École Polytechnique de Montréal

Département de génie informatique et génie logiciel

INF8085 - Cybersécurité

Travail Pratique 2

Hiver 2026

Table des matières

1. Directives

Tous les travaux devront être remis avant 23h55 le jour de la remise sur le site Moodle du cours. À moins que cela ne soit explicitement demandé dans le sujet, vous ne devez remettre qu'un fichier PDF nommé selon le format TPX-matricule1-matricule2.pdf. Vous pouvez inclure des annexes dans votre rapport si vous jugez que cela améliore la lisibilité (code source, . . .)

- Voir la date de remise du rapport de ce laboratoire dans le plan du cours.
- Le travail devra être fait par équipe de deux. Toute exception (travail individuel, équipe de trois) devra être approuvée au préalable par la·le professeur·e.
- L'orthographe et la forme seront prises en compte pour chaque question.
- Indiquez toutes vos sources d'information, qu'elles soient humaines ou documentaires.

NOTE : POUR TOUTES LES QUESTIONS, VOUS DEVEZ MONTRER COMMENT VOUS AVEZ OBTENU LES RÉPONSES, INCLUANT DANS VOTRE RAPPORT LES CAPTURES D'ÉCRAN MONTRANT LES COMMANDES UTILISÉES ET LEUR SORTIE.

2. Scénario

Votre équipe a été mandatée pour réaliser un audit de sécurité d'une application web utilisée en interne par l'entreprise de logiciels-services Cumulocode. L'application étant très simple, la direction du service informatique ne s'attend pas à ce que vous y trouviez de failles significatives.

Votre objectif : devenir `root` sur le serveur situé à l'adresse `10.22.0.11`, et expliquer à l'entreprise comment corriger les vulnérabilités que vous allez exploiter pour y parvenir. Il se peut que vous trouviez une chaîne d'attaque différente de celle suggérée dans les instructions qui suivent ; des points bonus seront accordés en conséquence.

Les différentes parties du TP se suivent de manière logique, mais des raccourcis sont disponibles pour permettre de réaliser chaque partie de manière indépendante.

3. Reconnaissance [/3]

Lancez la machine virtuelle `TP2` située dans `/home/VM/INF4420a/H2024/TP2/` et connectez-vous en SSH à la machine Kali Linux[1] en utilisant la commande `ssh root@localhost -p 2222` et le mot de passe `password`. Cette machine est située sur le même réseau que le serveur de Cumulocode.

Utilisez nmap[2] pour scanner le serveur situé en **`10.22.0.11`**. Identifiez les services et le système d'exploitation de la machine. Vous pouvez ignorer le port 2222.

Plus tard dans le TP, il se peut que d'autres services apparaissent sur cette machine. Revenez alors à cette question et indiquez les nouveaux services que vous observez.

4. Accès initial [/7]

Ouvrez un navigateur web et naviguez sur le site `http://localhost:8080/`. Il s'agit de l'appli- cation web de l'entreprise Cumulocode, qui permet de contrôler le statut des différents services hébergés par l'entreprise. Cette première partie consiste à obtenir un accès `admin` sur le site.

1. [/1] Montrez que la page de connexion est vulnérable aux injections SQL. [3]
2. [/2] En utilisant une injection SQL, connectez-vous sur le site. Avec vos mots, expliquez comment fonctionne votre attaque.
3. [/1] Montrez que la page de soumission de tickets est vulnérable aux injections XSS. [4]
4. [/2] L'administratrice du système consulte régulièrement les tickets de support. Utilisez une injection XSS pour récupérer les cookies de son navigateur web. [5, 6]
5. [/1] En utilisant le cookie récupéré, connectez-vous au site web en tant que **`admin`**.

5. Analyse en boîte blanche [/5]

L'objectif de cette partie est de comprendre le fonctionnement du site web, d'identifier des vulnérabilités supplémentaires, et de proposer des correctifs.

