**Introduction**

Dans le cadre de notre projet de cybersécurité, nous avons simulé une attaque ciblant une machine fonctionnant sous Windows. L'objectif principal de cette simulation était d'obtenir un reverse shell sur la machine cible en utilisant un payload personnalisé, transmis via une campagne de phishing sophistiquée.

La méthodologie adoptée pour ce projet repose sur l'utilisation d'un payload spécialement conçu pour établir une connexion UDP inverse (reverse shell). Le payload est déguisé en un logiciel légitime de récupération de données perdues ou effacées, une approche couramment utilisée dans les scénarios de phishing. La diffusion du payload a été réalisée par l'envoi d'un e-mail attrayant et convaincant, rédigé avec soin pour inciter la cible à télécharger et exécuter le fichier malveillant.

Reconnaissance

La première étape d'une cyberattaque consiste en la collecte d'informations essentielles sur la cible. Cette phase, cruciale pour la réussite de l'attaque, implique l'identification et la collecte de données sur divers aspects de la cible.

1. Description

Dans le cadre de notre projet, la phase de reconnaissance a été directement orientée par les exigences du sujet : l'attaque devait être dirigée contre une machine fonctionnant sous Windows 10, et le payload devait être transmis par e-mail pour obtenir un reverse shell via UDP. En raison de ces contraintes spécifiques, la reconnaissance s'est concentrée sur l'identification des caractéristiques de Windows 10 et des méthodes de livraison d’e-mail efficaces. Bien que la cible ait été clairement définie, cette étape a impliqué la collecte d'informations sur les versions et configurations potentielles de Windows 10, ainsi que sur les pratiques courantes de sécurité liées aux e-mails. Cette analyse nous a permis de concevoir un payload compatible et de rédiger un e-mail de phishing attrayant, en veillant à ce que le message et le fichier exécutable apparaissent suffisamment légitimes pour tromper la cible tout en contournant les filtres de sécurité.

1. Techniques utilises

Pour identifier les configurations potentielles d'un système Windows 10, nous avons utilisé des outils de scan et des scripts pour obtenir des informations sur les paramètres système, les services en cours d'exécution, et les logiciels installés. Comme Nmap qui peut être employés pour interroger le système et recueillir des détails sur les ports ouverts, les versions des logiciels, et les configurations de sécurité. Cette reconnaissance permet de cibler les vulnérabilités spécifiques à la configuration du système.

Armement

L'armement est l'étape où l'attaquant prépare le malware ou l'exploit qui sera utilisé pour infiltrer le système cible. Cette étape inclut la création, l'emballage et l'optimisation du payload.

1. Description

Dans notre projet, l'armement a impliqué la création du payload en C#, qui a été conçu pour obtenir un reverse shell. Ce payload a été ensuite compilé en un exécutable Windows et préparé pour être envoyé par e-mail.

Techniques Utilisées :

* Développement en C# : Utilisation de C# pour coder le payload, profitant de sa capacité à interagir directement avec le système Windows.

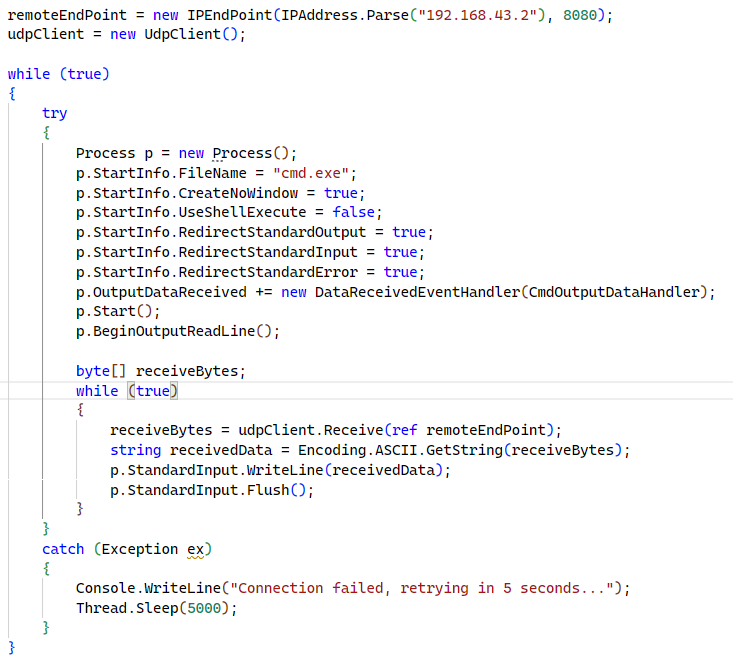


Figure Code source du payload

* Compilation en Exécutable : Compilation du code C# en un fichier exécutable (.exe) compatible avec Windows.
* Obfuscation et Déguisement : Le payload a été déguisé en logiciel de récupération de données pour masquer ses véritables intentions.

