Documentation Finale - LD_PRELOAD Malware FARSSI SAMY 3SI1

🔹 Introduction

Ce projet implémente un **malware basé sur LD_PRELOAD** qui intercepte les **credentials SSH** (nom d'utilisateur et mot de passe) envoyés par les clients à un **serveur OpenSSH**.

Objectif & Contexte :

- Attaque réalisée dans un **assumed breach** (post-exploitation)
- Extraction des credentials en se plaçant entre l'application et les appels systèmes
- Envoi des données à un **serveur de Command & Control (C2)**
- **Port knocking** pour masquer le serveur C2
- **Techniques d'évasion** pour cacher les connexions malveillantes

---

Fonctionnalités Principales

Interceptions des Credentials SSH

Le malware détourne les appels systèmes read() et write() pour capturer les **noms d'utilisateur, mots de passe** et **clés privées SSH**.

Fonctionnement :

- Injection de **LD_PRELOAD** pour intercepter les appels
- Identification des **clés SSH** et credentials
- Enregistrement dans credentials.log
- Envoi au **serveur C2**

Serveur de Command & Control (C2)

Le **serveur C2** reçoit et stocke les **credentials** volés.

Fonctionnalités :

- Enregistrement des machines infectées

- Récupération des **identifiants volés + ip**
- Communication avec les malwares actifs

Port Knocking (Furtivité & Accès Restreint)

Le serveur C2 **n'est pas accessible** tant que la séquence correcte de **port knocking** n'est pas réalisée.

Étapes :

1. Le serveur **n'écoute pas sur le port 5555** au démarrage
2. Exécution de la séquence correcte :

nc 127.0.0.1 8000

nc 127.0.0.1 8001

nc 127.0.0.1 8002

3. Le port 5555 s'ouvre et le **serveur C2 démarre**

Évasion & Anti-Forensic

Le malware met en œuvre des techniques de **furtivité avancées** pour éviter d'être détecté.

Techniques :
Cacher le processus de netstat, ss et lsof (Les ports ouverts par le malware sont masqués de netstat, ss et lsof). Cependant, ss et netstat peuvent lire directement dans /proc, ce qui nécessite d'aller plus loin avec un hook sur openat() et read()

Effacer les logs d'authentification pour éviter la détection
Supprimer les fichiers sensibles après exploitation

Fonctionnalités des Fichiers

hook_read.c (Interception des Credentials)

- Intercepte les appels read() pour capturer les credentials SSH.

- Recherche les chaînes **"BEGIN OPENSSH PRIVATE KEY"** pour détecter les clés SSH.
- Stocke les credentials dans credentials.log.
- Envoie les données au serveur C2 via c2_client.c.

---

hook_hide_ports.c (Furtivité & Masquage des Connexions)

- Intercepte netstat, ss, lsof pour masquer le port 5555.
- Filtre les résultats avant affichage à l'utilisateur.
- Empêche la détection des connexions du serveur C2.

---

hook_write.c (Surveillance des Données Transmises)

- Intercepte les appels write() pour surveiller les **données envoyées**.
- Capture les communications SSH en sortie.

---

c2_client.c (Communication avec le C2)

- Établit une connexion avec le **serveur C2**.
- Transmet les credentials volés.
- Utilise un **socket TCP** pour envoyer les données.

---

c2_server.c (Serveur de Commande & Contrôle)

- **Reçoit** et **stocke** les credentials envoyés par le malware.
- Identifie les machines infectées.
- Peut potentiellement exécuter des commande sur les machines compromises.

---

port_knocking.c (Sécurité & Activation du C2)

- Attend la bonne **séquence de ports** (8000, 8001, 8002).

- Déclenche l’ouverture du port 5555 après validation.
- Masque le **serveur C2** tant que la séquence correcte n’est pas complétée.

Installation & Déploiement

Compilation du Malware

make clean && make

Infection de la Machine Victime

Lancer SSH avec le malware injecté :

sudo LD_PRELOAD=$(pwd)/hook.so /usr/sbin/sshd -D

Connexion d’un Utilisateur Cible

Sur une autre machine, se connecter à la machine infectée :

ssh -i private_key.pem testuser@localhost

Déclenchement du Port Knocking

nc 127.0.0.1 8000

nc 127.0.0.1 8001

nc 127.0.0.1 8002

Une fois la séquence validée, **le serveur C2 démarre automatiquement**.

Tests & Résultats

Vérification de l'Interception des Credentials

- Les credentials SSH volés doivent être enregistrés dans credentials.log :

cat credentials.log

- Les données doivent être envoyées au **serveur C2**.

Vérification de la Furtivité

Tester la visibilité du **serveur C2** :

netstat -tulnp | grep 5555

ss -tulnp | grep 5555

Résultat attendu : aucun affichage (port caché).

Vérifier que les logs ont été supprimés :

cat /var/log/auth.log | grep "sshd"

Résultat attendu : aucune trace de connexion.

Explications Techniques

- **Fonctionnement du Linker & LD_PRELOAD**

LD_PRELOAD permet de **remplacé dynamiquement des fonctions systèmes** avant l'exécution d'un programme.

Injection dans sshd

Détournement des appels read() et write()
Récupération des credentials **sans modifier le binaire SSH**

- **Threads sous Linux**

Le malware utilise des **threads pour intercepter les données sans bloquer SSH**.

Gestion parallele des connexions
Asynchrone pour ne pas ralentir le processus principal
Synchronisation via des mutex pour éviter les conflits

Limitations & Perspectives d'amélioration

Il y'a une limitation avec les versions modernes d'OpenSSH, qui empêchent l'interception des mots de passe en clair. Seules les clés SSH ont pu être récupérées efficacement. De plus, bien que la suppression des logs ait été fonctionnelle, une

approche plus furtive avec la modification des logs en temps réel permettrait une meilleure évasion.

Conclusion

Ce projet **démontre comment LD_PRELOAD** peut être utilisé pour créer un **malware furtif** et un **serveur de commande & contrôle** avancées. Grâce à des techniques de **furtivité, port knocking et évasion**, il permet une exfiltration discrète des credentials SSH tout en restant caché des outils de détection classiques.