UNIVERSITE VIRTUELLE DU SENEGAL
foo nekk foofu la

Sécurisation et optimisation des applications mobiles et web

M. Birahim BABOU

Séquence 2 : Chapitre 2 : Failles de sécurités

Table des matières

Table des figures

Table des tableaux

1. Chapitre 2 : Failles de sécurités

1.1. Gestion des identités et manipulation de vos données de manière sécurisée

1.1.1. Protection des données

1.1.2. Sécurisation des communications avec le TLS

Pour assurer la confidentialité des données, l'intégrité des données et l'authentification, l'utilisation de protocoles TLS ou SSL demeure indispensable.

Une attaque de **l'homme du milieu**, qui intercepte des communications entre deux utilisateurs en faisant référence à ce serveur intermédiaire qui se fait passer pour votre interlocuteur (banque par exemple).

L'utilisation répandue du protocole TLS est le sujet de nombreuses études qui aboutissent régulièrement à la découverte de **nouvelles vulnérabilités**. Compte tenu des corrections et des améliorations apportées au fil des années, à la fois aux spécifications et aux implémentations, il est essentiel d'**utiliser les dernières versions des équipements et logiciels impliqués dans la sécurisation des communications.**

Toutefois, pour fixer les idées, TLS assure 3 fonctions majeures pour la sécurité des communications entre les applications :

- **la confidentialité ;**
- **l'intégrité ;**
- **et l'authentification**.

1.1.3. Gestion de l'authentification d'une application

Il s'agit ici de permettre à un utilisateur de prouver sa bonne foi.

Plusieurs méthodes existent :

1.1.3.1. La connaissance d'un mot de passe associé à son identifiant

C'est une méthode simple, mais qui présente l'inconvénient d'être facilement recopiée, sans que le propriétaire légitime puisse s'en rendre compte.

La parade consiste à forcer le changement régulier de ce dernier, de donner une durée de validité avant de devoir le ressaisir et à mettre en place des mécanismes permettant de compter le nombre de fois où ce dernier est en cours d'utilisation.

Le mot de passe doit être suffisamment complexe (lié au nombre de caractères) pour résister à une attaque intelligente comme par exemple une attaque « force brute » utilisant des dictionnaires, ou l'utilisation des informations connues de l'utilisateur (date de naissance, prénom des enfants, etc.).

Il n'est pas nécessaire que ce soit une chaîne de caractères complexes, impossible à mémoriser, une phrase assez longue en français, avec chiffres et accents, peut suffire.

1.1.3.2. Tout autre mécanisme plus complexe à dupliquer (token, clé RSA, carte à puce, détection biométrique)

Pour utiliser le code d'accès, il faut physiquement utiliser l'objet, et le propriétaire peut se rendre compte d'un vol.

1.1.3.3. Contrôle du contexte d'utilisation (un mot de passe où le jeton est spécifique de l'endroit où il est utilisé)

On ne peut pas utiliser celui-ci hors d'une zone géographique ou sur n'importe quel ordinateur.

1.2. Les vulnérabilités

1.2.1. Vulnérabilités Web et applicatives

La cartographie d'un site est le plan du site qui est constitué de plusieurs pages. L'URL et l'URI sont les emplacements de la ressource au niveau de l'espace d'hébergement du site. La ressource peut être locale ou externe.

1.2.1.1. Cross-Site Scripting (XSS)

L'Open Web Application Security Project (OWASP) considère la vulnérabilité à XSS comme une faille critique car elle est très répandue et facile à détecter. Les attaques s'appuient principalement sur les formulaires des applications Web. Les victimes sont les utilisateurs des applications Web vulnérables.

L'ANSSI signale que les scripts frauduleux peuvent endommager la base de registre de la victime, afficher des formulaires dont les saisies seront envoyées à l'attaquant, récupérer les cookies présents sur la machine de la victime, exécuter des commandes systèmes et construire des liens déguisés vers des sites malveillants.

L'attaque XSS par réflexion (reflected XSS) s'appuie sur le fait que l'application Web affiche ce que l'utilisateur vient de saisir dans un formulaire dans une page de résultat. Le navigateur de la victime exécute alors le code frauduleux généré dans la page de résultat. Tous les champs de formulaire sont donc une faille de sécurité potentielle que l'attaquant peut exploiter par XSS. L'attaquant crée un lien déguisé vers l'application Web dont un des paramètres contient du code JavaScript frauduleux. En utilisant ce lien, la victime fait exécuter par son navigateur le code JavaScript. Le Web 2.0 et ses systèmes de gestion de contenu ont popularisé cette attaque en permettant de publier des liens aisément et visibles sur tout le Web.

