. Count-words as input to advanced combinator attacks with Tmesis

2.5. Combining N-grams with Tmesis

Questions?

2.3 Advanced rule generation – why make rules

Everyone has "rules" for their passwords:

"Most secure passwords include phrases" → ***I love pancakes***
"No, it's better to use multiple words stuck together" → ***sexypassword***
"Just use L33T T3X7 !!!" → ***i4mL33t&l34tn00b5***
"Add smilies/funny faces to your passwords" → ***[-_-] securepass***
"Use your keyboard letter sequence" → ***q1w2e3r4t5y6u7i8o9p0***

2.3 Advanced rule generation - with HashcatUtils morph

How to use morph for generating targeted rulesets:

1. On a new list of hashes, run one pass with generic rulesets and dictionaries
2. Use the first batch of cracked passwords as input for morph to create custom rules and crack even more passwords

Cracked Passwords
123456
qwerty**123**
123pass
abc**pass**word
1!1**pass**23
1234561**23**

Morph

Rule	Explanation
i01 i12 i23	Insert at position 0 **(1)** position 1 **(2)** position 2 **(3)**
i61 i72 i83	Insert at position 6 **(1)** position 7 **(2)** position 8 **(3)**
i4p i5a i6s i7s	Insert at position 4 **(p)** position 5 **(a)** position 6 **(s)** position **(s)**
i72i83	Insert at position 7 **(2)** position 8 **(3)**

MORE Passwords
123Destiny
123desperado
buddy123
abcpassword
darkpass24

2.3 Advanced rule generation - with HashcatUtils Tmesis (new)

How does Tmesis work?

Tmesis will create rules that insert contents of one wordlist into all positions on another wordlist. The input worldlist source can be anything. (Dates , numbers , special characters ….)

For example:

1. Input wordlist contains one word: “password”
2. Destination wordlist contains one word: “123456”
3. Tmesis will make Hashcat rules that insert “password” at each possible position within “123456” and this will result in the following password candidate words:

password123456
1password23456
12password3456
123password456
1234password56
12345password6
123456password

One way of using Tmesis for generating targeted rulesets:

1. On a new list of hashes, run one pass with generic rulesets
2. Use the first batch of cracked passwords as input for Tmesis to create insertion rulesets to be used with Hashcat

2.3 Advanced rule generation – importance of custom root words

- cat crackedpasswords.txt | tr –d [:digit:] | tr –d [:punct:] | sort | uniq –ic | sort -rn

LinkedIn	last.fm	MangaTraders	Seals With Clubs
Interesting root Words			
linkedin = 6048	lastfm = 1229	manga = 474	password = 1105
link = 3390	last = 659	love = 280	poker = 688
linked = 2759	lfm = 2759	anime = 278	qwerty = 470
alex = 1492	abc = 1492	querty = 258	qwer = 444
mike = 1391	LastFm = 1391	naruto = 257	xxxxxx = 386
june = 1262	leo = 1262	dragon = 255	seals = 192
password = 1240	lol = 1240	sakura = 254	bitcoin = 166
love = 1225	last = 1225	mangatraders = 178	sealswithclubs = 105
john = 1157	alex = 1157	shadow = 171	pokemon = 104
Linkedin = 1093	Love = 1093	april = 165	last = 1225

Top **alpha** root words found during password analysis

2.3 Advanced rule generation – importance of custom root words

```
cat crackedpasswords.txt | tr –d [:a-z:] | tr –d [:A-Z:] | sort | uniq –ic | sort -rn
```

LinkedIn	last.fm	MangaTraders	Seals With Clubs Bitcoin Poker
interesting non-letter words			
123 = 59964	1 = 57874	1 = 21407	1 = 4252
01 = 54694	123 = 19258	123 = 11613	123 = 3457
12 = 54694	12 = 15068	12 = 6441	12 = 1083
1234 = 14923	. = 10351	1234 = 2336	12345 = 360
@ = 10884	_ = 8387	666 = 1912	123456789 = 211
! = 10529	= 6086	= 1413	123123 = 208
2011 = 9860	! = 5755	! = 1241	111111 = 190
2008 = 8994	@. = 5410	_ = 1219	4444 = 167
. = 8964	- = 5205	- = 1115	. = 66
123 = 59964	@ = 5133	. = 1069	! = 22

