Generating Personalized Wordlists With Natural Language Processing by Analyzing Tweets

Utku Sen
utku@utkusen.com

Abstract

Adversaries need to have a wordlist or combination-generation tool while conducting password guessing attacks. To narrow the combination pool, researchers developed a method named ”mask attack” where the attacker needs to assume a password’s structure. Even if it narrows the combination pool significantly, it’s still too large to use for online attacks or offline attacks with low hardware resources. Analyses on leaked password databases showed that people tend to use meaningful English words for their passwords, and most of them are nouns or proper nouns. Other research shows that people are choosing these nouns from their hobbies and other interest areas. Since people are exposing their hobbies and other interest areas on Twitter, it’s possible to identify these by analyzing their tweets. Rhodiola tool is developed to narrow the combination pool by creating a personalized wordlist for target people. It finds interest areas of a given user by analyzing his/her tweets, and builds a personalized wordlist.

1 Introduction

Passwords are our main security mechanism for digital accounts since the beginning of the internet. Because of that, passwords are one of the main targets of attackers. There are couple of major ways that an attacker can use to find a target’s password. The attacker can prepare a phishing website to trick a target into entering their passwords to a rogue website. Or, an attacker can conduct a password guessing attack through brute forcing. Password guessing attacks can be described in two main categories: online attacks and offline attacks.

Online password guessing attack is where the attacker sends username/password combinations to a service like HTTP, SSH etc. and tries to identify the correct combination by checking the response from the services. An offline password guessing attack is usually conducted against hashed forms of passwords. The attacker has to calculate a password’s hash with a suitable cryptographic hashing function and should compare it with target hash. For both online and offline attacks, the attacker usually needs to have a password wordlist.

Most of the web applications have password complexity rules where users have to use at least one number, upper/lower case letters and a special character. Also, Active Directory, which is widely used in enterprises forces users to use complex passwords. Therefore, reducing the brute force pool to an acceptable size is very important for attackers.

2 Mask Attacks

Mask attack is one of the main methods for reducing the brute force pool to an acceptable size. Mask attack refers to specifying a fixed password structure and generating candidate passwords according to that. For example, to crack ”Julia1984” as a password, we need to calculate 13.537.086.546.263.552 different combinations. But if we set a mask with its structure, we can reduce the combination pool to 237.627.520.000.[1]

In the real world, a password’s structure is an unknown value, just like the password itself. To guess the password’s structure, we need to understand human behaviour regarding designing passwords. Leaked password databases can be analyzed to see common masks used by people. The following statistics were found when the leaked Ashley Madison wordlist[2] was analyzed with PACK (Password Analysis and Cracking Kit)[3]

```
Password Length:
8: 24%
6: 19%
7: 18%
9: 13%
10: 9%
```

```
Character-set:
loweralphanum: 47%
loweralpha: 33%
numeric: 12%
mixedalphanum: 2%
upperalphanum: 1%

Popular Masks:
?l?l?l?l?l?l: 7%
?l?l?l?l?l?l?l?l: 7%
?l?l?l?l?l?l?l: 6%
?l?l?l?l?l?l?d?d: 4%
?l?l?l?l?l?l?l?l?l: 4%
```

Following statistics are found when leaked Myspace wordlist[2] is analyzed with PACK:

```
Password Length:
8: 22%
7: 22%
9: 17%
6: 15%
10: 14%

Character-set:
loweralphanum: 75%
loweralphaspecial: 7%
loweralpha: 6%
upperalphanum: 2%
loweralphaspecialnum: 2%

Popular Masks:
?l?l?l?l?l?l?d: 7%
?l?l?l?l?l?l?l?d: 6%
?l?l?l?l?l?l?d?d: 5%
?l?l?l?l?l?l?l?l?d: 5%
?l?l?l?l?l?d?d: 4%
```

By checking these statistics, we can observe that passwords are usually designed with sequential alphabetic characters.

The combination pool of "?l?l?l?l?l?l"(?l refers to loweralpha) mask is 308.915.776. It's not a big pool for an offline attack with good hardware resources, but it's still big for online attacks and offline attacks with poor hardware resources. Even if we specify a password structure with masks, we are still brute forcing items in the mask. To narrow our combination pool more, we need to create meaningful outcomes from sequential alpha characters and numbers without brute forcing them.

3 Part-of-speech Analysis of Leaked Password Databases

Since both Ashley Madison and Myspace wordlists are mostly consists of sequential alpha characters, there is a high probability that they are meaningful words. If they are somehow meaningful, we can fill the mask with meaningful words instead of brute forcing the characters.

The first step is understanding if a letter sequence is a meaningful word in the English language. We can state that a letter sequence is an English word if it's listed in an English lexicon. Wordnet (a lexical database for English created by Princeton University)[4] is used as the lexicon.

