07

Mãos à obra: Utilizando a EntityManager

Os métodos presentes na classe `UsuarioDaoImpl` estão realizando a concatenação a *query* diretamente com os parâmetros passados no formulário e com isso a aplicação está suscetível as injeções de códigos SQL. Vamos utilizar a especificação da JPA e a `EntityManager` para realizar essa comunicação com o banco.

Remova a linha onde instanciamos a classe `ConnectionFactory` e vamos fazer a injeção do `EntityManager` :

```
@PersistenceContext
private EntityManager manager;
```

Feito isso, vamos no método **salva** e vamos utilizar o método `persist` para realizar a persistência dos usuários.

```
public void salva(Usuario usuario) {
    manager.persist(usuario);
}
```

Na sequência, vamos alterar o método **procuraUsuario** desvinculando a *query* dos parâmetros passados do formulário:

```
public Usuario procuraUsuario(Usuario usuario) {
     TypedQuery<Usuario> query = manager.createQuery("select u from Usuario u where u.email=:ema
    query.setParameter("email", usuario.getEmail());
    query.setParameter("senha", usuario.getSenha());
    Usuario usuarioRetornado = query.getResultList().stream().findFirst().orElse(null);
    return usuarioRetornado;
}
```

Uma vez que essas mudanças foram realizadas, volte para o Kali Linux para que possamos fazer o teste e confirmar que de fato agora os parâmetros não estão mais sendo injetados. Antes de executarmos os testes, vamos precisar remover os testes previamente realizados do SQLMAP com o comando:

```
rm -r /root/.sqlmap/output/
```

Na sequência, execute o mesmo teste do SQLMAP que fizémos na etapa anterior:

```
sqlmap -u "[URL]" --dump -T usuario -D owasp --data="alex@gmail.com&senha=123"
```

Qual foi o resultado? Foi possível obter as entradas presentes na tabela usuario do banco owasp?