Résumé des objectifs et priorités pour la création du malware FUD (partie 1)

Dans le cadre de ce projet, vous devez fournir un **malware FUD** qui soit **rapide à générer**, **bypassant l'analyse statique, l'analyse dynamique** et **fonctionnel** aussi bien sous Windows 10 que Windows 11. L'architecture présentée en briques (PE Loader, Sandbox Evasion, DLL Lookup Stub, Payload) est indicative : vous pouvez imaginer une structure plus complexe, tant que vous conservez une **modularité** vous permettant d'ajuster rapidement les composants pour faire évoluer votre malware.

1. **PE Loader**
    - Cible : produire l'exécutable de base, de petite taille, difficile à détecter par un simple scan statique.
    - Vous pouvez l'assembler et le patcher avec les réglages adaptés pour réduire les faux positifs.
    - L'évaluation portera sur sa taille, son taux de détection et sa robustesse en analyse statique.
2. **Sandbox Evasion Module**
    - Ensemble de fonctions ou routines incorporées (par exemple dans un fichier .h ou .c) pour détecter un environnement suspect.
    - Doit **autoriser l'exécution** si le nom de la machine commence par **"CYCORP-"**, même si elle est virtualisée.
    - L'évaluation s'intéressera à la pertinence et l'efficacité de votre détection.
3. **DLL Lookup Stub**
    - Objectif : résoudre dynamiquement les appels aux API Windows, sans recourir aux imports habituels.
    - Critères de notation : techniques d'évasion employées, robustesse pour éviter les crashs (red team = fiabilité !), et capacité à demeurer discret vis-à-vis des signatures AV.
    - La partie la plus importante dans cette première partie.
4. **Payload**
    - Dans un premier temps, limitez-vous à un simple **MessageBox** ou fonctionnalité inoffensive.
    - Plus tard, vous développerez des injections plus avancées, mais il est prioritaire d'avoir une version **fonctionnelle** dès maintenant.
    - Son exécution dépend du stub, et votre but est de rester sous les radars.

Méthodologie de réalisation

- **Commencez** par réaliser une version **basiquement opérationnelle** de **chaque brique** (même si elle est rudimentaire).
- N'investissez pas tout votre temps sur une seule brique : un malware inutilisable ne sert à rien en red team.
- Assurez-vous que l'exécutable final **tourne** sur Windows 10/11 avec le flux semblable a:
    - Loader → Anti-Sandbox → DLL Lookup → Payload.
- **Le comportement "malveillant" sera approfondi plus tard (prochains cours)** : pour l'instant, visez d'abord la fiabilité et la furtivité du squelette de votre malware.

Évaluation finale

- **Taille** du binaire.
- **Maintenabilité** (structure propre, briques indépendantes).
- **Scores de détection** (AV, sandbox).
- **Efficacité** des routines d'évasion.
- **Inventivité** du développeur.
- **Qualité du rapport** : décrivez les techniques testées ou étudiées et justifiez vos choix par rapport aux alternatives.

En somme, votre priorité est de disposer d'un **malware modulable** et **opérationnel**, dont les briques peuvent être enrichies progressivement. Vérifiez dès à présent que chaque composant fonctionne, car un exécutable non exécutable n'a aucune valeur lors d'une mission red team.