Raccourci : Cette partie suppose que vous avez obtenu un accès **`admin`** sur le site web dans la partie précédente. Si ce n'est pas le cas, utilisez le mot de passe `iL0veTrains#78` pour vous connecter en tant que `admin`.

Depuis l'application web, cliquez sur le bouton rouge pour allumer le service FTP. Vérifier qu'il est bien accessible en utilisant **nmap**, et mettez à jour la liste des services établie dans la partie 3.

1. [/1] Connectez-vous au service FTP et récupérez le code source du site. [7]
2. [/1] Expliquez précisément comment corriger la vulnérabilité qui permet cette connexion.
3. [/1] Localisez dans le code PHP l'injection SQL exploitée dans la partie 4 et corrigez la.
4. [/2] En vous basant sur le code source PHP et Javascript de l'application web, expliquez précisément le fonctionnement du mécanisme qui permet de basculer l'état des services. [8, 9]

6. Compromission d'un compte utilisateur [/4]

L'objectif de cette partie est de compromettre le compte de l'utilisatrice `carol` et de récupérer le code source d'un programme développé par Cumulocode.

1. [/2] En utilisant une injection SQL, récupérez le hash du mot de passe de `carol`. [10]
2. [/1] Casser le hash pour obtenir le mot de passe de `carol`. Indice : le mot de passe est composé uniquement de 6 chiffres. [11]
3. [/1] Connectez- vous en ssh au compte de `carol`, et récupérez le fichier `controller.c`. [12]

7. Élévation de privilèges [/6]

L'objectif de cette partie est d'obtenir un shell en tant que `root` sur le serveur.

Raccourci : Cette partie suppose que vous avez obtenu le fichier `controller.c` dans la partie précédente. Si ce n'est pas le cas, vous trouverez le fichier sur Moodle.

1. [/2] Utilisez une injection de commande PHP dans le mécanisme qui permet de basculer l'état des services analysé dans la partie 5.4 pour obtenir un shell en tant que `www-data`. [13]
2. [/2] Que fait le programme `controller.c` ? Identifiez une vulnérabilité dans ce programme causée par un débordement de tampon et expliquez comment la corriger. [14]
3. [/2] Exploitez le dépassement de tampon dans le programme `controller` et devenez `root`. [15]

8. Accès physique = Game Over [3 points bonus]

Obtenez un shell `root` sur la machine virtuelle `TP2`.

NOTE : LA QUESTION BONUS EST CORRIGÉE EN MODE RÉUSSITE OU ÉCHEC. VOUS NE POUVEZ DONC PAS AVOIR DE POINTS PARTIELS POUR CETTE QUESTION. LA NOTE MAXIMALE POUR CE TRAVAIL EST DE 100%, LES POINTS NE SERONT PAS REDISTRIBUÉS À UNE ÉVALUATION SUBSÉQUENTE.

Références

[1] Kali Linux => https://www.kali.org/

[2] nmap => https://nmap.org/

[3] Injection SQL => https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/

[4] Injection XSS => https://owasp.org/www-community/attacks/xss/

[5] Récupération de cookies => https://medium.com/@laur.telliskivi/pentesting-basics-cookie-grabber-xss-8b672e4738b2

[6] requestbin => https://pipedream.com/requestbin

[7] vsftpd => https://linux.die.net/man/5/vsftpd.conf

[8] Tutoriel PHP => https://www.freecodecamp.org/news/the-php-handbook/

[9] Tutoriel Javascript => https://www.tutorialrepublic.com/javascript-tutorial/javascript-ajax.php

[10] sqlmap => https://www.sqlinjection.net/sqlmap/tutorial/

[11] hashcat => https://hashcat.net/hashcat/

[12] git => https://git-scm.com/

[13] Injection de commandes => https://www.stackhawk.com/blog/php-command-injection/

[14] Dépassement de tampon => https://medium.com/techloop/understanding-buffer-overflow-vulnerability-85ac22ec8cd3

[15] gdb => https://oxasploits.com/posts/simple-buffer-overflow-exploitation-walkthrough-gdb/