TP Insecure Deserialization

Outils nécessaires pour la réalisation de ce TP :

- wamp/xamp

Exercice 1 (10 points) :

Pour cet exercice, vous allez devoir modifier un cookie afin de réaliser une élévation de privilège. Ensuite, vous profiterez de vos privilèges pour analyser un fichier afin de réaliser une injection SQL !

Tout d'abord, cloner ce repo dans le dossier "www" de wamp.
Placez-vous dans le dossier exercice-1 (Celui de php).

(Vous pouvez créer un virtualhost avec wamp pour atteindre le projet plus simplement, n'hésitez pas à demander si vous ne savez pas comment en créer un)

allez sur phpmyadmin, et importez le dump SQL présent dans le dossier dump. Une base de données nommée "**tp_insecure_deserialization**" a dû être créée.

dans le projet, vous devez configurer vos informations de connexion à la BDD, pour cela, modifiez le fichier php -> /config/config.php.

Rendez-vous sur ce projet en local. Vous devez normalement tomber sur un formulaire des plus basique.

Vous êtes maintenant prêt à commencer !

Question 1 (2 points) :

- Connectez-vous (Peu importe les identifiants, cela va juste créer un cookie). Vous devriez voir un cookie sur votre navigateur nommé session ! Actuellement la valeur n'est pas très claire, car elle est encodée pour les urls.

- Récupérez la valeur du cookie et décodez-la. Vous devriez obtenir un objet php sérialisé. Analysez le et faites en sorte de devenir un admin en changeant la valeur du cookie avec celle que vous avez modifié !

Question 2 (TOTAL : 5 points) :

Maintenant que vous disposez de privilèges intéressants, parlons de la classe User. Pour cette application, la classe User ré-écris une méthode magique appelée **__wakeup(),** cette méthode est appelée automatiquement quand votre objet a fini d'être dé-sérialiser. Elle va vérifier si vous êtes admin est utilisé la fonction **eval()** sur le champ **$adminAutoCommand** de la classe User**,** La fonction **eval()** a pour but d'exécuter du php que vous lui passerez en chaine de caractère. Les administrateurs de l'application étant feignant alors ils se servent de cette technique pour lancer leur script préféré à chaque fois qu'ils se connectent. Vous allez profiter de cette feignantise !

- Modifiez le cookie de session en changeant la valeur de **adminAutoCommand** pour que la valeur soit une String de php (Attention pensez bien à changer la taille du champ).

A. (2 points) Commencer par afficher un message, n'hésitez pas à aller plus loin.

B. (3 points) Faites en sorte d'afficher le contenu du fichier **"./controller/BasicController.php".** (Pensez bien que la méthode **eval()** exécute n'importe quel script php !)

Une fois le contenu obtenu vous pouvez utiliser ce site pour formater le code et y voir plus clair.

N'hésitez pas à demander des indices ou de l'aide si cela ne vous semble pas clair!

Question 3 (3 points) :

Après avoir analysé le code du fichier BasicController.php:
A. Trouver comment afficher le formulaire réservé aux admins, grâce au contenu du fichier BasicController.php.
B. Une fois que vous aurez réussi à afficher le formulaire, re-analysez le code du BasicController, quelle attaque sévère pensez-vous pouvoir réaliser ? (Indice à la question suivante, mais jouez le jeu !)

C. Réaliser une injection SQL pour faire des dégâts, par exemple supprimer la base de données (L'important est de réaliser une injection SQL, peu importe l'opération effectuée).

Exercice 2

Prérequis :

Vous rendrez votre compte rendu dans un document **Word** ou **PDF** nommé **ex2 -nom-prenom**

Outils nécessaires :

- Docker
- Java
- Un editeur de code (IntelliJ, NetBeans, sublim text, atom …)

Le but de cet exercice va être de réaliser une attaque de désérialisation insécurisée par laquelle vous allez exécuter du code à distance sur un serveur fictif.

