MSSI

DJEBIEN Tarik
RAKOTOBE Eric
STIENNE Rudy

INTRODUCTION

- Web 2.0 : apparition de nouveaux types d’ attaques qui exploitent les faiblesses de conception des usages du Web.

- Présentation de 3 vulnérabilités :
  - Attaque par injection SQL
  - Faille XSS
  - Faille XSRF

PLAN

1) ***Injections SQL***
   a) Définition et méthodologie d'attaque.
   b) Risques et menaces.
   c) Réduction des risques : bonnes pratiques et plans d'actions.

2) Failles XSS
   a) Définition
   b) Risques et réduction des risques

3) Failles XSRF
   a) Définition
   b) Risques et réduction des risques

INJECTIONS SQL - Définition

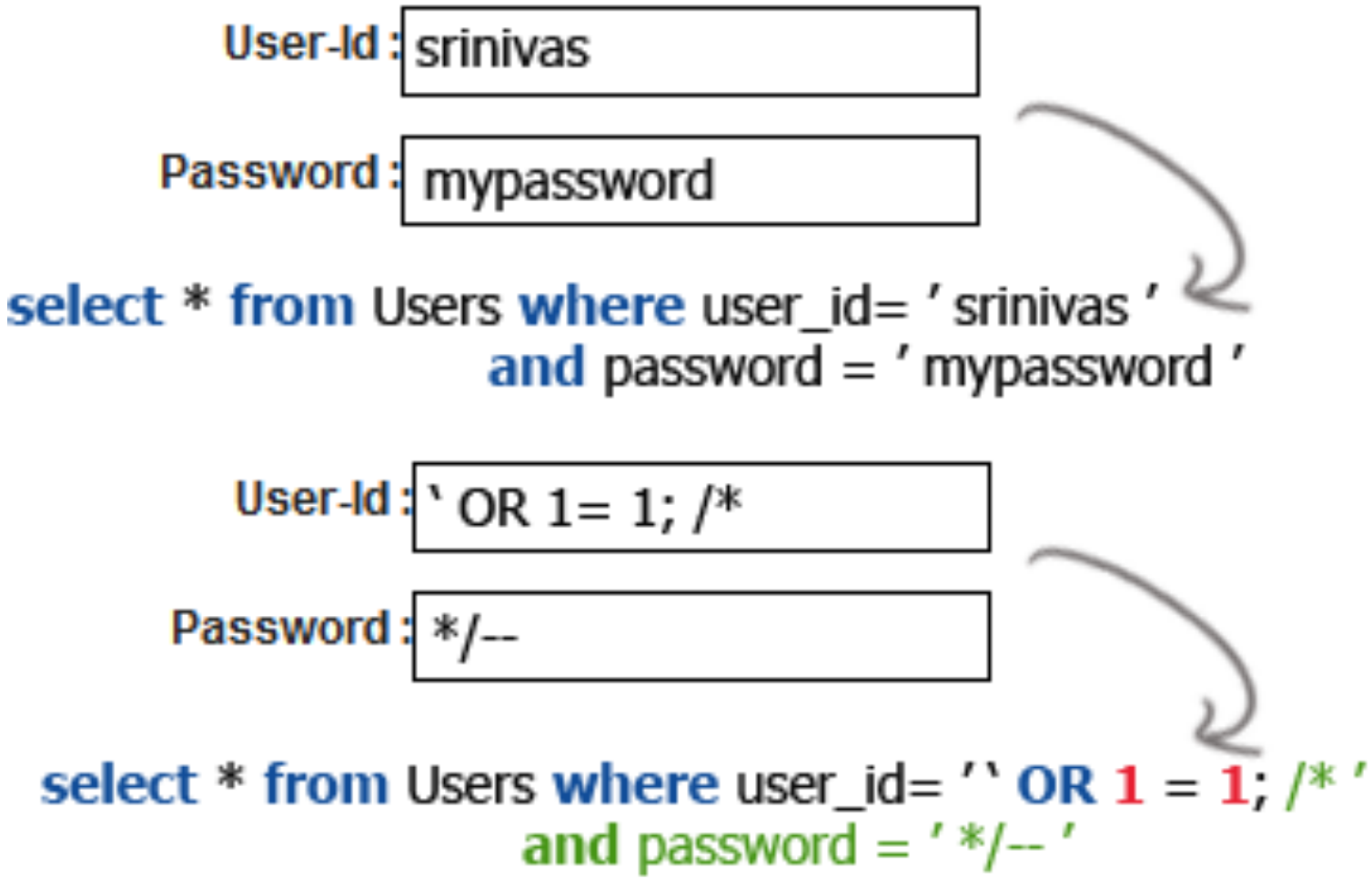
Définition :

L'attaque par injection SQL vise les sites Web qui proposent des transactions mal construites dont les résultats sont emmagasinés dans une base de données relationnelle.

