**Travaux Pratiques - Projet Sécurité & Débogage**

**Objectifs :**

Ce projet a pour but de vous familiariser avec la gestion de la sécurité sur Windows via PowerShell, la création et le débogage de scripts, ainsi que la gestion des erreurs. Vous utiliserez Visual Studio Code et Git pour structurer et versionner vos travaux.

**Prérequis :**

- Visual Studio Code installé

- Git installé

- PowerShell (version 5.1 minimum recommandée)

- Système Windows (Home ou Pro)

**Organisation du projet :**

1. Créer un dossier de travail sur votre machine.

2. Initialiser un dépôt Git local dans ce dossier.

3. Créer plusieurs scripts PowerShell correspondant aux exercices ci-dessous.

4. Documenter vos scripts avec des commentaires expliquant chaque étape.

5. Valider régulièrement vos modifications avec Git.

**1. Gestion des règles de pare-feu Windows :**

- Affichez la configuration actuelle des profils de pare-feu.

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

- Ajoutez une règle de pare-feu qui bloque le trafic sortant sur un port TCP précis (au choix).

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

- Vérifiez que la règle est bien appliquée.

Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.

- Supprimez la règle créée.



**Objectif : Comprendre comment gérer le pare-feu avec PowerShell et manipuler des règles.**

**2. Création et débogage d’un script PowerShell**

- Créez un script PowerShell prenant un paramètre en entrée (par exemple un nom).

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

- Affichez des informations liées à l’exécution du script, notamment les arguments reçus et le chemin du script.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

- Mettez en place un point d’arrêt dans le script.



- Exécutez le script et utilisez les outils de débogage PowerShell pour analyser le comportement.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

**Objectif : Apprendre à créer des scripts avec paramètres et utiliser les fonctions de débogage intégrées.**

**3. Gestion des erreurs dans un script**

- Écrivez un script qui tente d’accéder à un fichier qui n’existe pas.



- Implémentez un bloc try/catch/finally pour capturer et gérer l’erreur.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

- Affichez un message utilisateur clair lorsque l’erreur est détectée.

- Assurez-vous que le script continue ou se termine proprement.



**Objectif : Savoir gérer les exceptions et assurer la robustesse d’un script.**

**Travail à rendre :**

- Tous vos scripts doivent être déposés dans le dossier du projet.

- Chaque script doit comporter des commentaires expliquant son fonctionnement.

- Rédigez un fichier README.md qui décrit votre démarche, les choix techniques, et les difficultés rencontrées.