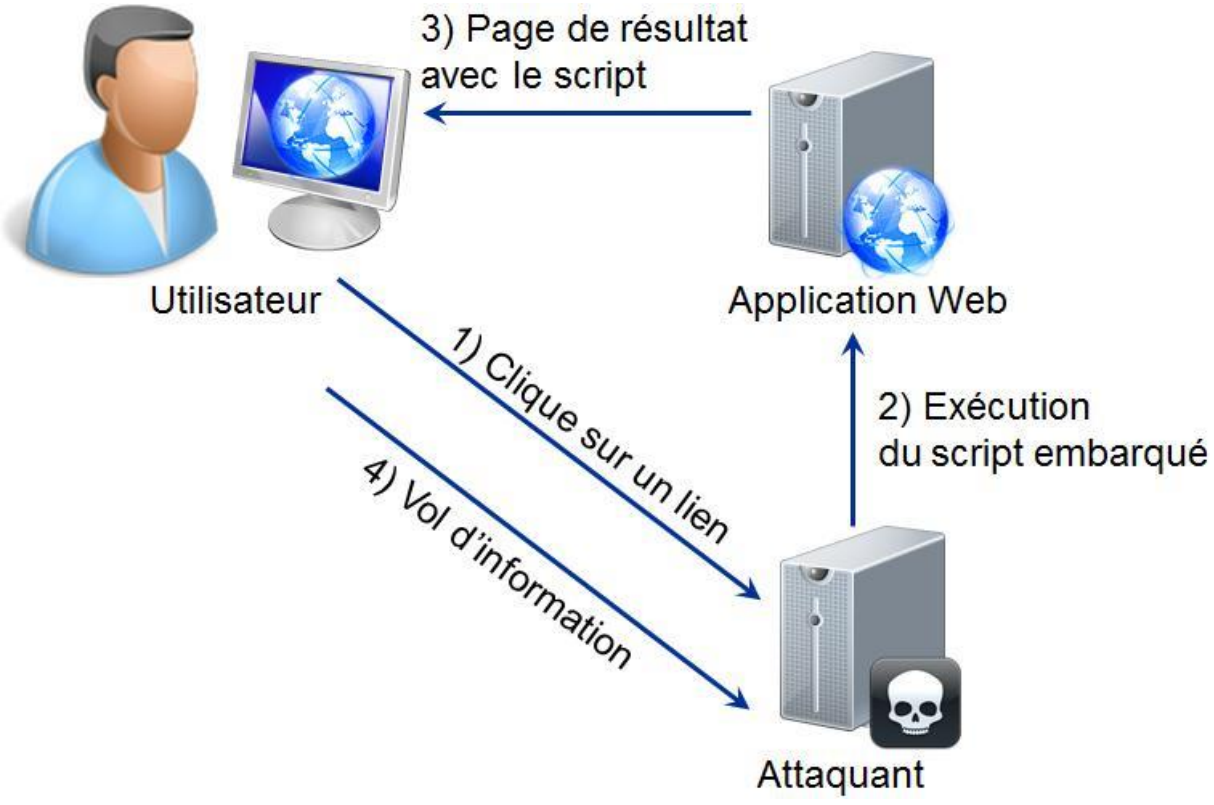

Figure 1: Principe d'une attaque XSS par réflexion

L'attaque XSS stockée (stored XSS) s'appuie sur le fait que l'attaquant réussisse à stocker dans la base de données du code frauduleux qui sera exécuté par la victime lorsqu'elle tentera d'afficher la donnée malveillante. Cette attaque est plus dangereuse que la première car le code fait partie intégrante des données de l'application Web et peut atteindre plusieurs victimes.