After it's confirmed that the letter sequence is included in Wordnet, hence it's an English word, we need to do part-of-speech tagging (POS tagging). There are eight parts of speech in the English language: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection[5]. POS tagging is the process of marking up a word in a text as corresponding to a particular part of speech.[6] NLTK Python library is used for POS tagging.[7]

To understand which part of speech is usually located in human-designed passwords, we've analyzed two popular leaked password databases: Ashley Madison and Myspace. The code that was used to analyze named "word_classifier.py" is located under Rhodiola tool's directory.

Part-of-speech details of Myspace Wordlist:

```
Wordlist size: 37127
64% (23830) Non-English word or
    Unidentified string
32% (12181) Singular Noun (NN)
3% (1409) Plural Noun (NNS)
1% Other POS-tags and Digit-only: Adverb(RB
    ), Determiner(WDT), Adjective(JJ), Verb
    (VB) etc.
```

Part-of-speech details of Ashley Madison Wordlist:

```
Wordlist size: 375853
58% (218753) Non-English word or
    Unidentified string
27% (102903) Singular Noun (NN)
12% (46313) Digit-only
2% (8532) Plural Noun (NNS)
1% Other POS-tags: Adverb(RB), Determiner(
    WDT), Adjective(JJ), Verb(VB) etc.
```

According to these data, 30% of the Ashley Madison database and 36% of Myspace database contains meaningful English words and most of them are a singular noun. If we use all words in the Oxford English Directory, the combination pool will be 171,476[8]. If we use

”?l?l?l?l?l?l” mask to brute force all six-character alphabetic strings, the combination pool will be 308.915.776. So, trying all English words in dictionary would be 1801 times faster than using a mask. But 171,476 is a still big number for online attacks. We can reduce this number if we can identify what kind of words are usually chosen by people.

According to experiments conducted by Carnegie Mellon and Carleton universities, most people are choosing words for their passwords based on personal topics such as hobbies, work, religion, sports, video games, etc.[9][10] So if we can identify the candidate words from interest areas of a person, we can reduce the combination pool significantly.

On Twitter, people tend to share posts mostly related to their area of interest.[11] Because of that, Twitter is a good candidate to identify a user’s personal topics and generate related words about it to reduce the combination pool for password guessing attacks.

4 Analyzing User Tweets

Twitter’s API allows us to download a user’s latest 3200 tweets.[12] Therefore, our analyze will be limited to that threshold.

4.1 Cleaning Tweet Data

Since our goal is to identify a user’s personal topics and generate related words about it, we need to remove unnecessary data (stop words) from downloaded tweets. Both NLTK’s stopwords extension and a custom list are used.[13] Lists contains high-frequency words like ”the,a,an,to,that,i,you,we,they”. These words are removed before processing the data.

4.2 Identifying Most Used Nouns and Proper Nouns

As shown in the previous section, almost 30% of the user passwords are consists singular nouns. Therefore, our first goal is to identify the most used nouns and proper nouns. The topics that the user is interested most can be identified with them. The most used nouns and proper nouns are identified with NLTK’s POS tagging function.[14]. The following example shows the most used nouns and proper nouns of @elonmusk[15] user on Twitter.

Words that identified as noun and their occurrences are: (car’, 49), (falcon’, 34), (rocket’, 26), (mars’, 12), (earth’, 12), (flamethrower’, 11), (production’, 10), (dragon’, 10), (live’, 9), (amazing’, 9)

Words that identified as proper noun and their occurences are: (’tesla’, 117), (’falcon’, 32), (’spacex’, 28), (’boring’, 21), (’earth’, 10), (’mars’, 9), (’california’, 8), (’hyperloop’, 8), (’roadster’, 8), (’china’, 7), (’easter’, 7), (’north’, 7)

4.3 Pairing Similar Words

In some cases, nouns can be used together. To create meaningful word pairs, we need to analyze their semantic similarities. For this purpose, NLTK’s path_similarity[16] is used with the first noun meaning (n.01) on Wordnet for all identified nouns. The *path_similarity* returns a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy. The score is in the range 0 to 1. Our algorithm pairs any two nouns if their similarity score is higher than 0.12.

4.4 Finding Related Helper Words

Researchers have found that some of the most used semantic themes in passwords are locations[17] and years.[18]. Therefore, related locations and years to a user’s interest areas should’ve been found. Wikipedia[19] is used for both works. Our algorithm visits each proper noun’s Wikipedia page and parses years with regex and identifies city names with its hardcoded city list[20].

5 Rhodiola Tool

Rhodiola is written in Python 2.7[21] and mostly based on NLTK and textblob[22] libraries. With a given Twitter handle, it can automatically can compile a personalized wordlist with the following elements: Most used nouns&proper nouns, paired nouns&proper nouns, cities and years related to detected proper nouns.

5.1 Usage

Before using Rhodiola, Twitter API keys should be gathered from the Twitter Developer Platform[23] and written inside the code. Rhodiola has three different usage styles: base, regex and mask.

