SSN Project Proposal: Practical Passphrase Cracking

Dirk Gaastra · Lennart van Gijtenbeek · Luc Gommans

September 21st, 2017

1. Introduction

For as long as users have shared computer systems, people have implemented authentication methods. Most commonly, this was in the form of a username and password: the username identifies a user and the password authenticates. Since computers are becoming ever faster, people need to choose ever stronger passwords. Where shorter passwords provided sufficient protection in the past, it is now recommended to use completely random strings of at least twelve characters. However, memorizing twelve or more random characters is not reasonable for most people.

A passphrase is proposed as an alternative that is easier to remember. They are short strings, consisting of a few random (dictionary) words. Only a few words can already be as strong as a strong password. Most of the research into the topic has been on passwords rather than passphrases, and most of the cracking tools only work well for passwords. In this work, we will attempt practical attacks on passphrases, aiming to further the research in this field. The approach we take in this investigation, is to reduce the passphrase search space by incorporating personal data about the target, in order to generate more likely passphrases (using a self-implemented cracking tool).

2. Research Questions

We have outlined several research questions in this section. The investigation consists of both theoretical and practical aspects. Please see these question below:

1. To what degree [quantify] do users incorporate interests in their passphrases?
2. How feasible is it for an attacker to crack a passphrase when a victim's interests are known?
3. Can we significantly reduce the passphrase search space using an automated tool?
4. What resources are required to generate such a database of interests (storage) / brute a passphrase given a database (CPU)?

3. Scope

Previous research has been done into predictability of passphrases (see section 7) and tools have been built which can be used to crack passwords. To the best of our knowledge, there exists no practical passphrase cracking tool which can be seeded with user's interests and/or hobbies. For instance, if we know someone is into a specific band, we can focus our search efforts by generating passphrases that incorporate their lyrics, and thereby effectively decrease

the search space. This transforms the problem of cracking a passphrase from an impossible task (with respect to computational time) to more manageable proportions. The focus of this research will therefore be on ways to seed a passphrase cracking tool with useful information. This tool will then work out passphrases that incorporate the information that was used to seed the tool.

It is out of scope to determine the interests of users in an automated fashion. We assume that the interests of a person are given. By means of a survey we will figure out which themes are prevalent in users' passphrases. We will not attempt to create multiple, comprehensive datasets for various topics; instead, we will attempt to construct such a database for a specific prevalent topic.

4. Implementation

The implementation of the tool will probably be done in *Java*. This language is fast enough for our purposes, while at the same time providing the functionality we need. The seed database will have a custom format, that will be loaded into memory by the tool.

5. Planning

Week 1

- Perform a survey to collect passphrases and personal data.
- Set up a seed database, for 1 or 2 specific hobbies.
- Determine the design/coding plan for the cracking tool.
  - Review literature for potential good designs.

Week 2

- Start building the tool.
- At the end of this week, the tool should have basic functionality.

Week 3

- Continue improving the tool on performance and accuracy.
- Start writing on the project report (draft version) at the end of this week.

Week 4

- The tool should be finished at the start of this week.
- Extensively test the tool, using the survey data for validation purposes.
- Reflect on the research questions.
- Finalize the project report.
- Prepare presentation.

6. Ethical Considerations

For this research we will do a survey, asking people to come up with a couple of different passphrases and list things such as their hobbies, favorite bands, and favorite TV shows. For improved results, this survey will be anonymous, such that participants are encouraged to fill in the survey truthfully. Based on the data obtained through the survey, one might be able to identify individuals; however, the chance of this happening is unlikely and the impact negligible (since the passphrases are supposed to be fictitious).

We would like to see our final product (the passphrase cracker) in action in order to test it. This means that we might have to find a dataset containing passwords linked to real user accounts. Publishing the code to any tool used for cracking can have initial negative effects on the overall security of systems. However, our aim is that this will only increase the security practices of people and systems in the long run.

7. Related work

The benefits of using passphrases are discussed by [1] and [2]; these papers claim that with good design, the security of passwords increases dramatically while not increasing the amount of login failures.

In terms of research into cracking, P. Sparell et al. have looked into using Markov chains to predict the next word most often used when given a few words [3]. This research can provide us with some insight into how we can construct our passphrases.

Furthermore, A. Roa et al. have gotten interesting results when accounting for grammatical structures while cracking passwords [4]. Most interestingly, they showed that using longer passphrases did not mean they were more secure. They also managed to decrease the search space for passphrases by about 50% by being grammatically aware.

While not specific for passphrases (even though they are included), [5] gives insight into making 'educated' password guesses instead of just brute-forcing it. The paper is very extensive and will be helpful to get a better idea of how to predict passphrases.

For a more hands-on approach, [6] demonstrated that using publicly-available data, one can build a large dictionary with famous sentences which can be used to crack millions of passphrases, reaffirming previous research stating that a longer password does not equal more security.

References

[1] Mark Keith, Benjamin Shao, and Paul Steinbart. A behavioral analysis of passphrase design and effectiveness. *Journal of the Association for Information Systems*, 10(2):2, 2009.

[2] Jianxin Yan, Alan Blackwell, Ross Anderson, and Alasdair Grant. The memorability and security of passwords–some empirical results. Technical report, University of Cambridge, Computer Laboratory, 2000.

[3] Peder Sparell and Mikael Simovits. Linguistic cracking of passphrases using markov chains. *IACR Cryptology ePrint Archive*, 2016:246, 2016.

[4] Ashwini Rao, Birendra Jha, and Gananand Kini. Effect of grammar on security of long passwords. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 317–324. ACM, 2013.

[5] Charles Matthew Weir. *Using probabilistic techniques to aid in password cracking attacks*. The Florida State University, 2010.

[6] Hugo Labrande. Crack me i'm famous: cracking weak passphrases using publicly-available sources.