openHPI Course: Digital Identities – Who am I on the Internet?

# Password Length and its Importance

**Prof. Dr. Christoph Meinel**

Hasso Plattner Institute
University of Potsdam, Germany

# Password Length

**Password length** has a great influence on the strength/security of a password and the efficiency of possible password attacks.

**Reminder**: Notes on generating secure passwords:

- □ Upper and lower case letters
- □ Different character classes (letters, numbers, special characters ($% &:; -_? §! ...))
- □ At least 12 characters long
- □ Not from the dictionary
- □ Cannot be derived from the user context
- □ No reuse

**What are the reasons for these indications?**

# Brute Force Attacks (1/2)

**Brute force attacks** are the simplest and most straight forward attacks to crack a password

- **Idea**: Systematic testing of all possible character combinations for selected character classes at a given length
- With sufficient time resources Brute Force always leads to the goal, so to find a password
- Calculation formula for the number of all password candidates:

$$\text{Number\_of\_password\_candidates} = (\text{range\_of\_characters})^{\text{Password length}}$$

# Brute Force Attacks (2/2)

**Idea**: Systematic testing of all possible character combinations for selected character classes at a given length.

$$\text{Number\_of\_password\_candidates} = (\text{range\_of\_characters})^{\text{Password\_length}}$$

Expected value for the average number of attempts to find a password:

$$\text{Average\_number\_of\_attempts} = \text{Number\_of\_password\_candidates}/2$$

To protect against brute force attacks, the number of password candidates must be as large as possible.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

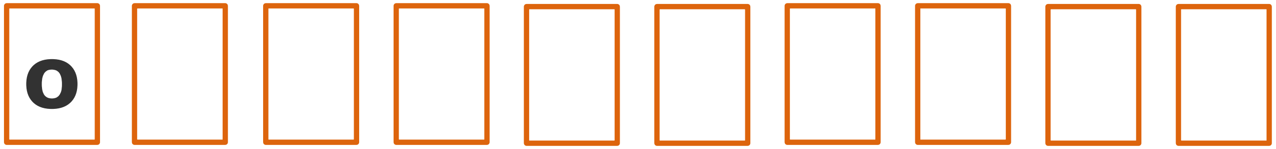
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 1**



o

36

= 36

≙ < 0,001 sec*

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the
Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

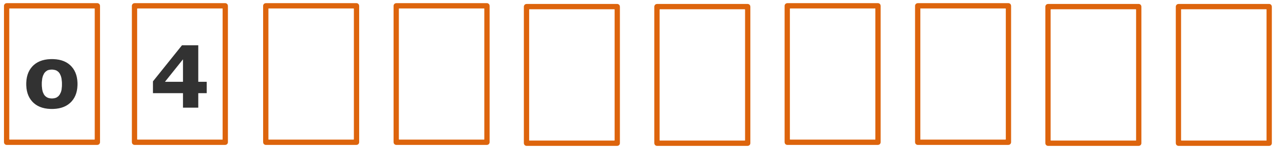
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 2**



**36 * 36 = 1.296**

**≙ < 0,001 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

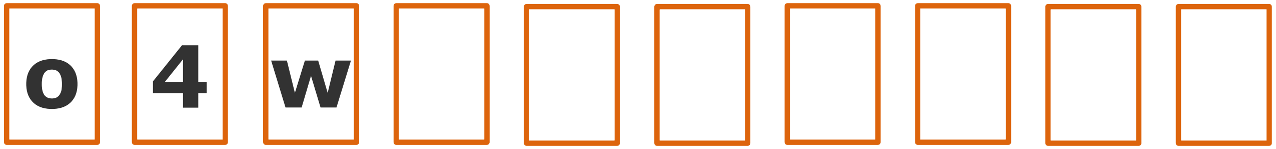
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 3**



o 4 w

**36 * 36 * 36 = 46.656**

**≙ < 0,001 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

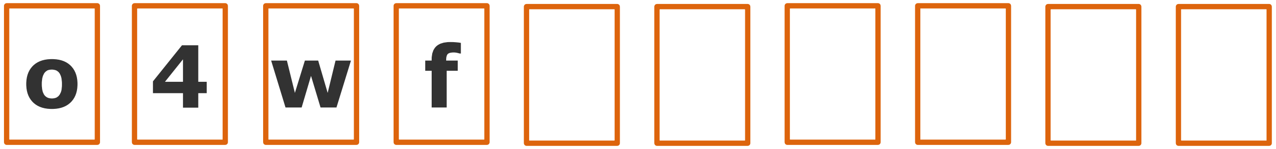
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 4**



