Securing User Credentials

Wojciech Lesniak
AUTHOR

@voit3k

Spring Security

Browser

Resource

Data

User Registration

Register

Firstname

Enter firstname

Lastname

Enter lastname

Username

Enter username

Email address

Enter email

Password

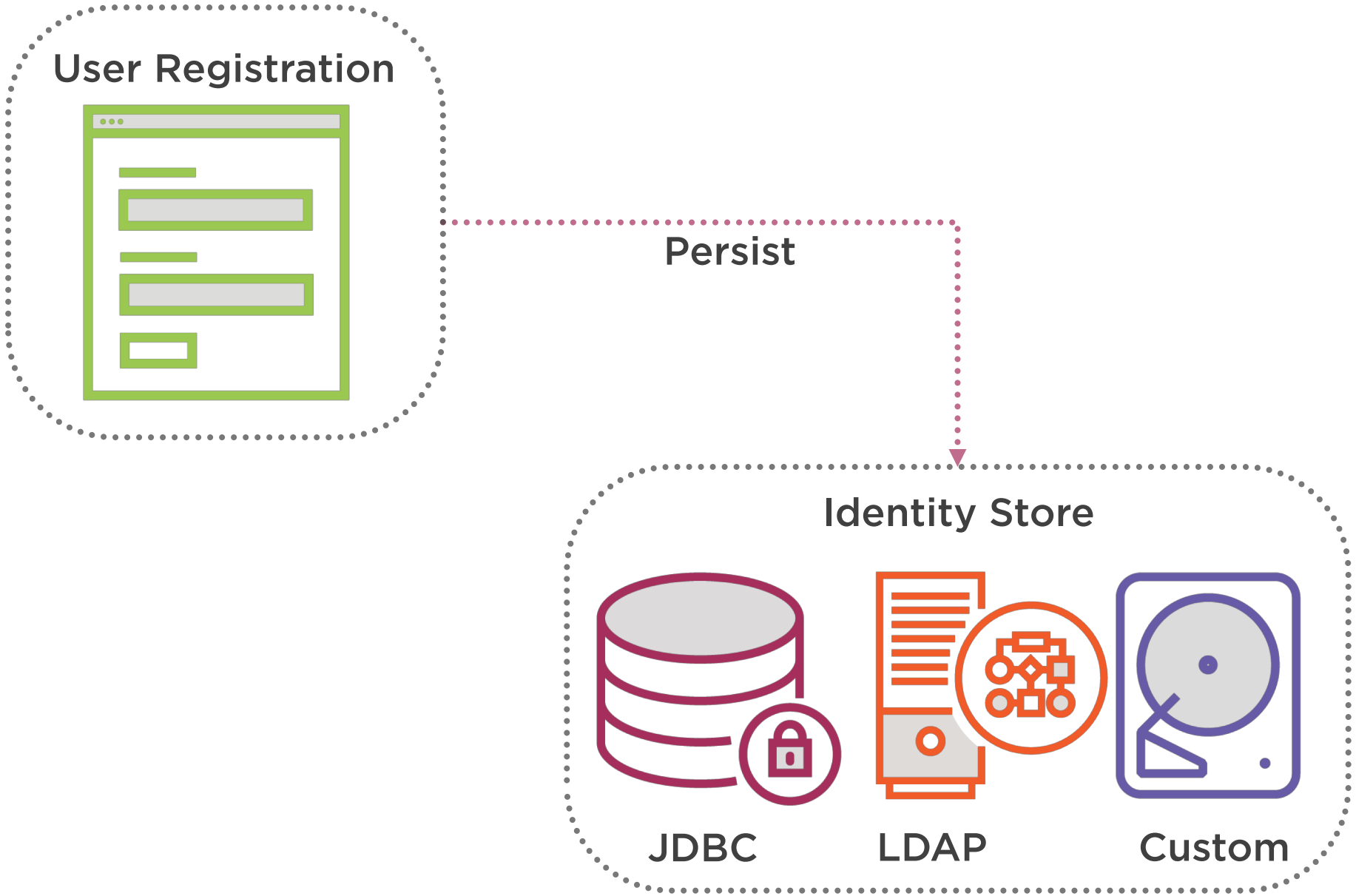
Password

Password

Password

Sign Up

Already have an account? Sign in.


