Ateliers Sécurité des applications Web

Pré-requis :

- Node
- IDE (IntelliJIDEA, Eclipse, STS, VSCode)
- Librairie lombok : https://projectlombok.org/downloads/lombok.jar
- Docker
- Git

Table des matières

Ateliers 1 : Introduction

1.1 Démonstration des attaques

Démarrage de l'application insecure fournie :

```
./mvnw spring-boot:run
```

1.1.1 XSS

Essayer les URLS suivantes

XSS Reflected :
http://localhost:8080/search?q=toto
Puis
http://localhost:8080/search?q=toto<script>alert('toto')</script>

DOM Reflected
http://localhost:8080/search?q=toto#%3Cimg%20src=%22a%22%20widh=%220%22%20onerror=%22alert('toto')%22/%3E

1.1.2 CSRF

Connecter vous à l'application avec user/user sur http://localhost:8080/authenticated
Visualiser les logs de l'application un message doit y être inscrit

Démarrer l'application node.js attacker

```
npm start
```

Accéder à la page d'accueil

Revoir les logs de l'application java

1.1.3 SSRF

Accéder à http://localhost:8080/ssrf

Faites effectuer au backend une requête non prévue (http://localhost/Admin par exemple)

1.2 Cryptographie

BCrypt

Ateliers 2 : OWASP

2.1 L’application WebGoat

https://github.com/WebGoat/WebGoat/

Ateliers 3 : Tests de la sécurité

3.1 Reconnaissance

3.1.1 Recherche de sous-domaine

Sur un domaine que vous connaissez, essayer :

Une recherche Google de type
`site :<domaine> -inurl:www log in`

Utilisation dnscan, si vous avez Python installé
`git clone https://github.com/rbsec/dnscan.git`

`cd dnscan`

`pip install -r requirements.txt`

`./dnscan.py -d <votre-domaine>`

Récupérér le code javascript fourni brute-domain.js , le comprendre puis le tester
`npm start`

3.1.2 Identifier les technos :

Saisissez un domaine connu sur Netcraft :
https://sitereport.netcraft.com/

Utiliser nmap pour la découverte de ports non standard

3.1.3 Identifier les APIs

java -jar OWASP.jar