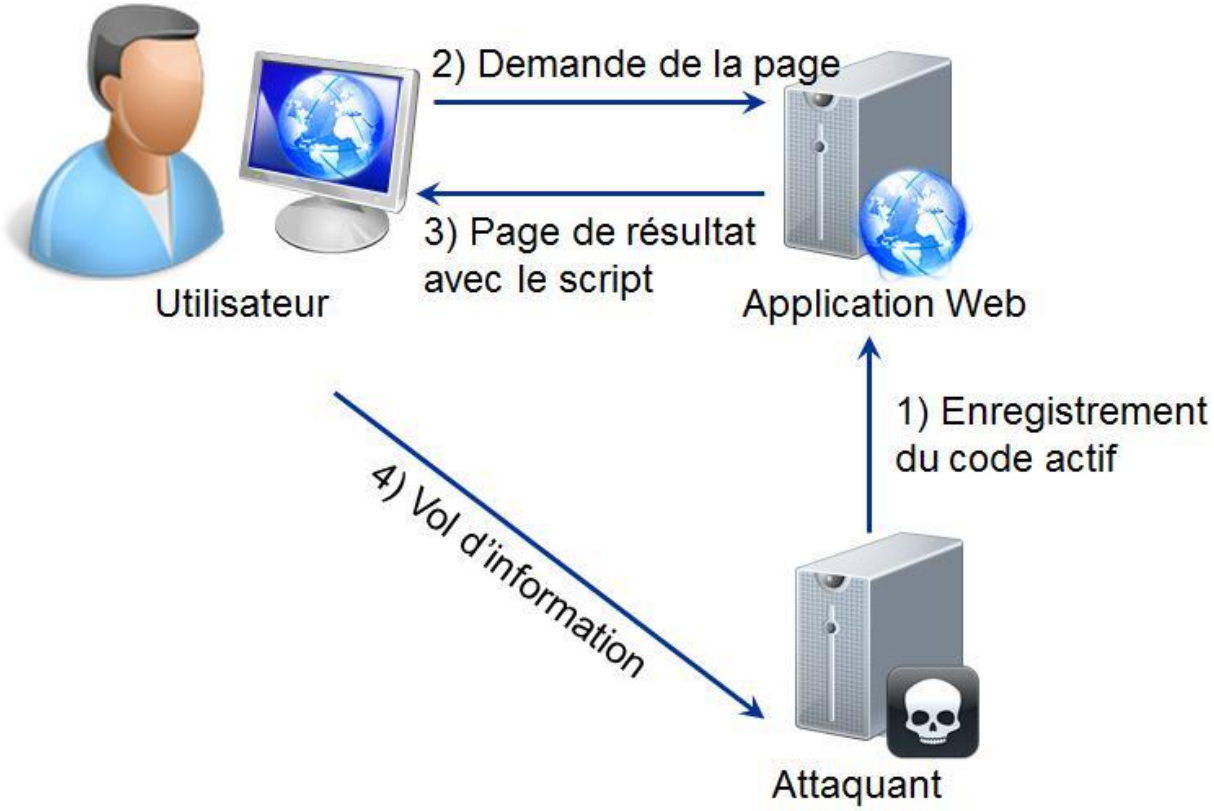

Figure 2: Principe d'une attaque XSS stockée

```
http://www.forum-vulnérable.com/recherche.php?parametre=<script>alert(&#8216;attaque XSS')</script>
```

Figure 3: Exemple de lien malveillant exploitant une faille XSS

1.2.1.2. Injections SQL

L'attaque par injection est évaluée par l'OWASP comme étant la plus risquée, car la faille est assez répandue, il est facile de l'exploiter et l'impact peut être très important. Cela va de la simple récupération de données à la prise totale de contrôle du serveur. La victime de l'attaque est un des composants techniques de l'application Web.

L'attaque par injection SQL consiste à injecter du code SQL qui sera interprété par le moteur de base de données. Le code malicieux le plus répandu est d'ajouter une instruction pour faire en sorte que la requête sous-jacente soit toujours positive. Cela permet par exemple d'usurper une identité pour se connecter à une application Web, de rendre l'application inutilisable ou de supprimer toutes les données de la table visée, voire de la base de données complète.

1.2.1.3. Les failles de PHP
1.2.1.4. Upload de fichier
1.2.1.5. Les symboles (< et &lt ;)

1.2.2. Vulnérabilités réseaux

Tout réseau privé qui communique en utilisant le protocole TCP/IP est un intranet La communication entre un intranet et un extranet est complètement indépendant du type de réseau utilisé.

