# Projet KeyLogger

## Labo SSI Ynov

Groupe 6

Antoine Rebérat
Yanice Hourcade
Dillan Huon

2024

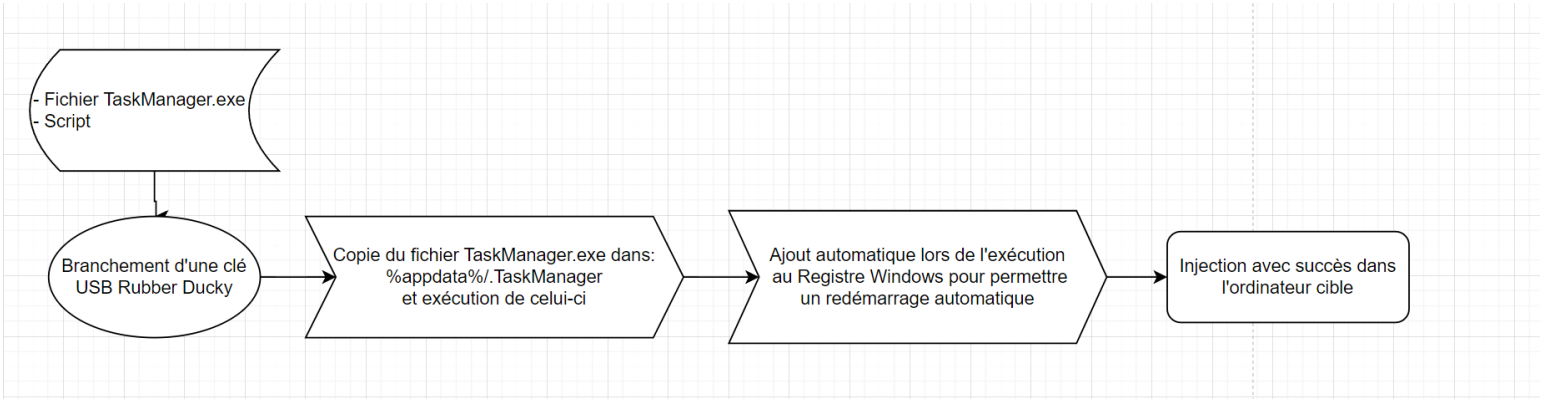
# Table des matières

# Introduction

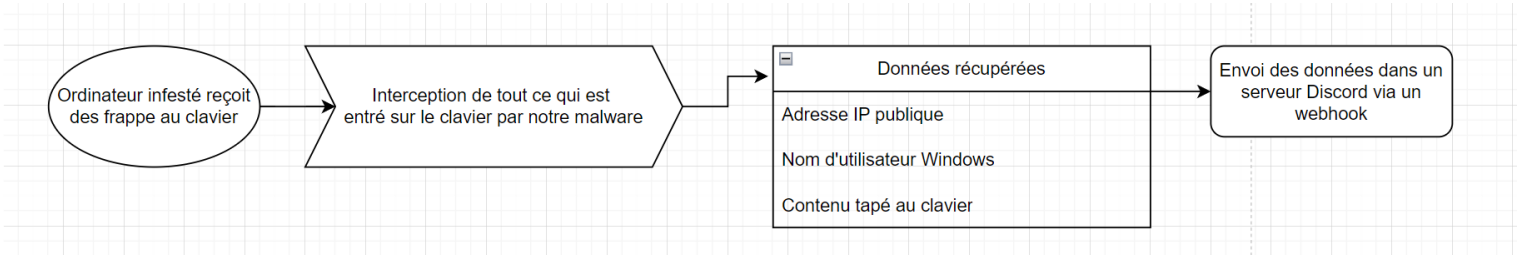
Dans le cadre du Ydays Labo SSI (*Système de sécurité informatique*), nous devions produire tout au long de l'année un projet en rapport avec la cybersécurité. Pour cela nous avons développé un malware de type KeyLogger en C# avec comme plateforme cible Windows.

# Schémas

## Injection




## Exploitation




# Justification du projet

Le choix d'un KeyLogger comme type de malware à développer est basé sur la capacité de ce genre de virus à récolter des informations. Dès l'instant où nous pouvons intercepter toute action réaliser au clavier nous pouvons facilement reconstituer les actions de l'utilisateur (sites web consultés, recherches récentes, documents tapés, …), ainsi que

récupérer les données confidentielles (adresse e-mail, adresse physique, mot de passes, …). Ce qui fait donc des KeyLogger des outils puissants pour les attaquants.

Nous avons également basé notre choix sur la simplicité de sa création, étant en B1 Informatique nous n'avions pas beaucoup de cordes à notre arc donc le choix de la difficulté du projet été un point de clé dans notre décision.