Top **non-alpha** root words found during password analysis

2.3 Advanced rule generation – importance of custom root words

The two previous lists of top root words (alpha and non-alpha) were combined using below command allowing us to crack much more complex passwords as shown below:

- **./combinator.bin root_words_alpha.txt top_mutations.txt | ./oclhashcat.bin –r sexy.rules –o crackedmassivepasswords.txt**

Sample cracked passwords

LinkedIn	last.fm	MangaTraders	Seals With Clubs
Linkedin!1	LAST!FM	mangatrad3rs	seals12345
LinkedIn!1234	LAST!fm	mangatrada12345	sealsbitcoin
linkedin!2011	LAST!fm4me	mangatrader-123	sealssux
linkedin!42	LAST#FM#99	mangatraders!@QWASZX	sealsum
linkedin!^	LAST#fmmusic	mangatraders0Pass	sealsw7
LinkedIn!	LAST%FM	mangatraders4fun	sealswithclubs7

2.3 Advanced rule generation – Tmesis in action

TOP ROOT WORDS
4408 linkedin
2130 linked
1311 l1nk3d1n
1044 Linkedin
856 password
732 LinkedIn

tmesis

Rule	Explanation
i0l i1i i2n i3k i4k i5e i6d i7i i8n	**linkedin123456**
i1l i2i i3n i4k i5k i6e i7d i8i i9n	**1linkedin23456**
	12linkedin3456
i2l i3i i4n i5k i6k i7e i7d i9i iAn	**123linkedin456**
	1234linkedin56
i3l i4i i5n i6k i7k i8e i9d iAi iBn	**12345linkedin6**
	123456linkedin

New Cracked Passwords	
tri**linkedin**pod	MY**l1nk3d1n**p4$$w0rd
tijl**Linkedin**01	trustno1@**l1nk3d1n**
Whiskey**linkedin**pass99	my**l1nk3d1n**l0g1n
wachtwoord**linkedin**2011	EDF**l1nk3d1n**2011
notimportant**password**1980	**l1nk3d1n**_Passw0rd

Agenda

we are here

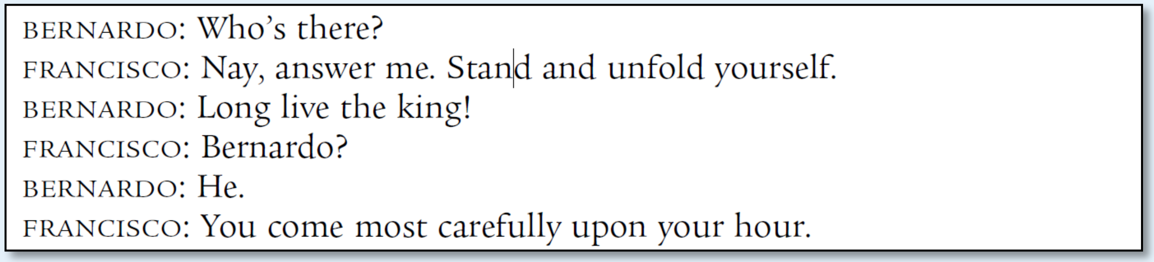
2.4 Count-words as input to advanced combinator attacks with Tmesis

Count words (count-words.pl):

"Count words" can take any input and create a list of most common (by occurrence) pairs of words.

Example:

Use all pdfs with literature with as input



BERNARDO: Who's there?
FRANCISCO: Nay, answer me. Stand and unfold yourself.
BERNARDO: Long live the king!
FRANCISCO: Bernardo?
BERNARDO: He.
FRANCISCO: You come most carefully upon your hour.

and run command:

"index of/" pdf shakespeare → PDFtoTXT

Sample count-worlds.pl outputs:

2 Word pair	Occurrences
"of the "	1417909
"in the "	787392
"to the "	761271
"on the "	606695
"for the "	536111

3 Word pair	Occurrences
"one of the "	454511
"to use the "	435369
"the number of "	441356
"be able to "	440477
"part of the "	438941