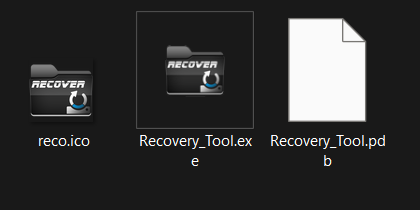


Figure Déguisement

* Tests de Compatibilité : Vérification du bon fonctionnement du payload sur différentes versions de Windows.
* Ingénierie Sociale : Conception du payload pour qu'il apparaisse comme un logiciel légitime de récupération de données, augmentant les chances que la cible l'exécute.

1. Difficulté

L'une des principales difficultés rencontrées durant l'armement a été d'éviter la détection par les logiciels anti-malware et les mécanismes de sécurité intégrés à Windows. Les solutions antivirus modernes sont particulièrement efficaces pour repérer et bloquer les exécutables malveillants.

Pour surmonter cette difficulté, nous avons adopté les stratégies suivantes :

* Évasion des Anti-Malwares : Nous avons utilisé des techniques d'obfuscation pour masquer les fonctionnalités malveillantes du payload. Cela incluait la modification de signatures de code et l'utilisation de techniques de polymorphisme pour rendre le code plus difficile à analyser.



Figure 3 Resultats analyse de Windows defenders

* Raccourcissement d'URL : Le payload a été intégré dans un lien TinyURL pour masquer son véritable objectif et réduire les chances de détection par les filtres de sécurité des e-mails.

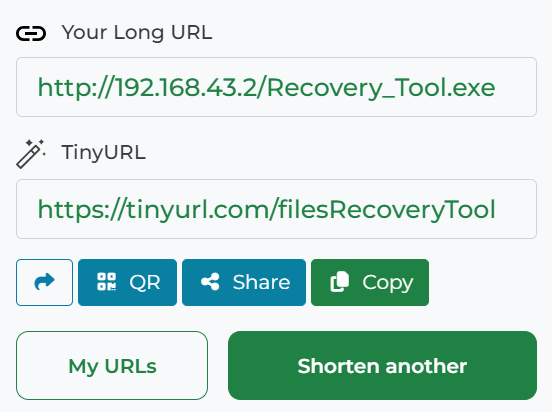


Figure Masquage de l'url

Livraison

L'étape de livraison consiste à transmettre le payload à la cible. Cette phase est cruciale car elle détermine si le malware atteindra ou non la machine cible.

1. Description

Dans notre projet, la livraison du payload a été réalisée par le biais d'une campagne de phishing soigneusement conçue. L’e-mail de phishing a été rédigé pour apparaître comme une annonce légitime de logiciel de récupération de données, incitant la cible à cliquer sur un lien de téléchargement. L’e-mail a été envoyé à l'aide d'un script Python utilisant le protocole SMTP, ce qui nous a permis d'automatiser et de personnaliser l'envoi.

1. Techniques utilisés

* Création d'Email de Phishing : Rédaction d'un e-mail attrayant et convaincant, utilisant des techniques d'ingénierie sociale pour inciter la cible à télécharger et exécuter le payload.
* Utilisation de Python et SMTP : Envoi automatisé de l'email de phishing via un script Python utilisant le protocole SMTP.