Internet est le réseau des réseaux, c'est une interconnexion de plusieurs réseaux locaux. Il prend en compte la diversité de ces différents réseaux interconnectés à traver des protocoles et autres. Toutes ces machines interconnectées sont capables de communiquer en utilisant la pile protocolaire dite TCP/IP.

Les réseaux sans fil sont les plus vulnérables que les réseaux filaires.

Les 3 piliers de la sécurité des réseaux :

- Analyser, détecter et corriger ;
- Traiter les problèmes de vulnérabilité, la gestion des correctifs et audit
- Résoudre les problèmes le plus rapidement et efficace possible

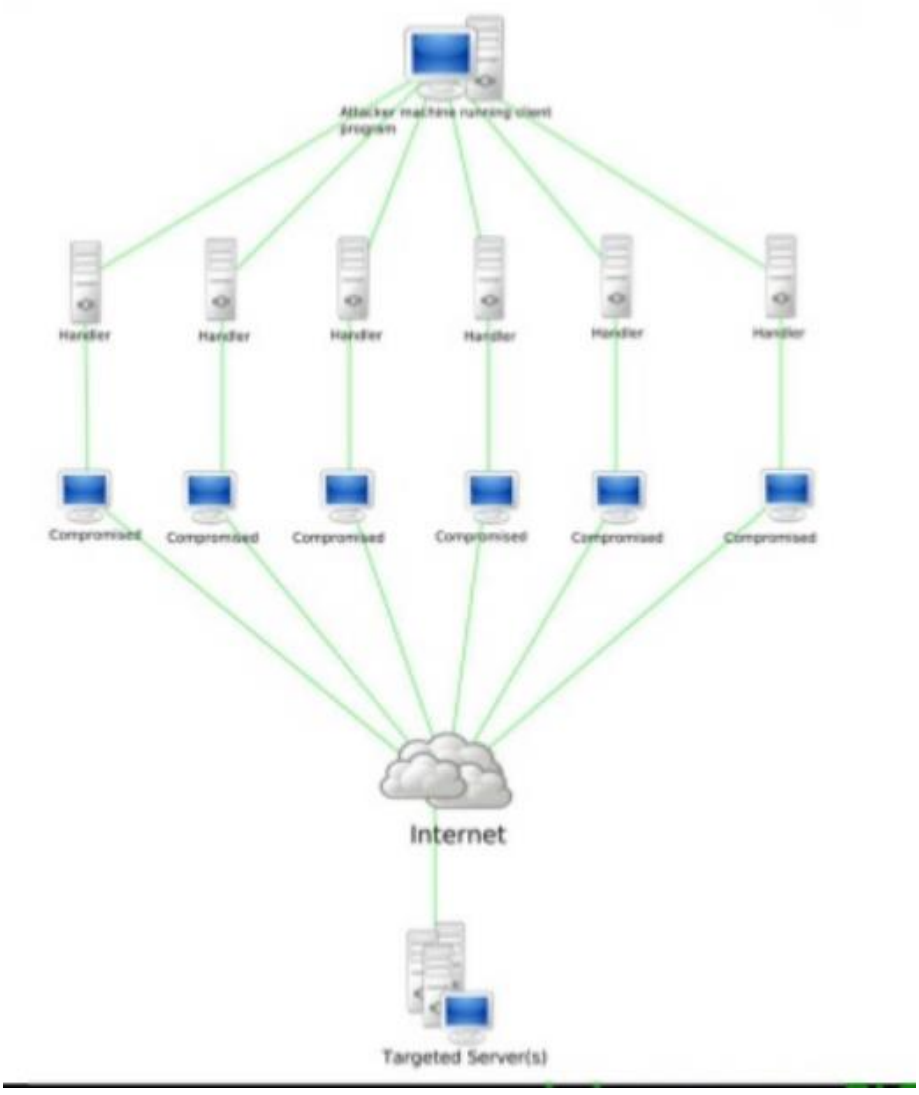

Figure 4: Réseau

1.3. Menaces et risques applicatifs

1.3.1. Types d'attaques

1.3.2. Risques de sécurités

Les 10 principaux risques pour la sécurité des applications Web d'après l'OWASP :

1.3.2.1. Injection

Des failles d'injection, telles que l'injection SQL, NoSQL, OS et LDAP, se produisent lorsque des données non fiables sont envoyées à un interpréteur dans le cadre d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent inciter l'interprète à exécuter des commandes involontaires ou à accéder aux données sans autorisation appropriée.