User Registration
Persist
Identity Store
JDBC
LDAP
Custom

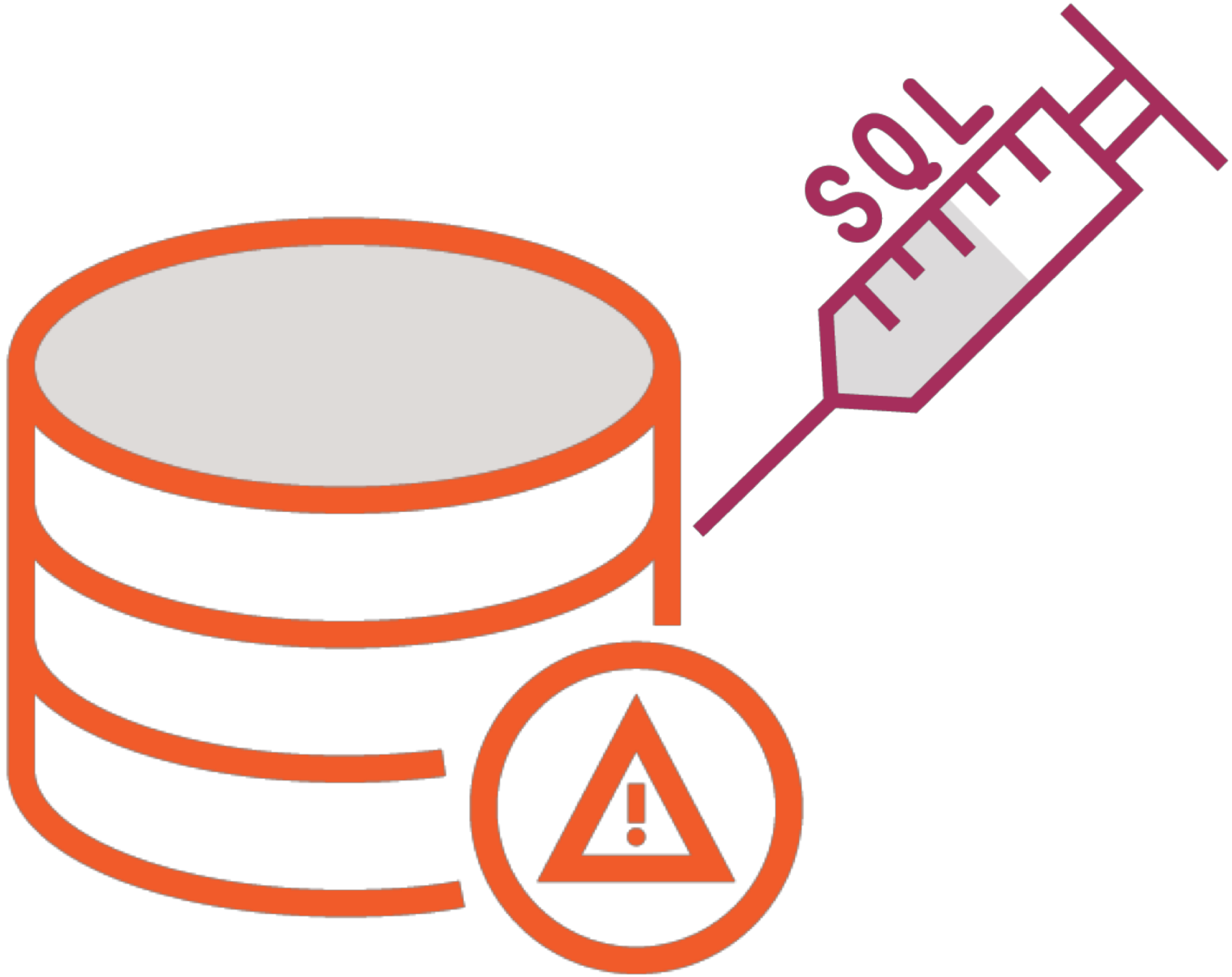
You Will Learn

Validate mandatory fields populated, and email and username are unique

Enforce password strength policies as per OWASP recommendations

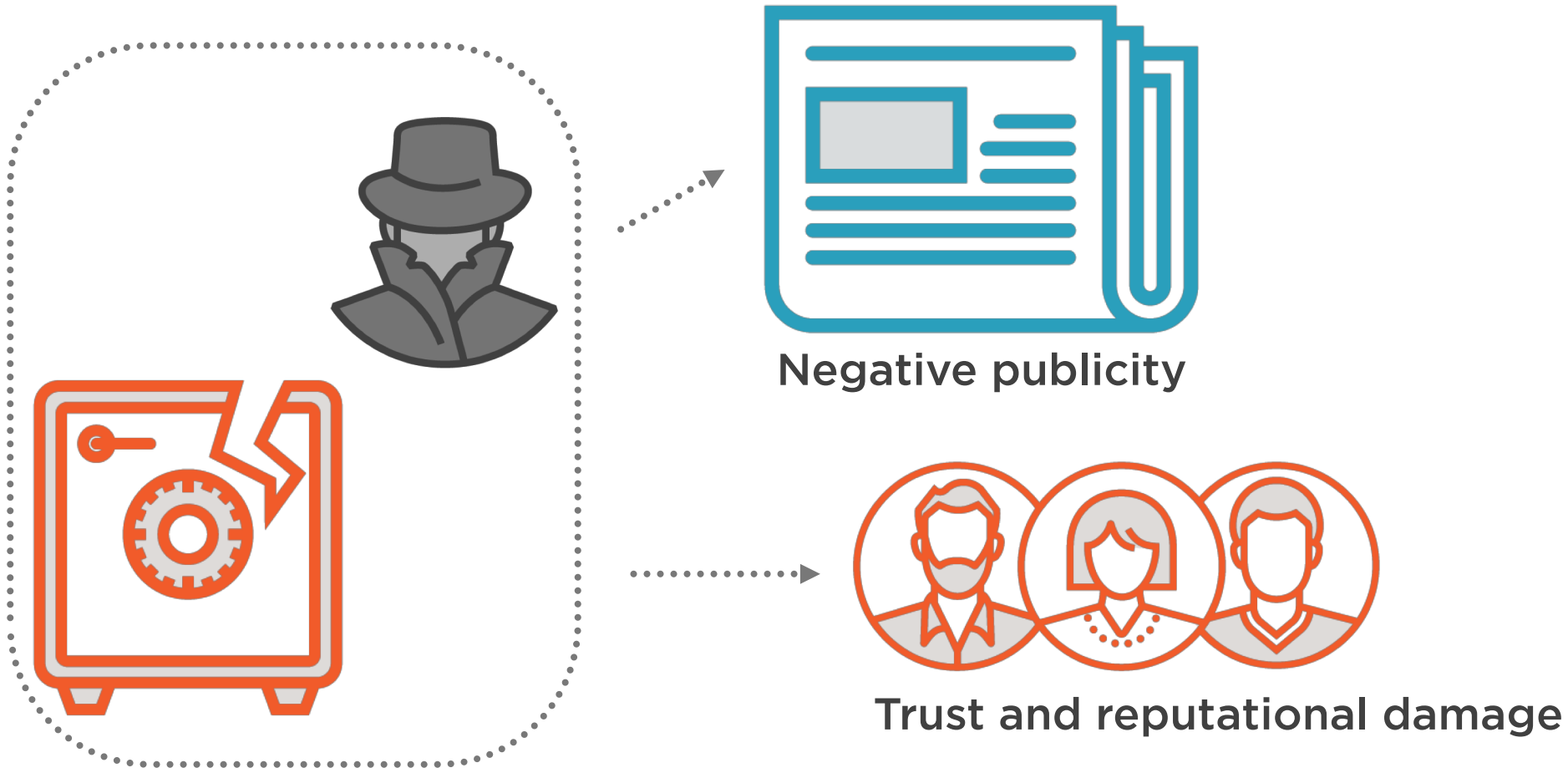
Secure credentials with Bcrypt encoder

Retrieve credentials from a JDBC or Custom Identity Store

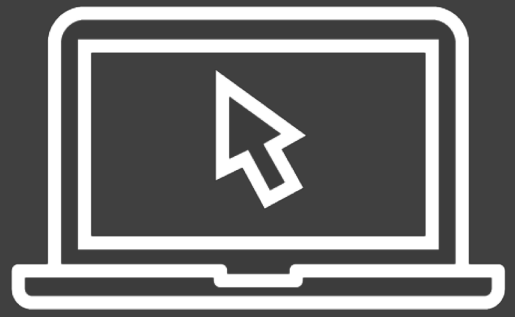

SQL


TOP
SECRET
Keystore password
Database credentials
API tokens

High Profile Hacking Incidents



Demo



Injections – stealing user credentials

Registration Validation

Mandatory fields are populated

Enforce password strength policies as per OWASP recommendations

Secure credentials with Bcrypt encoder

Retrieve credentials from a JDBC or Custom Identity Store

Secrets

Database credentials

TLS certificates, keystore location and passwords

Tokens for APIs your application uses

Secret Sprawl

Source control

Property file

Env variables

Configuration management

Source code

Centralized Secret Management

Provide fine grained access to the secrets

You also want your secrets to be encrypted both a rest and in transit to the client

The ability to rotate secrets

Audit who and when was the secret accessed by

vault.conf

```
backend "inmem" {
}
listener "tcp" {
  address = "0.0.0.0:8200"
  tls_disable = 1
}
disable_mlock = true
```

Summary

You now know how to:

- Configure Spring Security to work with your identity store
- Create a user registration page and perform validation
- Enforce password strength policies recommended by OWASP
- Using Spring password encoders to securely encode and decode passwords
- Using Bcrypt effectively
- How to use Spring Cloud Vault to secure your application secrets

Summary

Key takeaways:

- Enforce password strength as recommended by OWASP
- Always store your passwords as a one way hash and not in plaintext
- If using Bcrypt ensure the work factor is appropriate
- Centralize your secrets, ensure they are encrypted, audited