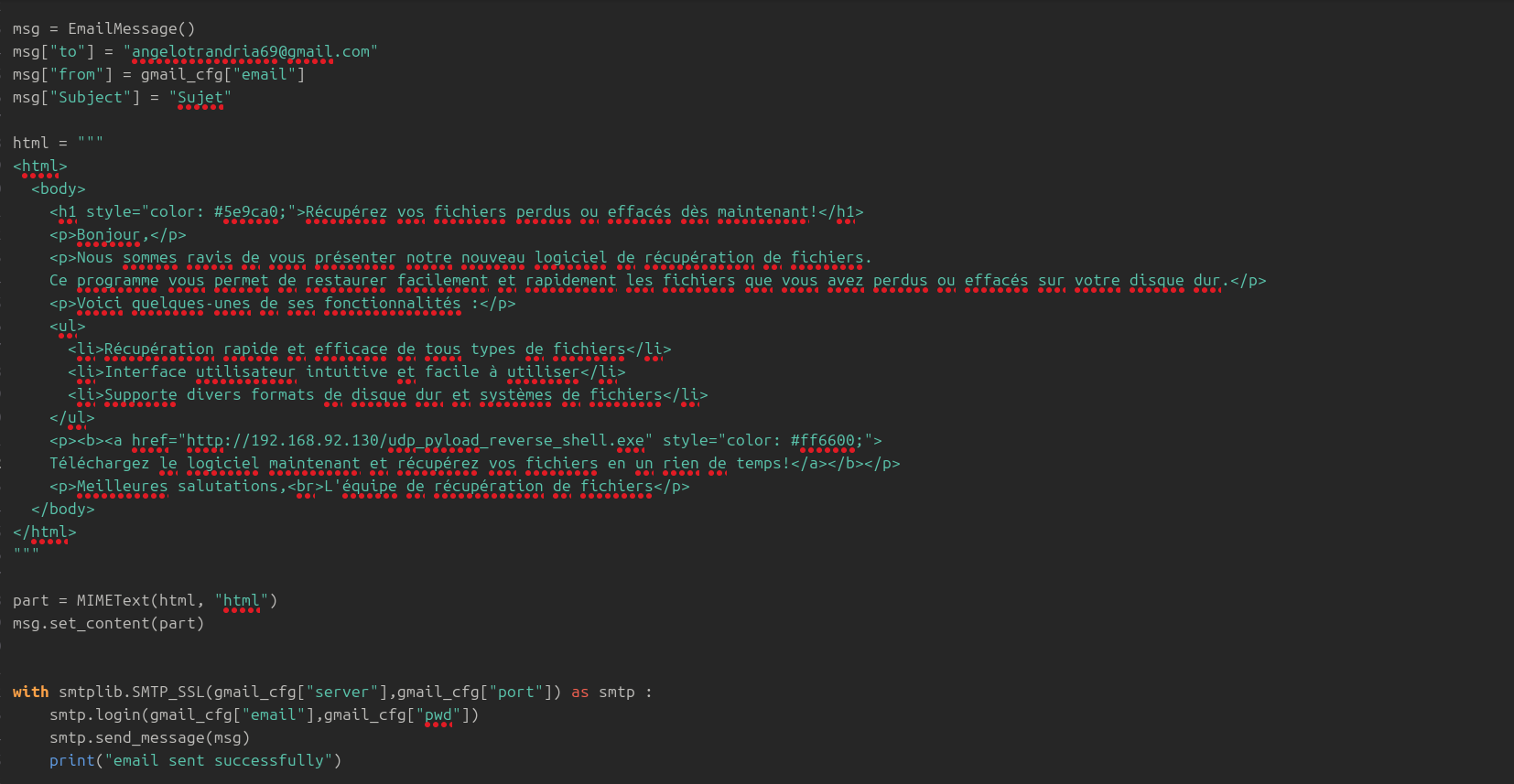


Figure 5 Script python d'envoie de mail avec SMTP

* Raccourcissement d'URL : Intégration du payload dans un lien TinyURL pour masquer la nature du fichier et augmenter les chances de contournement des filtres de sécurité.

1. Difficulté

L'une des principales difficultés rencontrées durant la livraison a été d'assurer que l’e-mail de phishing ne soit pas détecté et filtré par les systèmes de sécurité de l’e-mail de la cible. Les filtres anti-spam et les logiciels de sécurité des e-mails sont devenus de plus en plus sophistiqués, rendant difficile l'envoi de messages contenant des liens potentiellement malveillants.

Pour surmonter cette difficulté, nous avons mis en œuvre plusieurs stratégies. L’e-mail a été rédigé de manière à imiter le style et le ton des communications officielles, en évitant les mots et les phrases souvent associés aux messages de phishing. De plus, nous avons effectué plusieurs tests en envoyant des e-mails de phishing à des comptes de messagerie de test pour affiner notre approche et garantir que l’e-mail puisse passer à travers les mesures de sécurité en place.



Figure 6 E-mail de phishing

Ces efforts combinés ont permis de livrer le payload avec succès à la cible, en contournant les obstacles posés par les systèmes de sécurité des e-mails.

Exploitation

L'étape d'exploitation consiste à tirer parti des vulnérabilités de la cible pour exécuter le payload et obtenir l'accès souhaité au système.

1. Description

Dans notre projet, l'exploitation a été réalisée lorsque la cible a cliqué sur le lien de téléchargement dans l’e-mail de phishing et a exécuté le payload déguisé en logiciel de récupération de données. Une fois exécuté, le payload a établi une connexion de reverse shell permettant à l'attaquant de prendre le contrôle du système cible.

1. Techniques utilisées

* Ingénierie Sociale : Rédaction d'un email convaincant pour inciter la cible à exécuter le payload.
* Exécution du Payload : Le payload en C#, converti en exécutable, a été conçu pour s'exécuter discrètement et établir une connexion de reverse shell.
* Établissement d'une Connexion UDP : Utilisation du protocole UDP pour établir la connexion de reverse shell, évitant certaines détections de pare-feu.

