Password cracking

Keywords: combinatorics, probability and statistics, combinatorics, product rule

With the development of the Internet and long-distance communication went hand in hand the need to verify whether the person on the other side of the monitor is actually the person with whom we are communicating or just someone pretending to be an acquaintance. Similar to the situation, when introducing two friendly spies in a foreign territory, the possibility of using a password is offered. Today, one encounters passwords in cyberspace on a daily basis, when logging into email, school or work accounts, or online banking.

But does the mere existence of passwords guarantee secure user authentication? Ongoing reports of new hacks and stolen accounts tell us not. The methods by which attackers get to a user's password can basically be divided into two groups, depending on whether it is stolen or guessed. Since the following problem deals with the second case, let's take a closer look at it.

The brute force attack, which we will learn about in the task, consists of trying all possible passwords. Depending on the computing power of the computer and the software used, the speed of testing can range from a few thousand to several hundred billion passwords per second. Thus, very short passwords can be guessed by the computer in a relatively short time (i.e., instantly or within hours).

A more sophisticated form of brute force attack is the *dictionary attack*, where the computer does not try passwords at random, but selects them from a dictionary of prepared words. In addition to actual words, this contains commonly used passwords such as `password1234` or `password`. If the victim's password is in the attacker's dictionary, the cracking time is significantly reduced compared to a conventional brute force attack.

An essential protection against both types of attacks is the use of sufficiently long passwords (at least 12 characters) made up of upper and lower case letters, numbers and other special characters.

Figure 1: Hacking

Assignment

The hacker program, in a brute force attack, is guaranteed to crack an eight-character password made up of upper and lower case letters of the English alphabet in about 22 minutes. (Assume that the set of keyboard characters to be tested can be set in the program settings.)

Exercise 1. How many passwords does the program try in 1 second?

Solution. Since the English alphabet has 26 characters, there can be 52 possibilities in each position of an eight-character password (upper and lower case letters). Using the combinatorial product rule, we can deduce that the total number of possible passwords is equal to 52^8.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.

Funded by the European Union

The number of passwords the program tries in one second is the total

$$\frac{52^8}{22 \cdot 60} \doteq 40\ 500\ 000\ 000.$$

Exercise 2. How long would it take the program to crack an eight-character password if we also allow using digits?

Solution. By adding ten new characters, there can be 62 different characters in each position. According to the combinatorial product rule, the number of possible passwords is 62^8; using the result of the previous problem, we get the time t in which the program tries all the passwords as

$$t = \frac{62^8}{40\ 500\ 000\ 000} \doteq 5\ 391 \text{ s} \doteq 90 \text{ min}.$$

Exercise 3. How many characters would a password consisting of numbers and lowercase or uppercase letters of the English alphabet have to be strong enough, i.e. guaranteed to take at least 100 years to crack? How does the result change if we allow for the possible use of another 40 special keyboard characters?

Solution. We assume that each year has 365 days, i.e. 31 536 000 seconds. Let's denote n the required number of characters and add them as in the previous problem. But now we get an exponential equation with unknown n, which we solve:

$$\begin{aligned}
\frac{62^n}{40\ 500\ 000\ 000} &\geq 100 \cdot 31\ 536\ 000 \\
62^n &\geq 40\ 500\ 000\ 000 \cdot 3\ 153\ 600\ 000 \\
n \log 62 &\geq \log(40\ 500\ 000\ 000 \cdot 3\ 153\ 600\ 000) \\
n &\geq \frac{\log(40\ 500\ 000\ 000 \cdot 3\ 153\ 600\ 000)}{\log 62} \doteq 11{,}22
\end{aligned}$$

A password with the required security would have to be at least 12 characters long.

If we allow an additional 40 characters on the keyboard, we obtain by similar calculation a result of the form

$$n' \geq \frac{\log(40\ 500\ 000\ 000 \cdot 3\ 153\ 600\ 000)}{\log 102} \doteq 10{,}01.$$

A password with the required security must now have at least 11 characters.

References and literature

Literature

- Raza, Mudassar & Iqbal, Muhammad & Sharif, Muhammad & Haider, Waqas. (2012). A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication. *World Applied Sciences Journal*. 19. 439–444.

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.

Funded by the European Union

- National Cyber and Information Security Agency. *Bezpečný pohyb v kybersvětě* [online]. Available from `https://www.nukib.cz/cs/kyberneticka-bezpecnost/vzdelavani/verejnost/` [cit. 30. 6. 2023].

Sources of images

- Hacking password, Santeri Viinamäki, CC BY-SA 4.0, available from `https://commons.wikimedia.org/wiki/File:Hacking_password_illustration.jpg` [cit. 30. 6. 2023].

The portal Math4U is developed within the international projects of the Department of Applied Mathematics (VSB-Technical University of Ostrava) and other institutions in Czechia, Poland, Slovakia, and Spain.

Funded by the European Union