

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.....	2
1.1 Démonstration des attaques.....	2
1.1.1 XSS.....	2
1.1.2 CSRF.....	2
1.1.3 SSRF.....	2
1.2 Cryptographie.....	2
Ateliers 2 : OWASP.....	3
2.1 L'application WebGoat.....	3
Ateliers 3 : Construire une application secure.....	4
Ateliers 4 : Tests de la sécurité.....	5
4.1 Reconnaissance.....	5
4.1.1 Recherche de sous-domaine.....	5
4.1.2 Identifier les technos :.....	5
4.1.3 Identifier les APIs.....	5
4.2 L'outil de pénétration ZAP.....	5

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%20width=%220%22%20onerror=%22alert\('toto'\)%22/%3E](http://localhost:8080/search?q=toto#%3Cimg%20src=%22a%22%20width=%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 : Construire une application secure

3.1 Code Review et analyse de code

dependency-check

CheckMarx

FindSecurityBug

Ateliers 4 : Tests de la sécurité

4.1 Reconnaissance

4.1.1 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>
```

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

4.1.2 Identifier les technos :

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

Utiliser nmap pour la découverte de ports non standard

4.1.3 Identifier les APIs

```
java -jar OWASP.jar
```

4.2 L'outil de pénétration ZAP

Définition des utilisateurs et des rôles

Passive scanning

Active scanning

Automatisation et intégration dans pipeline