Mise en oeuvre :

- Identifier la technologie sur laquelle repose l'application web.
- Établir la liste de toutes les saisies utilisateur possibles.
- Trouver la saisie utilisateur vulnérable.

INJECTIONS SQL - Exemple



INJECTIONS SQL - Risques

Popularité :	**8**
Simplicité (Vraisemblance) :	**8**
Impacts :	**9**
Confidentialité	*9*
Disponibilité	*9*
Intégrité	*9*
Évaluation du risque :	**9**

Menaces possibles :

- *Contournement de l'authentification.*
- *Récupération de codes de cartes de crédits.*
- *Prise de contrôle complète de la base de données sur un serveur distant.*

INJECTIONS SQL
- Réduction des risques

Principes à respecter :

- Contrôler et contraindre la saisie client/serveur.
- Faire une analyse lexicale et syntaxique pour échapper les caractères prohibés appartenant au ordres SQL.
- Utiliser au mieux les requêtes préparées.
- Désactiver les informations relatives au erreurs.
- "Principle of least privilege".

Sensibilisation aux enjeux de la sécurité via des bonnes pratiques de programmation pour les équipes développeurs.

INJECTIONS SQL
- Réduction des risques

Plan d'actions :

Langages de programmation	Actions clients <form>	Actions Serveurs DAOs
JEE	JSP avec JSTL pour les données saisies.	- JDBC PreparedStatement - JPA NamedQuery.
PHP		- PDO - mysql_real_escape_string() - addcslashes().
.NET	ASP : -RegularExpressionValidator -RangeValidator.	- Regex (untrusted client, library code) - SqlParameterCollection (PL/SQL dyn) - inputSQL.Replace()

FAILLES XSS

Définition :

- faille de sécurité des sites webs
- injection de code malveillant en langage de script dans un site web vulnerable
- execute du script cote client

Exemple : Rentrer dans un formulaire ou une url du javascript : **<script > alert (Hack) </script >**

FAILLES XSS

Principe :

1) Bob consulte la page faillible et repère une faille XSS.

2) Il héberge son script d'exploitation sur son serveur, ainsi qu'un script dit "de capture" (que nous pourrons appeler grabber), et intègre son script d'exploitation dans la page faillible.

FAILLES XSS

3) Il envoie le lien de la page modifiée à Alice.

4) Celle ci le consulte, croyant avoir affaire à la page originale.

5) Elle se fait alors piéger, envoyant des données au grabber de Bob qui enregistre les données sensibles.

FAILLES XSS

6) Bob récupère alors ces données et les utilise sur le site faillible (cookies, mot de passe, etc).

FAILLES XSS

Solutions en java :

- nettoyer les données reçues via le formulaire, en remplaçant toutes balises candidates à une éventuelle attaque XSS :

```
public static String sanitize(String string) {
  return string
     .replaceAll("(?i)<script.*?>.*?</script.*?>", "")   // case 1
     .replaceAll("(?i)<.*?javascript:.*?>.*?</.*?>", "") // case 2
     .replaceAll("(?i)<.*?\\s+on.*?>.*?</.*?>", "");     // case 3
}
```

FAILLES XSS

Solutions en java :

- utiliser l'attribut par défaut de JSTL **escapeXml = "true"** pour l'affichage d'une chaîne de caractère, au lieu d'utiliser les balises **<%= … %>** non imbriqué

```
<c:out value="${foo}" escapeXml="true" />
```

FAILLES XSS

Solutions en Php:

- **htmlentities ()** qui filtre tout les caractères équivalents au codage html ou javascript

ou/et

strip_tags qui supprime toutes les balises

ou/et

htmlspecialchars () qui convertissent les caractères spéciaux en entités HTML

FAILLES XSS

Solutions en Php:

- Désactiver **register_globals ,** pour éviter l’ enregistrement de toutes les variables (reçues via un formulaire par exemple) dans des variables globales systèmes

- Utiliser la prochaine bibliothèque PHP Anti-XSS de l'OWASP.

