Insecure Code Management

Description

Le passage en production d'une application web omet parfois de supprimer certaines informations qui ne devraient pas être exposées (fichiers de configuration, commentaires de code, fichiers de backup, scripts de déploiement, fichiers ou répertoires cachés, etc…). Les vulnérabilités de type Insecure Code Management ou Gestion de code non sécurisée peuvent représenter un risque en matière de sécurité dans les applications web.

Voici une description des différents types de vulnérabilités associées à ce domaine :

- **Inclusion de fichiers non sécurisée (Insecure File Inclusion)** : cette vulnérabilité se produit lorsque le code de l'application inclut des fichiers externes sans vérifier correctement les chemins d'accès. Un attaquant peut exploiter cette faille pour inclure des fichiers malveillants et potentiellement exécuter du code arbitraire sur le serveur.
- **Gestion incorrecte des droits d'accès (Insecure Permissions)** : lorsque les permissions d'accès aux fichiers et répertoires ne sont pas configurées de manière appropriée, des personnes non autorisées peuvent accéder à des ressources sensibles ou modifier des fichiers critiques de l'application.
- **Gestion faible des erreurs (Weak Error Handling)** : une mauvaise gestion des erreurs peut révéler des informations sensibles sur l'application et faciliter des attaques par ingénierie sociale. Les messages d'erreur peuvent donner des indications aux attaquants sur les vulnérabilités potentielles.
- **Gestion non sécurisée des mots de passe (Insecure Password Management)** : lorsque les mots de passe sont stockés de manière non sécurisée, comme en texte clair ou avec des algorithmes de hachage faibles, ils deviennent vulnérables à des attaques de récupération de mots de passe.
- **Gestion de sessions non sécurisées (Insecure Session Management)** : une gestion de session faible peut permettre à un attaquant de voler des cookies de session ou de manipuler les paramètres de session pour usurper l'identité d'un utilisateur authentifié.

Un utilisateur malveillant peut alors mettre à profit ces informations pour gagner des accès illégitimes sur l'application, le système qui l'héberge, ou même sur d'autres machines du système d'information.

Prérequis d'exploitation

Pour exploiter cette famille de vulnérabilité, il est nécessaire d’avoir une application qui expose des informations sensibles comme des identifiants de connexion.

Connaissances nécessaires

- Connaissances des divers risques liés à de mauvaises pratiques de développement.

Outils nécessaires

- Outils de modification et/ou d'interception de reqûetes (Burp, Curl) ;
- Outils de fuzzing (dirbuster, gobuster, feroxbuster, Nikto) ;
- En fonction de la nature des ressources obtenues, utiliser un outil adapté à son exploitation (MySLQ, git, binwalk, exiftool, etc.).

Flux d'exécution

Explorer

Naviguer sur l'application ou utiliser un fuzzer afin d'identifier des ressources susceptibles de contenir des informations sensibles.

Expérimenter

Rechercher des informations sensibles dans chaque ressource identifiée, et dans le cas d'une découverte de nouvelles ressources, réitérer la phase d'exploration.

Exploiter

Conséquences potentielles

L'exploitation réussie de ce type de vulnérabilité peut conduire à :

- L’accès à des ressources sensibles ;
- L’accès à des fonctionnalités sensibles ;
- La compromission du serveur hébergeant l'application.

Contre-mesures

Les contre-mesures suivantes peuvent être mises en œuvre :

- Adopter une politique de développement sécurisé (SSDLC) en prenant en compte la sécurité à chaque phase du cycle de vie du développement ;
- Réaliser des tests de sécurité (tests d'intrusion) avant le passage en production et idéalement, à chaque mise à jour de l'application ;
- Sensibiliser les équipes de développement aux meilleures pratiques en termes de développement et de sécurité.

Comment cela fonctionne

Les scénarios suivants peuvent être joués via l’exploitation de ce type de vulnérabilité :

- Un attaquant parvient à identifier une extension qui n'est pas bloquée par le site victime. En énumérant les endpoints grâce à un fuzzer, il découvre un fichier de backup contenant des identifiants de connexion à la base de données de l'application. Grâce à cela il parvient alors à se connecter à la base de données et à en extraire tout son contenu.
- Un attaquant parvient à identifier un répertoire “listable” sur le serveur Apache qui contient des fichiers sensibles. Il peut ainsi récupérer des informations sensibles et les utiliser pour essayer de se connecter sur un espace privé de l'application ou directement sur le serveur.

Exemple 1

Considérons le fichier “.htaccess” suivant, qui serait accessible en lecture :

```
<FilesMatch ".(bak|config|sql|fla|psd|ini|log|sh|inc|swp)$">
    Order allow,deny
    Deny from all
    Satisfy All
</FilesMatch>
```

Cette configuration permet de bloquer toutes les requêtes vers des fichiers portant les extensions listées dans la directive “FilesMatch”. L'absence, dans cette liste, de l'extension “\~” permet d'accéder à des fichiers tels que “backup.bak\~” ou “error.log\~”.

Or certains éditeurs de texte, en ouvrant un fichier, créent une copie locale avec l'extension “\~”. Si un administrateur ouvre un fichier sensible avec “vim”, un attaquant pourra alors accéder à son contenu.

Exemple 2

L'attaquant identifie un répertoire listable sur le serveur Apache qui contient des fichiers sensibles. Cela signifie que le répertoire est configuré de manière à permettre à Apache d'afficher une liste des fichiers et des dossiers qu'il contient lorsque l'accès à un fichier spécifique est restreint. L'attaquant peut trouver le répertoire en explorant les différents endpoints de l'application ou en utilisant des outils de fuzzing.

Cette attaque survient lorsque le serveur Apache est mal configuré pour autoriser l'indexation des répertoires, ce qui permet à quiconque d'accéder et de lister le contenu complet d'un répertoire, y compris les fichiers sensibles.

Correction de la vulnérabilité :

Pour corriger cette vulnérabilité, il est nécessaire de désactiver l'indexation des répertoires dans la configuration d'Apache. Cela peut être réalisé en modifiant le fichier de configuration Apache (généralement httpd.conf ou apache2.conf) et en désactivant l'option “Indexes” pour les répertoires concernés.

Exemple de correction dans la configuration Apache en quelques étapes :

1 - Ouvrir le fichier de configuration Apache avec un éditeur de texte :

```
sudo nano /etc/apache2/apache2.conf
```

2 - Recherchez les directives qui définissent les options pour les répertoires concernés et supprimez ou commentez l'option “Indexes” :

```
<Directory /var/www/html/repo_sensible>

    Options -Indexes
    AllowOverride None
    Require all granted
</Directory>
```

3 - Sauvegardez et fermez le fichier de configuration.

4 - Redémarrez Apache pour appliquer les modifications :

```
sudo service apache2 restart
```

Avec cette configuration, Apache n'autorisera plus l'indexation des répertoires spécifiés et empêchera l'accès et la liste des fichiers sensibles qui y sont stockés.

References

URL :

- https://assets.ctfassets.net/r3ffzvy7le94/7sNLETiclVpHZePxDMLxKz/b5807e82fd7eacf56eb62a66604cc9fa/IS___Fixing_Broken_Authentication.pdf
- https://owasp.org/www-pdf-archive/OWASP_Top_10_And_Root_Causes_Cincy_Feb_26_08_Final.pdf
- https://owasp.org/www-pdf-archive/1_OWASP-geneva-Spring-09-GIORIA.pdf

Retour fiches vulnérabilités

- Cyber fiches vulnérabilités