**36 * 36 * 36 * 36** **= 1.679.616**

**≙ < 0,001 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

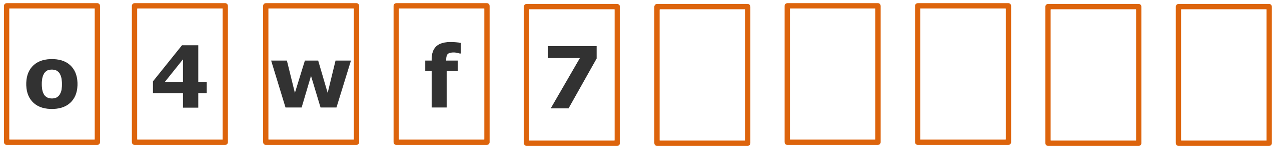
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 5**



**36 * 36 * 36 * 36 * 36 = 60.466.176**

**≙ < 0,001 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

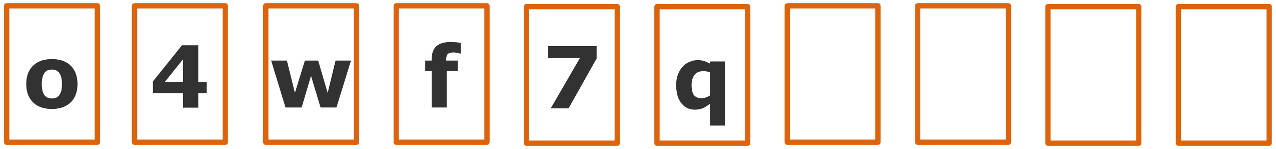
- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 6**


o 4 w f 7 q

**36 * 36 * 36 * 36 * 36 * 36 = 2.176.782.336**

**≙ ~ 0,022 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 7**

**36 * 36 * 36 * 36 * 36 * 36 * 36 = 78.364.164.096**

**≙ ~ 0,784 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 8**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **o** | **4** | **w** | **f** | **7** | **q** | **2** | **1** | | |

**36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 = 2.821.109.907.456**

**≙ ~ 28,211 sec***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 9**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **o** | **4** | **w** | **f** | **7** | **q** | **2** | **1** | **n** | |

**36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 = 101.559.956.668.416**

**≙ ~ 16,927 min***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Calculation of the
# Number of Possible Password Candidates

**Example**: Password consists of lower case letters …

- abcdefghijklmnopqrstuvwxyz: 26 possible characters

…and numbers.

- 0123456789: 10 possible characters

Number of possible characters in each position: 26 + 10 = **36**

**Password length = 10**

**o 4 w f 7 q 2 1 n t**

**36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 * 36 = 3.656.158.440.062.976**

**≙ ~ 10,156 h***

* Time required to generate all possible password candidates, when 100 billion passwords can be generated per second.

# Brute Force Attacks on Password Hashes

- Brute force attacks are often performed against password hashes
  - □ possible password candidates are hashed
  - □ generated password hash is compared with the target hash
  - □ if they match, the password candidate is the password you are looking for
- The speed of brute force attacks depends on the calculation speed of the hash function used
  - □ MD5 hashes can be calculated much faster than SHA-512 Hashes

# Cracking Complexity

| Password length until | Figures [0-9] | Numbers + lower case letters [0-9a-z]. | Alphanumeric [0-9a-zA-Z]. | Alphanumeric + Special characters 0-9a-zA-Z$% &:; -_? §!…] |
|---|---|---|---|---|
| 5 | < 1 sec | < 1 sec | < 1 sec | < 1 sec |
| 6 | < 1 sec | < 1 sec | < 1 sec | ~ 7,43 sec |
| 7 | < 1 sec | < 1 sec | ~ 35,79 sec | ~ 11,76 min |
| 8 | < 1 sec | ~ 29,02 sec | ~ 36,99 min | ~ 18,62 hours |
| 9 | < 1 sec | ~ 17,41 min | ~ 1,59 days | ~ 2,43 months |
| 10 | < 1 sec | ~ 10,45 hours | ~ 3,25 months | ~ 19,24 years |
| 11 | ~ 1 sec | ~ 2,24 weeks | ~ 16,82 years | ~ 18.28 c. |
| 12 | ~ 11 sec | ~ 1.55 years | ~ 10.43 century | almost eternal |
| 13 | ~ 1.85 min | ~ 55,79 years | almost eternal | almost eternal |
| 14 | ~ 18.5 min | ~ 20.08 century | almost eternal | almost eternal |
| 15 | ~ 3.09 hours | almost eternal | almost eternal | almost eternal |
| … | | | | |
| 20 | ~ 35.33 years | almost eternal | almost eternal | almost eternal |

Time needed to create all possible password candidates when 100 billion passwords can be generated per second