FAILLES XSS

Solutions en .Net:

- Utiliser la bibliothèque Microsoft Anti-XSS 1.5

- Vérifier que dans le fichier Machine.config du serveur, la validation de requête est activée :

```
<system.web>
  <pages buffer="true" validateRequest="true" />
</system.web>
```

FAILLES XSS

- Utiliser les fonctions prédéfinies en .Net, par exemple **HtmlEncode** ou créer vos propres fonctions de nettoyage

```
<script runat="server">

  void submitBtn_Click(object sender, EventArgs e)
  {
    // Encode the string input
    StringBuilder sb = new StringBuilder(
                           HttpUtility.HtmlEncode(htmlInputTxt.Text));
    // Selectively allow  <b> and <i>
    sb.Replace("&lt;b&gt;", "<b>");
    sb.Replace("&lt;/b&gt;", "");
    sb.Replace("&lt;i&gt;", "<i>");
    sb.Replace("&lt;/i&gt;", "");
    Response.Write(sb.ToString());
  }
</script>
```

FAILLES XSS

Les risques liés:

- Redirection (parfois de manière transparente) de l'utilisateur (souvent dans un but de hameçonnage)
- Vol d'informations, par exemple sessions et cookies.
- Actions sur le site faillible, à l'insu de la victime et sous son identité (envoi de messages, suppression de données, etc.)
- Rendre la lecture d'une page difficile (boucle infinie d'alertes par exemple).

FAILLES XSS

Réduction des risques:

- Vérifier les données saisies par les utilisateurs

- Télécharger des plugins pour analyser si votre application contient des failles XSS : https://addons.mozilla.org/fr/firefox/addon/xss-me/

FAILLES XSS

- Pour les cookies :

enregistrer l'IP du client pour lequel on forge le cookie, et refuser la connection et détruire le cookie si elle est demandée avec le même cookie mais que l'IP diffère de celle de l'enregistrement.
Une connection depuis un autre ordinateur directement par cookie et non par identifiants signifierais donc peut-être une tentative d'attaque.

FAILLES XSS

- se protéger des failles de type XSS à l'aide d'équipements réseaux dédiés tels que les **pare-feux applicatifs.**
Ces derniers permettent de **filtrer** l'ensemble des **flux HTTP** afin de détecter les requêtes suspectes.

FAILLES XSRF

Définition : Une faille XSRF (ou Cross Site Request Forgery) est une attaque qui permet de vous faire exécuter n'importe quelle action sur un site (à condition qu'il soit exploitable).

Utilisateur = **déclencheur**

Complice sans en être **conscient**.

Objectif : générer une requête par le navigateur de la victime

FAILLES XSRF

Fonctionnement par l'exemple :

Bob se connecte sur sa webmail et reste connecté et authentifié grâce aux systèmes de sessions pendant qu'il navigue sur d'autres pages.

Alice veut supprimer tous les messages de la webmail de Bob, elle a trouvé le lien qui permettait de le faire (un lien du type : http://www.webmail.fr/index.php?id=123456&delete=all)

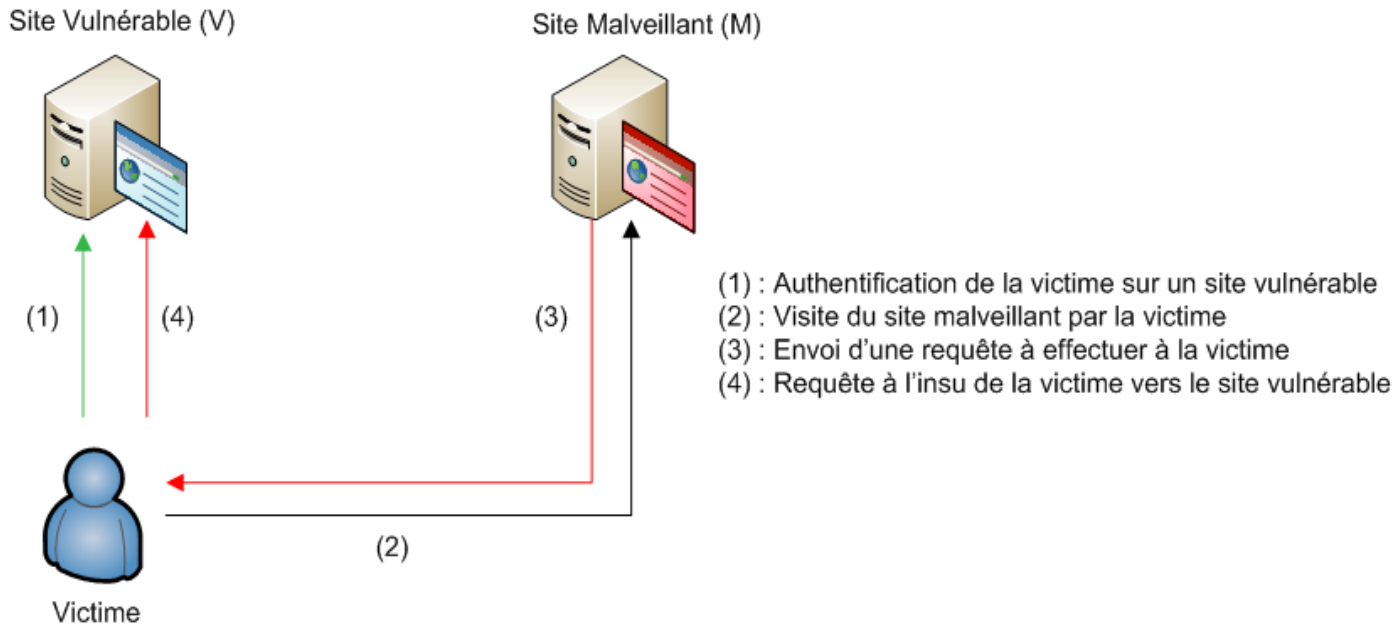
Elle n'a plus qu'à envoyer le lien par n'importe quel biais à Bob.

