LAB 1 – Extension Avancée : Projet de Simulation Keylogger

Ce document propose une version étendue du TP Keylogger initial. L'objectif est de passer d'un simple script de capture de frappes à une simulation cybersécurité plus réaliste, composée de plusieurs éléments interconnectés. L'ensemble doit être exécuté strictement dans des machines virtuelles VirtualBox, à des fins pédagogiques uniquement.

1. Objectifs

- Simuler une machine victime exécutant un keylogger avancé.
- Simuler une machine attaquante recevant, stockant et analysant les logs exfiltrés.
- Développer un contrôleur léger permettant de gérer la victime à distance.
- Laisser une flexibilité totale : les étudiants choisissent les langages, frameworks et architectures.

2. Vue d'Ensemble du Système

Les étudiants doivent concevoir un système composé de trois éléments :
1. VM Victime : exécute le keylogger, capture les frappes, les encode et les envoie.
2. VM Attaquant : reçoit les logs, les organise et les prépare pour l'affichage.
3. Contrôleur : interface simple (CLI ou web) permettant de surveiller et contrôler la victime.

3. Exigences pour la Machine Victime

- Capturer les frappes clavier en temps réel.
- Normaliser les caractères et les encoder en JSON.
- Générer un identifiant unique (UUID) au démarrage.
- Implémenter au moins une méthode d'exfiltration :
  - HTTP POST
  - Socket TCP
- Implémenter une résilience (mécanisme de retry, tampon local optionnel en cas de perte réseau).

4. Exigences pour la Machine Attaquante

- Mettre en place un récepteur (serveur HTTP, serveur socket, etc.).
- Stocker les logs par victime, dans des dossiers structurés.
- Optionnel : analyser les logs (mots-clés, séquences répétitives, comportements suspects).

5. Exigences pour le Contrôleur

Le contrôleur doit permettre :
- Lister les victimes actives.
- Afficher les logs en temps réel (streaming ou rafraîchissement périodique).
- Envoyer des commandes à distance, telles que :
  - start_capture
  - stop_capture

- switch_mode (http/tcp)
- flush_logs

Le contrôleur peut être sous forme de menu CLI, interface web simple ou tableau de bord léger.

6. Configuration VirtualBox

- Utiliser au minimum deux machines virtuelles : une victime et un attaquant.
- Configurer le réseau en mode Réseau Interne.
- L'exécution sur des machines physiques n'est pas autorisée.

7. Livrable : Rapport Technique

Les étudiants doivent fournir un rapport incluant :
- Un schéma représentant l'architecture globale.
- Une explication du code réalisé.
- Des captures d'écran d'exécution.
- Les limites observées et des propositions d'amélioration.

8. Flexibilité et Directives de Développement

Les étudiants sont libres de choisir :
- Le langage (Python, C, Java, Go, etc.).
- Les frameworks pour le serveur ou le contrôleur.
- Les formats de données et les approches de communication.
- Toute fonctionnalité additionnelle.

Le code peut être développé progressivement sous forme de fonctions vides, templates à compléter ou approche guidée permettant aux étudiants de remplir étape par étape les parties manquantes.