2.4 Count-words as input to advanced combinator attacks with Tmesis

Traditional Combinator attacks can append/prepend and the word pairs as shown below:

2 Words pair
"of the " or "ofthe"
"in the " or "inthe"
"to the " or "tothe"
"on the " or "onthe"
"for the " or "forthe"

Combinator with wordlists →

New Cracked Passwords
ofthedark123456
ofthepeoplebythepeopleonline
oftheopera1234556
oftheprincess1995
ofthespotlessmind

3 Words pair
"one of the " or "oneofthe"
"to use the " or "tousethe"
"the number of " or "thenumberof"
"be able to " to "beableto"
"part of the " or "partofthe"

Combinator with wordlists →

New Cracked Passwords
oneofthetribe971
tousethe1nt3rn3t
thenumberofthebeast42
beabletowin3
partofthebeautyofme

2.4 Count-words as input to advanced combinator attacks with Tmesis

Tmesis can create more complex rulesets for inserting word pairs at any position within a string and crack complex passwords.

Word pair
“of the ” or “ofthe”
“in the “ or “inthe”
“to the “ or “tothe”
“on the “ or “onthe”
“for the “ or “forthe”

tmesis

rule	New Cracked Passwords
i9o iAf iBt iCh iDe	axelQueen**ofthe**Night
i8i i9n iAt iBh iCe	M2lcolm**inthe**middle
i4t i5o i6t i7h i8e	back**tothe**future3-d
i4o i5n i6t i7h i8e	secz**onthe**beach89
i8f i9o iAr iBt iCh iDe	linkedin**forthe**book

3 Word pair
“one of the ” or “oneofthe”
“to use the “ or “tousethe”
“the number of “ or “thenumberof”
“be able to “ to “beableto”
“part of the “ or “partofthe”

tmesis

New Cracked Passwords
the**oneofthe**best123
trainacat**tousethe**toilet
unveiling**thenumberof**thebeast
2**beableto**see
lam**partofthe**1%

2.4 Count-words as input to advanced combinator attacks with Tmesis

Combining Combinator and Tmesis for the ultimate attack. Allowed for cracking 30+ character passwords with alpha/numeric/symbol.

Combinator	Tmesis rule	New Cracked Passwords
mybabyily + iloveyou	-r tmesis(Ilove) -r (1qaz2wsx)	**mybabyililoveyil1qaz2wsxoveyou**
0.123456 + abcd123+	-r abcd123 –r superman	**0.123456abcd123abcsupermand123+**
1492 + shadow	-r dragon -r 1492	14**dragon**92**shadow**1492

Agenda

2.5. Using N-grams with Tmesis

N-grams can help predict the next item in a sequence, for example a Markov chain: If the probability that “the ***nth*** character of a word is ***x***”, can be defined as a function dependable on the previous ***n-1*** character, then we have a Markov Chain.

The below per-position 2-grams and 3-grams were extracted from RockYou and can be the input of Tmesis for creating new rulesets to better attack remaining passwords

2-grams	
pair	**Occurrences**
an	1216714
ar	834574
er	795998
ma	745282
in	741061
12	735937
19	650542
el	588617
on	581316
al	575873

3-grams					
pair	**Occurrences**	**pair**	**Occurrences**	**pair**	**Occurrences**
mar	88621	and	44381	ama	29705
ove	62930	cha	43118	lly	29696
lov	62729	ill	42225	nny	24827
ilo	54507	lil	42212	ris	23264
ove	54001	123	42052	and	22526
lov	51904	83	41914	lli	22076
ari	50539	ell	36159	amo	21325
sha	48340	ara	35296	gel	20934
85	45572	aby	35053	lle	19441
87	44472	ani	33950	lla	18732

2.5. Using N-grams with Tmesis

Sample cracked password with N-grams + Tmesis attack

3brob**mar**7
iloveo**mar**forever1
acs**mar**taudi123
rosaAzulo**mar**9
them**mar**egistry
themyan**mar**laws
jeanky**mar**reroc
Zhal**mar**1987
Zhda**mar**0211
allen_**mar**gaux27

Agenda

we are here

Questions?

@yiannistox