# Mise en place

Pour mettre en place l'utilisation du malware suivez ces étapes :

1. - Équipez vous d'une clé USB Rubber Ducky
2. - Téléchargez le malware TaskManager.exe sur le github (https://github.com/Aube33/keylogger-labo-ssi/releases)
3. - Téléchargez le script inject.bin pour la Rubber Ducky sur le github (https://github.com/Aube33/keylogger-labo-ssi/releases)
4. - Placez le fichier TaskManager.exe à la racine de votre Rubber Ducky
5. - Placer le fichier inject.bin à la racine de votre Rubber Ducky
6. - Brancher la clé sur l'ordinateur cible
7. - Ajouter le WebHook discord sur un de vos serveurs (lien disponible sur le Github)

# Problèmes

Lors du développement nous avions rencontrés plusieurs problèmes :

- Quel langage de programmation utiliser ?

Pour répondre à cette question nous nous sommes concentré sur lequel serait le plus pratique pour notre plateforme cible, et étant donné que nous voulions cibler les ordinateurs sous Windows le choix du C# fut une évidence grâce à ces librairies de développement pour Windows. Grâce à ce choix la récupération des entrées du clavier ainsi que l'intégration dans le système d'exploitation étaient des tâches bien plus aisées que nous le croyons.

- Comment injecter notre malware ?

Dans un premier temps nous avions prévu de cacher notre malware dans un logiciel déjà existant sauf qu'au vu de la complexité de la tâche et du peu de résultat concret que nous obtenions nous nous sommes plutôt tourner vers une alternative plus simple grâce aux attaques HID (Human Interface Device), et notamment les clés USB Rubber Ducky.

- Comment extraire les informations interceptées ?

À ce stade nous parvenions à écrire le contenu tapé au clavier dans un fichier texte dissimilé, mais après une courte réflexion c'était vraiment inutile. Donc l'initiative de rediriger les données interceptées vers un salon Discord a été proposé par un Dirlab du Labo SSI. Nous avions donc utilisé la technologie des Webhooks discord pour mettre en place ce système grâce à de simples requêtes effectuées par l'ordinateur cible.

- Comment intercepter les frappes du clavier ?

C'était notre plus gros problème car nous ne savions pas si il fallait plutôt partir vers une simple application invisible qui tourne en fond, ou bien un service Windows. Au vu des avantages proposés par les services Windows (démarrage automatique et exécution en fond notamment) nous nous sommes tournés vers cette solution en premier lieu. Sauf qu'on apprendra plus tard dans le développement que les services Windows n'ont pas accès aux interfaces de l'utilisateur, soit qu'on ne pouvait pas récupérer les frappes du clavier. Nous avons donc repris le développement depuis zéro en nous tournant cette fois ci vers une application Windows qui tourne en fond et se lance toute seule au démarrage et qui récupère correctement les touches du clavier utilisées par l'utilisateur grâce à l'utilisation des Hooks sur l'API Win32 de Windows (plus de détails ici)

# Axes d'amélioration

Pour l'instant notre malware est basique, voici les fonctionnalités qui peuvent être ajoutées :

- Obfuscation du code source : Obfusquer notre code source nous permettrait une plus grande sécurité en tant qu'attaquant en évitant une lecture du code source.

- Cacher le programme dans le gestionnaire des tâches : Actuellement lorsque notre programme se lance il reste affiché dans le Gestionnaire des Tâches de Windows. L'idéale serait donc de le mêler à la masse des programmes qui tournent en fond afin de cacher le malware.

- Redémarrage automatique du programme si il est stoppé : Lorsque l'ordinateur se démarre le programme se lancer automatiquement mais si l'utilisateur parvient stopper le programme manuellement celui-ci ne se relance pas tout seul après quelques secondes. Le mieux serait donc de réaliser cette amélioration.

# Conclusion

Pour finir, ce projet nous a sensibilisé à la cybersécurité et à l'efficacité des malwares de type KeyLogger notamment. Nous sommes mieux préparés à contrer ce genre d'attaque grâce à un compréhension plus approfondie.

Nous avons également acquis de nouvelles compétences et connaissances. Tout d'abord sur la compréhension du fonctionnement basique d'un KeyLogger, bien que son utilisation soit simple elle reste tout à fait intéressante car elle marque bien la relation entre l'attaquant et la cible (sur l'extraction et l'envoi des données interceptées).

Nous avons aussi de nouvelles connaissances sur le développement de logiciel sous Windows, notamment la programmation en C# et l'utilisation de l'API Win32 et l'existence des services Windows.