Installation

L'étape d'installation consiste à implanter une backdoor ou un autre logiciel malveillant persistant sur le système cible, permettant un accès continu pour l'attaquant.

1. Description

Dans notre projet, l'installation a impliqué l'implantation d'une backdoor sur la machine cible via le payload, assurant un accès persistant même après un redémarrage du système.

1. Techniques utilisées

* Backdoor Intégré : Le payload a été conçu pour créer une backdoor sur le système cible, permettant une connexion de reverse shell UDP récurrente.
* Persistant Service : Modification des paramètres de démarrage de Windows pour inclure le lancement automatique de la backdoor lors de chaque démarrage du système.

Commande et Contrôle

L'étape de commande et contrôle consiste à établir un canal de communication sécurisé entre l'attaquant et le système compromis, permettant à l'attaquant de recevoir des informations et de donner des instructions à la machine cible.

1. Description

Dans notre projet, la commande et le contrôle ont été réalisés en maintenant la connexion de reverse shell UDP établie par le payload, permettant à l'attaquant de contrôler la machine cible.

1. Technique utilisée

Serveur C2 : Déploiement et configuration d'un serveur de commande et contrôle pour gérer les communications avec le payload sur la machine cible.

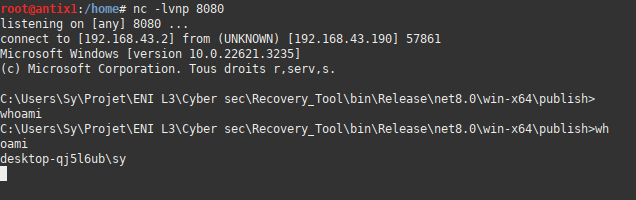


Figure Server C2 netcat

1. Difficulté

Maintenir une connexion stable et non détectée avec le serveur C2 a été un défi majeur, en particulier avec les pares-feux et les systèmes de détection d'intrusion en place.

Pour surmonter cette difficulté :

* Techniques d'Obfuscation de Réseau : Utilisation de techniques d'obfuscation pour masquer le trafic de communication avec le serveur C2.
* Surveillance en Temps Réel : Surveillance active des communications pour détecter et répondre rapidement à toute tentative de détection ou de blocage.

Grâce à ces mesures, nous avons réussi à établir et maintenir une connexion C2 stable, permettant une communication continue et sécurisée avec le système compromis.

Actions sur l’objective

L'étape des actions sur l'objectif consiste à réaliser les objectifs finaux de l'attaque, qui peuvent inclure la collecte de données, la perturbation des services, ou l'exécution de commandes malveillantes sur le système cible.

1. Description

Dans notre projet, les actions sur l'objectif ont consisté à utiliser le reverse shell pour explorer et manipuler le système cible, démontrant ainsi le succès de la simulation d'attaque.

1. Technique utilisée

* Exploration du Système : Utilisation du reverse shell pour naviguer dans les répertoires du système cible, identifier les fichiers et les configurations importantes.
* Exécution de Commandes : Exécution de diverses commandes sur le système cible pour démontrer le niveau de contrôle obtenu.

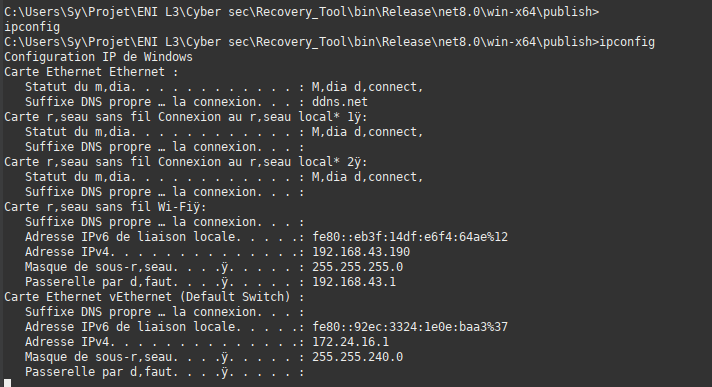


Figure Execution de Commande ipconfig

* Collecte de Données : Extraction de données sensibles pour prouver l'efficacité de l'attaque.

1. Difficulté

Maintenir la discrétion tout en exécutant les commandes et en collectant les données a été une difficulté majeure, car toute activité suspecte pourrait déclencher des alertes de sécurité.

Pour surmonter cette difficulté :

* Commandes Discrètes : Utilisation de commandes discrètes qui n'éveilleraient pas de soupçons immédiats.
* Contrôle des Ressources : Surveillance et gestion de l'utilisation des ressources système pour éviter de déclencher des alertes basées sur des comportements anormaux.