howest
hogeschool

Object Oriented Architectures and Secure Development

Password Hashing

Matthias Blomme
Mattias De Wael
Frédéric Vlummens

Storing user passwords

- Various applications require users to authenticate.
- Typically done using a username and corresponding password.
- This information needs to be stored somewhere (e.g. database).
- Storing passwords as plain text in the database is an absolute NO.
- Reason:
  - Not secure
  - Anyone obtaining access to the database can consult the passwords.
  - This does not only compromise our own apps/websites, but also other apps or websites, since users often reuse the same passwords for different services.

Encrypting user passwords

- Never use a two-way encryption algorithm.
- If the two-way encryption key/password leaks, all passwords can be obtained!
- If a user looses her/his password, just generate a new password instead
- Therefore, we will be using one-way hashing

But…

- What about two people using the same password?
- And dictionary attacks or rainbow tables?
- Solution: salting
  - Random data
  - Added as additional input to the one-way hashing function
  - Two users having the same password will result in a different hash
  - Unfeasible to create rainbow tables for all possible salts

Some algorithms

- Argon 2
- SCrypt
- **BCrypt**

- More info in the cryptography classes

BCrypt

- Example of a password hashing algorithm
- Can easily be incorporated into your Java applications
- Add to build.gradle:

```
dependencies {
    testCompile group: 'junit', name: 'junit', version: '4.12'
    // https://mvnrepository.com/artifact/org.mindrot/jbcrypt
    compile group: 'org.mindrot', name: 'jbcrypt', version: '0.4'
}
```

- Note: other libraries exist (e.g. Jasypt, more info on the website)

BCrypt – usage

- Encrypting:

```
String hash = BCrypt.hashpw("hello-world", BCrypt.gensalt());
```

$2a$10$H.cNyX6Wa.EXfTtO6kgdNuCtqTngpyn.JUnDrOzMZKPqBhH.TLl9O

- Comparing with user input:

```
String password = in.nextLine(); // get from user

if (BCrypt.checkpw(password, hash)) {
    // welcome
}
```


howest hogeschool