In the base mode, Rhodiola takes a Twitter handle as an argument and generates a personalized wordlist with the elements which are described in the previous section. Example usage:

```
python rhodiola.py --username elonmusk
```

Example output:

```
...
tesla
car
```

```
boring
spacex
falcon
rocket
mars
earth
flamethrower
coloradosprings
tesla1856
boringcompany2018
...
```

In the regex mode, a user can generate additional strings with the provided regex. These generated strings will be appended as a prefix or suffix to the words. For this mode, Rhodiola takes a regex value as an argument. ”regex_place” defines the string placement and is optional (default value is suffix). Example command:

```
python rhodiola.py --username elonmusk --
    regex "(root|admin)\d{2}" --regex_place
     suffix
```

Example output:

```
...
teslaroot01
teslaroot02
teslaroot03
...
spacexadmin01
spacexadmin02
spacexadmin03
...
tesla1856root99
...
boringcompany2018admin99
...
```

In the mask mode, user can provide hashcat style mask values. [1] Only \l (lower-alpha) and \u (upper-alpha) charsets are available. Example command:

```
python rhodiola.py --username elonmusk --
    mask "?u?l?u?u?l"
```

Example output:

```
...
TeSLa
CaR
BoRIng
SpACex
FaLCon
RoCKet
MaRS
EaRTh
FlAMethrower
CoLOradosprings
TeSLa1856
BoRIngcompany2018
...
```

6 Conclusion

Since people tend to use words from their interest areas for their passwords and expose those interest areas on Twitter, it’s possible for an attacker to create a wordlist by analyzing a target’s tweets. Beyond Twitter, any actor that has much more data about a person will have an ability to create more accurate wordlists. Therefore, users should avoid using words from the topics that are exposed in social media. It’s better to use random passwords that are stored in a password manager software.

References

[1] mask_attack [hashcat wiki]. (n.d.). Retrieved from https://hashcat.net/wiki/doku.php?id=mask_attack

[2] Passwords - SkullSecurity. (n.d.). Retrieved from https://wiki.skullsecurity.org/Passwords

[3] iphelix/pack. (2018, 8). Retrieved from https://github.com/iphelix/pack

[4] WordNet — A Lexical Database for English. (n.d.). Retrieved from https://wordnet.princeton.edu/

[5] The Eight Parts of Speech - TIP Sheets - Butte College. (n.d.). Retrieved from http://www.butte.edu/departments/cas/tipsheets/grammar/parts_of_speech.html

[6] POS tags and part-of-speech tagging — Sketch Engine. (n.d.). Retrieved from https://www.sketchengine.eu/pos-tags/

[7] Natural Language Toolkit NLTK 3.4 documentation. (n.d.). Retrieved from https://www.nltk.org

[8] How many words are there in the Engli... — Oxford Dictionaries. (n.d.). Retrieved from https://en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language/

[9] Stobert, E., & Biddle, R. (2014). The Password Life Cycle: User Behaviour in Managing Passwords.

[10] Ur, B., Noma, F., Bees, J., Segreti, S., Shay, R., Bauer, L., Christin, N. (2015). I Added ! at the End to Make It Secure: Observing Password Creation in the Lab.

[11] Michelson, M., & Macskassy, S. (2010). Discovering users' topics of interest on twitter: a first look.

[12] GET statuses/user_timeline. (n.d.). Retrieved from https://developer.twitter.com/en/docs/tweets/timelines /api-reference/get-statuses-user_timeline.html

[13] 2. Accessing Text Corpora and Lexical Resources. (n.d.). Retrieved from https://www.nltk.org/book/ch02.html

[14] 5. Categorizing and Tagging Words. (n.d.). Retrieved from https://www.nltk.org/book/ch05.html

[15] Elon Musk (@elonmusk) on Twitter. (n.d.). Retrieved from https://twitter.com/elonmusk

[16] nltk.corpus.reader.wordnet NLTK 3.4 documentation. (n.d.). Retrieved from http://www.nltk.org/_modules/nltk/corpus/reader/ wordnet.html

[17] Medlin, B. D., Cazier, J. A. (2007). An empirical investigation: Health care employee passwords and their crack times in relationship to hipaa security standards.

[18] Veras, R., Thorpe, J., Collins, C. (2012). Visualizing semantics in passwords: The role of dates.

[19] Wikipedia. (n.d.). Retrieved from https://wikipedia.org

[20] elyase/geotext. (n.d.). Retrieved February 6, 2019, from https://github.com/elyase/geotext/blob/master/geotext /data/cities15000.txt

[21] Welcome to Python.org. (n.d.). Retrieved from https://python.org/

[22] TextBlob: Simplified Text Processing TextBlob 0.15.2 documentation. (n.d.). Retrieved from https://textblob.readthedocs.io/en/dev/

[23] Twitter Developer Platform. (n.d.). Retrieved from https://developer.twitter.com/content/developer-twitter/en.html