Pour cela, nous allons simuler un pirate qui a détecté une faille dans le processus de sérialisation / désérialisation et qui va s'en servir pour exécuter du code à distance sur le serveur victime.

Dans ce TP, le pirate et le serveur seront dans un container docker. Le pirate (c'est-à-dire vous) va sérialiser un objet dans le container et le persister. Dans un second temps, et toujours dans le même container, nous allons simuler le serveur victime qui va désérialiser l'objet persister et exécuter des commandes malveillantes.

Mise en place du TP :

Prenez connaissance des fichiers ***JavaSerial.java*** *et* ***JavaDeserial.java*** présents dans les dossiers ***exercice-2/tmp/partie1*** **et** ***exercice-2/tmp/partie2*** à la racine du repo que vous avez cloné afin d'en comprendre leurs fonctionnements globaux.

Partie 1 :

Procédure :

1. Vous rendre dans le dossier ***exercice-2/tmp/partie1/*** en ligne de commande
2. Lancez la commande suivante :
   a. Sous windows (**PowerShell**) : **docker run --rm -it -v ${pwd}:/tmp:rw java /bin/bash**
   b. Sous Linux : **docker run -it -v "${PWD}":/tmp:rw java /bin/bash**
3. Une fois dans le container Docker, rendez-vous dans le dossier ***tmp/partie1*** puis lancer la commande suivante : **javac JavaSerial.java && java JavaSerial**
4. Lancez la commande suivante : **javac JavaDeserial.java && java JavaDeserial**

Question 1 (2 points)

Qu'observez-vous dans la sortie du container Docker ? Pourquoi ? Prenez une capture d'écran de la sortie que vous obtenez.

Partie 2 :

Dans la partie 1, nous avons pris la place d'un pirate ayant exécuté du code à distance sur un serveur fictif. La commande exécutée n'était pas vraiment malveillante, c'était simplement un exemple. Nous allons désormais simuler une réelle attaque ayant un impact bien plus conséquent.

Cette fois-ci, sur le serveur, est présent un fichier ***password.txt*****.** Nous allons afficher le contenu de ce fichier avant de le supprimer et tout cela avec du code exécuté à distance lors du processus de désérialisation.

<u>Question 1</u> (3 points) :

Rappel des commandes :

Il vous est demandé ici d'afficher le contenu du fichier ***password.txt*** avant de le supprimer, tout cela en exécutant du code à distance via un objet sérialisé.

Rendez-vous dans le fichier ***exercice-2*/tmp/partie2/JavaSerial.java.** Lisez le code et modifiez-le afin de pouvoir afficher le contenu du fichier ***password.txt*** dans un premier temps et dans un second temps supprimer ce fichier.

Prenez une capture d'écran du code que vous avez modifiez.

<u>Question 2</u> (3 points) :

Procédure :

1. Vous rendre dans le dossier ***exercice-2/tmp/partie2/*** en ligne de commande
2. Lancez la commande suivante :
   c. Sous windows (**<u>PowerShell</u>**) : **docker run --rm -it -v ${pwd}:/tmp:rw java /bin/bash**
   d. Sous Linux : **docker run -it -v "${PWD}":/tmp:rw java /bin/bash**
3. Une fois dans le container Docker, rendez-vous dans le dossier ***tmp/partie2*** puis lancer la commande suivante : **javac JavaSerial.java && java JavaSerial**
4. Lancez la commande suivante : **javac JavaDeserial.java && java JavaDeserial**

Qu'observez-vous dans la sortie du container Docker ? Pourquoi ? Prenez une capture d'écran de ce que la commande affiche.

<u>Question 3</u> (2 points) :

« Amusez-vous » à lancer quelques commandes à distance et vérifiez qu'elles fonctionnent. Vous donnerez les commandes que vous avez testé dans le compte rendu.

5. Quels problèmes pouvez-vous en déduire ?
6. Essayez d'exécuter d'autres commandes malveillantes sur le client en changeant la commande exécutée dans la classe Java correspondante.