Si Bob sans se méfier clique sur ce lien, le navigateur étant authentifié sur sa webmail, il a les droits pour effectuer n'importe quelle action dessus.

C'est ainsi que tous les messages de sa webmail vont être supprimés.

FAILLES XSRF

Schéma du fonctionnement :



FAILLES XSRF

Solutions en Java :

- jeton de sécurité : un hash MD5 de l'identifiant de session est envoyé avec les requêtes

- générer une chaîne aléatoire : une chaine de caractères aléatoire est générée => stockée en variable de session => HttpServletRequest.

La jsp renvoit la chaine avec un champ caché dans le formulaire vers la servlet. Comparaison effectuée pour vérifier si la chaine envoyée par le formulaire est égal à celle qui sera stockée dans la variable de session.

FAILLES XSRF

Solutions en PHP :

- token : un jeton de sécurité qu'on stokera dans une session et qu'on enverra dans le formulaire par le biais d'un champ caché, ceci permet de vérifier que la personne qui tente d'exécuter la page est bien passée par le formulaire avant, où on lui a délivré le jeton.

- referer : vérifier que la page qui a conduit le visiteur à une autre page a bien été chargé par le clic sur un lien ou bouton sur cette première page. Ceci pour vérifier si une requête ajax n'a pas été exécuté sur le navigateur du client à son insu. $_SERVER['HTTP_REFERER'] est une variable disponible n'importe où dans le code et qui contient l'adresse, si elle existe, de la page qui a amené le visiteur sur le script en cours.

FAILLES XSRF

Solutions en PHP :

- Empêcher l'exécution du code Ajax : échapper les caractères HTML nécessaires, par exemple le < et les ' " qui sont indispensables pour le type d'attaque par exécution de code Ajax.
Et ceci en utilisant la fonction htmlspecialchars().

FAILLES XSRF

Solutions en .NET :

- Un champ caché dans le formulaire : le framework ASP.Net place un état de vue dans chaque page sous forme d'un champ caché de formulaire, le champ VIEWSTATE

- Un champ de formulaire de validation : champ de formulaire supplémentaire de validation d'évènement : _EVENTVALIDATION. Pour éviter l'attaque qui consiste à publier des messages auprès de gestionnaires d'évènements à l' écoute

FAILLES XSRF

Risques liés :

- opérations sur un site sans le consentement d'un utilisateur

- destruction de données ou manque d’intégrités des données

- usurpation d’identitée

- divulgation d'informations

FAILLES XSRF

RÉDUCTION DES RISQUES :

Pour les développeurs :

- Passer le plus de données possibles en POST (éviter le GET)

- Sur une page traitement demander une confirmation avant de valider l'action

- Réduire la durée de la validité de la connexion

- Exiger une double connexion pour obtenir l'accès à la partie d'administration

FAILLES XSRF

RÉDUCTION DES RISQUES :

Pour les développeurs :

- Demander avant une action critique de retaper le mot de passe

En conclusion pour le développeur il faut choisir une ou un ensemble de protections qui combinent la praticité et efficacité, c'est à dire le compromis entre sécurité, facilité d'utilisation et la fonctionnalité

FAILLES XSRF

RÉDUCTION DES RISQUES :

Pour les utilisateurs :

- Se déconnecter de son compte une fois l'utilisation du site terminée

- Ne pas sauvegarder les identifiants et mots de passe sur son navigateur, effacer les cookies régulièrement.

- Ne pas suivre les liens suspects

- Utiliser un 2e navigateur pour que le site malicieux ne puisse pas récupérer vos droits.

CONCLUSION

Nombreuses possibilités offertes aux développeurs pour protéger les sites webs contre des attaques frauduleuses.

Sensibiliser les développeurs sur la sécurité des applications.

Sensibiliser les utilisateurs de navigateurs web, les informer sur les quelques mesures à prendre pour éviter ces risques.