M1 WEB - 2019

1) Qu'est-ce qu'une faille XSS ? Que peut-on faire avec? Comment s'en protéger? (4 points)

XSS (ou Cross-Site Scripting) est un type d'injection où un attaquant envoie un code malveillant (généralement du javascript) à un site. L'attaque se produit quand l'utilisateur affiche le site avec le script.
Grâce à cette attaque, il est possible d'avoir accès aux mêmes informations que le site. L'attaquant peut potentiellement lire les secrets (par exemple, une session stockée dans un cookie non protégé), manipulé le site (défacement, construction de formulaires ou exécution de requêtes pour effectuer une attaque CSRF, …), redirigé l'utilisateur vers d'autres sites, …
Plusieurs solutions existent pour s'en protéger:

- *Effectuer un traitement classique les données avant de les afficher ou de les manipuler (validation, échappement, sanitisation) dans la page*
- *Utiliser des librairies pour détecter et avertir en cas de tentative d'attaque*
- *Utiliser des outils d'analyse statique de code*
- *Tester correctement l'application*
- *Ne pas utiliser des fonctions dangereuses type eval*

Des solutions existent pour en réduire l'impact:

- *Stocker les secrets de manière à ce qu'ils soient inaccessibles depuis le javascript (Set-Cookie avec le flag httpOnly) et ne pas conserver les secrets plus que nécessaire*
- *Utiliser correctement les frameworks type Angular ou React*
- *Utiliser des en-têtes HTTP spécifique comme X-XSS-Protection ou CSP*
- *Certains navigateurs ont des mécanismes de détection (comme Chrome)*

2) Qu'est-ce que l'OWASP? Qu'est-ce que le TOP 10 OWASP? (1 point)

OWASP (pour Open Web Application Security Project) est une organisation qui a pour but d'aider à l'amélioration de la sécurité des applications Web.
Elle publie régulièrement (tous les 3/4 ans) une liste des risques les plus courants pour les applications Web.

3) Quel problème trouve-t'on dans ce code? Comment le corriger? (2 points)

```
String path = getInputPath();
if (path.startsWith("/safe_dir/")) {
  File f = new File(path);
  f.delete()
}
```

On voit que la variable "path" est utilisée sans aucun traitement. Un attaquant pourrait fournir une valeur spécialement conçue pour accéder à n'importe quel fichier (par exemple: /safe_dir/../../../etc/passwd). Il s'agit d'une attaque de type "Path Traversal" ou "Directory Traversal".

La solution la plus classique est de valider/échapper/sanitizer la variable. Une autre solution est de transformer sous une forme le chemin final (realpath en PHP ou JS) et de vérifier que le dossier (basename) est bien celui attendu.
Enfin, plus dans une approche "Defense in Depth", on peut empêcher l'utilisateur associé au serveur/application d'avoir accès (en lecture, écriture, …) aux fichiers qu'il n'est pas censé pouvoir lire; voire isolé le plus possible de composants avec chroot, Docker...

4) Quelle est la différence entre validation, sanitization et échappement? (2 points)

- *Validation: on vérifie que la donnée est bien dans une forme attendue par le système, aussi bien syntaxiquement (exemple: un age a pour format [0-9]{3}) que sémantiquement (une personne ne peut pas être âgée de plus de 150 ans)*
- *Sanitization: on nettoie/filtre les données pour supprimer les caractères potentiellement dangereux.*
- *Echappement: on transforme les données pour que celles-ci ne soient pas interprétés/traitées par un autre système*

5) La fonction suivante est une fonction qui fait de la validation? De la sanitization? De l'échappement? Qu'en pensez-vous? Proposez un ou des améliorations? (3 points)

```
public String removeScriptTags(String htmlCode, String mask) {
  return htmlCode.replaceAll("script", mask);
}
```

Ici, on peut considérer qu'on peut faire soit de la sanitization (si le masque est vide, on supprime le code) que de l'échappement (on pourrait remplacer par une forme neutre comme "s c r i p t")
Il est très facile de passer outre ces protections. Si par exemple, le masque est vide et que l'on soumette "scrscriptipt", on obtiendra "script". De plus, il n'est pas dit que le traitement soit insensible à la case. Si on considère que c'est la seule protection contre XSS, il est possible d'exécuter un script, Enfin, si le contenu contient le mot "script" qui sera supprimé (faux positif). Un exemple d'amélioration est le fait d'utiliser des librairies reconnues pour ce type de traitement.

6) Citez deux en-têtes HTTP servant à améliorer la sécurité et donnez leur définition. (2 points)

- *Cf. cours*

7) Citez deux ~~menaces (threats)~~ risques issus du TOP 10 OWASP. En donner une courte description (2 points)

- *Cf. cours*

8) Lorsque je teste une API avec la commande

```
curl -H 'Content-Type: application/json' http://site/search?q='
```

Et j'obtiens le résultat suivant:

```
< HTTP/1.1 500 Internal Server Error
< Content-Type: application/json;charset=utf-8
```

```
< Server: Jetty(9.4.6.v20170531)
{"response": "org.springframework.jdbc.BadSqlGrammarException:
StatementCallback; bad SQL grammar [select * from products
where lower(title) like '%'%']; at
SQLExceptionSubclassTranslator.java:91"}
```

Décrire le ou les problème(s), leur(s) causes probables et les bénéfices pour un attaquant (4 points)

- *Erreur 500: cela indique clairement à un attaquant que le serveur s'est retrouvé dans un état inattendu. Un attaquant peut continuer à tester différentes valeurs jusqu'à avoir un résultat. Une cause probable est le non traitement d'une erreur qui se propage et est traitée par le serveur applicatif*
- *Le type de serveur est dans l'en-tête. Avec un peu de chance, une CVE existe avec un exploit associé. Ici, le serveur n'a pas été correctement configuré pour ne pas envoyer cette information.*
- *Le message d'erreur montre clairement la cause de l'erreur et le code associé, Java en l'occurrence. Il s'agit d'une faille de type SQL Injection. A partir de là, un attaquant peut extraire certaines informations du site. Encore faut-il que le compte de la base de données n'est pas des droits limités.*

~~9) Citez un élément du modèle de sécurité de Java (1 point)~~

10) Citez un principe du développement sécurisé par Design (1 point)

Cf. Cours