1.3.2.2. Authentification brisée

Les fonctions d'application liées à l'authentification et à la gestion des sessions sont souvent implémentées de manière incorrecte, permettant aux attaquants de compromettre les mots de passe, les clés ou les jetons de session, ou d'exploiter d'autres failles d'implémentation pour assumer l'identité des autres utilisateurs de manière temporaire ou permanente.

1.3.2.3. Exposition de données sensibles

De nombreuses applications Web et API ne protègent pas correctement les données sensibles, telles que les finances, les soins de santé, etc. Les attaquants peuvent voler ou modifier ces données faiblement protégées pour commettre des fraudes à la carte de crédit, au vol d'identité ou d'autres délits. Les données sensibles peuvent être compromises sans protection supplémentaire, comme le chiffrement au repos ou en transaction, et nécessitent des précautions particulières lors de l'échange avec le navigateur.

1.3.2.4. Entités externes XML (XXE)

De nombreux processeurs XML plus anciens ou mal configurés évaluent les références d'entités externes dans les documents XML. Les entités externes peuvent être utilisées pour divulguer des fichiers internes à l'aide du gestionnaire d'URI de fichiers, des partages de fichiers internes, de l'analyse des ports internes, de l'exécution de code à distance et des attaques par déni de service.

1.3.2.5. Contrôle d'accès cassé

Les restrictions sur ce que les utilisateurs authentifiés sont autorisés à faire ne sont souvent pas correctement appliquées. Les attaquants peuvent exploiter ces failles pour accéder à des fonctionnalités et / ou des données non autorisées, comme accéder aux comptes d'autres utilisateurs, afficher des fichiers sensibles, modifier les données d'autres utilisateurs, changer les droits d'accès, etc.

1.3.2.6. Mauvaise configuration de la sécurité

Une mauvaise configuration de la sécurité est le problème le plus courant. Ceci est généralement le résultat de configurations par défaut non sécurisées, de configurations incomplètes ou ad hoc, d'un stockage cloud ouvert, d'en-têtes HTTP mal configurés et de messages d'erreur détaillés contenant des informations sensibles. Non seulement tous les systèmes d'exploitation, cadres, bibliothèques et applications doivent être configurés en toute sécurité, mais ils doivent être corrigés / mis à niveau en temps opportun.

1.3.2.7. Cross-Site Scripting XSS

Des failles XSS se produisent chaque fois qu'une application inclut des données non fiables dans une nouvelle page Web sans validation appropriée ni échappement, ou met à jour une page Web existante avec des données fournies par l'utilisateur à l'aide d'une API de navigateur qui peut créer du HTML ou du JavaScript. XSS permet aux attaquants d'exécuter des scripts dans le navigateur de la victime qui peuvent pirater des sessions utilisateur, défigurer des sites Web ou rediriger l'utilisateur vers des sites malveillants.

1.3.2.8. Désérialisation non sécurisée

Une désérialisation non sécurisée conduit souvent à l'exécution de code à distance. Même si les failles de désérialisation n'entraînent pas l'exécution de code à distance, elles peuvent être utilisées pour effectuer des attaques, notamment des attaques de relecture, des attaques par injection et des attaques par escalade de privilèges.

1.3.2.9. Utilisation de composants avec des vulnérabilités connues

Les composants, tels que les bibliothèques, les frameworks et autres modules logiciels, s'exécutent avec les mêmes privilèges que l'application. Si un composant vulnérable est exploité, une telle attaque peut faciliter une perte de données sérieuse ou une prise de contrôle du serveur. Les applications et les API utilisant des composants présentant des vulnérabilités connues peuvent saper les défenses des applications et permettre diverses attaques et impacts.

1.3.2.10. Journalisation et surveillance insuffisantes

La journalisation et la surveillance insuffisantes, associées à une intégration manquante ou inefficace avec la réponse aux incidents, permettent aux attaquants de continuer à attaquer les systèmes, de maintenir la persistance, de pivoter vers d'autres systèmes et de falsifier, d'extraire ou de détruire des données. La plupart des études de violation montrent que le temps de détection d'une violation est supérieur à 200 jours, généralement détecté par des parties externes plutôt que par des processus internes ou une surveillance.