Travaux Pratiques : Sécurité des applications

Rakotoarimalala Tsinjo Tony

Part I
Injection XSS

Introduction

L'objectif de ce TP est de comprendre comment les attaques XSS (Cross-Site Scripting) fonctionnent et comment elles peuvent être exploitées. Vous apprendrez également à vous défendre contre ces attaques en implémentant des mesures de sécurité appropriées.

Objectifs

- Identifier et exploiter des vulnérabilités XSS dans une application web vulnérable.
- Voler des cookies de session, rediriger les utilisateurs ou exécuter d'autres actions malveillantes via des scripts injectés.
- Implémenter des mesures de défense telles que l'échappement des données utilisateur et l'utilisation de Content Security Policy (CSP).
- Analyser les logs pour détecter les tentatives d'attaques XSS.

Étape 1 : Mise en place de l'application

- Développez une petite application web de ventes d'articles avec des commentaires vulnérable à XSS (authentification, détails articles et

commentaires).

- Assurez-vous que l'application permet aux utilisateurs de soumettre des commentaires

Étape 2 : Exploitation des vulnérabilités XSS pour afficher publicité

- Injectez des scripts malveillants dans les champs de saisie de l'application. Le script maleveillant permettra d'injecter des pubs venant d'un autre site.
- Observez l'effet de l'injection, comme l'apparition d'une publicité, le vol de cookies ou la redirection vers un site malveillant.

Étape 3 : Exploitation des vulnérabilités XSS pour voler des cookies publicité

- Injectez des scripts malveillants dans les champs de saisie de l'application. Le script maleveillant permettra de vol de cookies.
- Observez l'effet de l'injection

Étape 4 : Implémentation des mesures de défense

- Modifiez l'application pour échapper correctement les entrées utilisateur afin de prévenir l'exécution de scripts. Sécuriser les cookies en utilisant HttpOnly et Secure.
- Implémentez une politique de sécurité de contenu (CSP) pour restreindre les sources de scripts.
- Testez à nouveau l'application pour s'assurer que les attaques XSS sont bloquées.

Étape 5 : Analyse des logs

- Accédez aux logs de l'application pour rechercher des tentatives d'attaques XSS.
- Identifiez les signatures ou motifs dans les logs qui peuvent indiquer une tentative d'injection de script.

Conclusion

Ce TP vous a permis d'explorer les vulnérabilités XSS et de comprendre les méthodes pour protéger les applications contre ce type d'attaques. Les compétences acquises vous aideront à développer des applications plus sécurisées et à détecter les tentatives d'intrusion plus efficacement.

Part II
Injection SQL

Introduction

L'injection SQL est une vulnérabilité qui permet à un attaquant d'interagir directement avec la base de données d'une application en injectant des commandes SQL malveillantes dans des champs de saisie. Ce TP a pour objectif de vous familiariser avec cette technique d'attaque et de vous apprendre à protéger vos applications contre de telles vulnérabilités.

Objectifs

- Comprendre le fonctionnement de l'injection SQL.
- Identifier et exploiter des vulnérabilités SQL dans une application web.
- Étudier les conséquences potentielles d'une injection SQL réussie, telles que la divulgation de données sensibles ou la modification non autorisée des données.

- Implémenter des mesures de défense contre l'injection SQL, comme l'utilisation de requêtes préparées.

Étape 1 : Mise en place de l'environnement

- Installez une application web vulnérable à l'injection SQL, telle que DVWA (Damn Vulnerable Web Application) ou développez une petite application avec une base de données simple.
- Configurez la base de données avec des tables d'exemples contenant des données fictives (par exemple, utilisateurs, mots de passe, etc.).

Étape 2 : Exploitation de l'injection SQL

- Accédez à l'application et repérez les champs de saisie susceptibles d'être vulnérables (formulaires de connexion, formulaires de recherche, etc.).
- Injectez des commandes SQL malveillantes pour contourner l'authentification, comme :

```
' OR '1'='1
```

- Essayez d'extraire des informations de la base de données en modifiant les requêtes SQL.
- Analysez les résultats pour comprendre comment l'injection a pu interférer avec la requête SQL initiale.

Étape 3 : Étude des conséquences

- Explorez les différentes actions qu'un attaquant pourrait entreprendre après avoir exploité une vulnérabilité SQL, telles que l'accès non autorisé à des informations confidentielles ou la suppression de données.
- Simulez une attaque visant à obtenir la liste des utilisateurs et leurs mots de passe.

- Documentez les résultats obtenus pour chaque tentative d'injection.

Étape 4 : Implémentation des mesures de défense

- Modifiez le code de l'application pour utiliser des requêtes préparées (prepared statements) ou des procédures stockées afin d'empêcher les injections SQL.
- Utilisez des mécanismes d'échappement des entrées utilisateur pour sécuriser les requêtes SQL.
- Testez à nouveau l'application pour vérifier que les injections SQL sont désormais impossibles.

Conclusion

Ce TP vous a permis de comprendre en profondeur les risques liés à l'injection SQL et de découvrir les méthodes pour sécuriser vos applications contre cette menace. Les compétences acquises vous aideront à développer des applications plus robustes et à mieux protéger les données de vos utilisateurs.

Part III
Brute force

1. Sur l'une des applications précédentes faite une attaque brute force pour trouver le mot de passe d'un utilisateur (mettez des exemples simples pour commencer) et accéder au compte de l'utilisateur (on suppose qu'on connaît le login mais pas le mot de passe): utiliser Request de python pour faire l'attaque. (vous pouvez aussi utiliser Itertools pour générer tous les mots de passe possibles)
2. Mettez les sécurités nécessaires pour ce genre d'attaque (authentification multi-facteur, limite de nombre de tentatives échouées, blocage de certains IP,...)