

# SPYWARE



VERSION 1.5

# HISTORIQUE DES VERSIONS

Versions	Date	Auteurs	Commentaire
1.0	08/11/2023	Alexandre, Maxime	Screen pc et informations pc
1.1	22/11/2023	Alexandre, Maxime	Camera
1.2	17/01/2024	Alexandre	Password chrome
1.3	14/02/2024	Maxime	Password Windows
1.4	06/03/2024	Alexandre, Maxime	Keylogger
1.5	20/03/2024	Alexandre, Maxime	Wifi Stealer

# TABLE DES MATIERES

I.	Introduction .....	3
II.	Conceptions et fonctionnalités .....	4
III.	Méthodes d'injection.....	8
IV.	Défis et solutions .....	9
V.	Impacts et réflexions .....	10
VI.	Conclusion.....	11
	Annexes .....	11

## I. Introduction

Notre voyage dans l'univers de la cybersécurité a débuté par une exploration de diverses idées potentielles pour notre projet, incluant le ransomware, le malware classique, les attaques par botnet DDoS, et l'utilisation de dispositifs comme la rubber ducky. Nous avons initialement opté pour cette dernière, un choix qui nous a mené à l'acquisition d'un Digispark. Le Digispark, sorte de frère non officiel de l'Arduino présenté via Kickstarter, est une petite carte équipée d'un simple régulateur de tension et d'un microcontrôleur Attiny85. Rapidement, nous avons découvert que ses capacités limitées en terme de stockage et sa base Arduino ne convenaient pas à nos ambitions.

Face à ces contraintes, nous avons tenté d'utiliser un Raspberry Pi Zero, espérant bénéficier d'un framework spécialement adapté aux rubber ducky. Cependant, notre enthousiasme a été tempéré par la désuétude du framework et la complexité des modifications requises, domaines pour lesquels nous n'avions pas l'expertise nécessaire. Après trois séances de travail ardu et frustrant, nous avons décidé de changer de cap.

L'idée d'un malware est restée en suspens, mais nous avons conservé notre rubber ducky, envisageant de l'utiliser pour l'injection du code. Sans savoir précisément comment développer un malware ni quel langage utiliser, nous avons été chanceux de pouvoir compter sur les conseils de notre ami Julien Chamouveau. Grâce à ses explications sur les bases et les démarches à suivre, notre projet a pris une nouvelle direction. Nous avons opté pour la création d'un spyware, une alternative qui nous semblait abordable. En nous plongeant dans des vidéos YouTube et divers dépôts Git, nous avons acquis suffisamment de connaissances pour développer un premier script en Python capable de capturer des captures d'écran toutes les secondes. Ce fut le début de notre aventure dans le développement de notre Spyware.

Inspirés par d'autres projets, nous avons ensuite élargi les capacités de notre spyware, intégrant la capture vidéo via la caméra et le vol d'informations sensibles, le tout orchestré depuis notre serveur Discord grâce à des webhooks. Ce projet, démarré sur des bases incertaines, a évolué en une exploration passionnante de la cybersécurité.

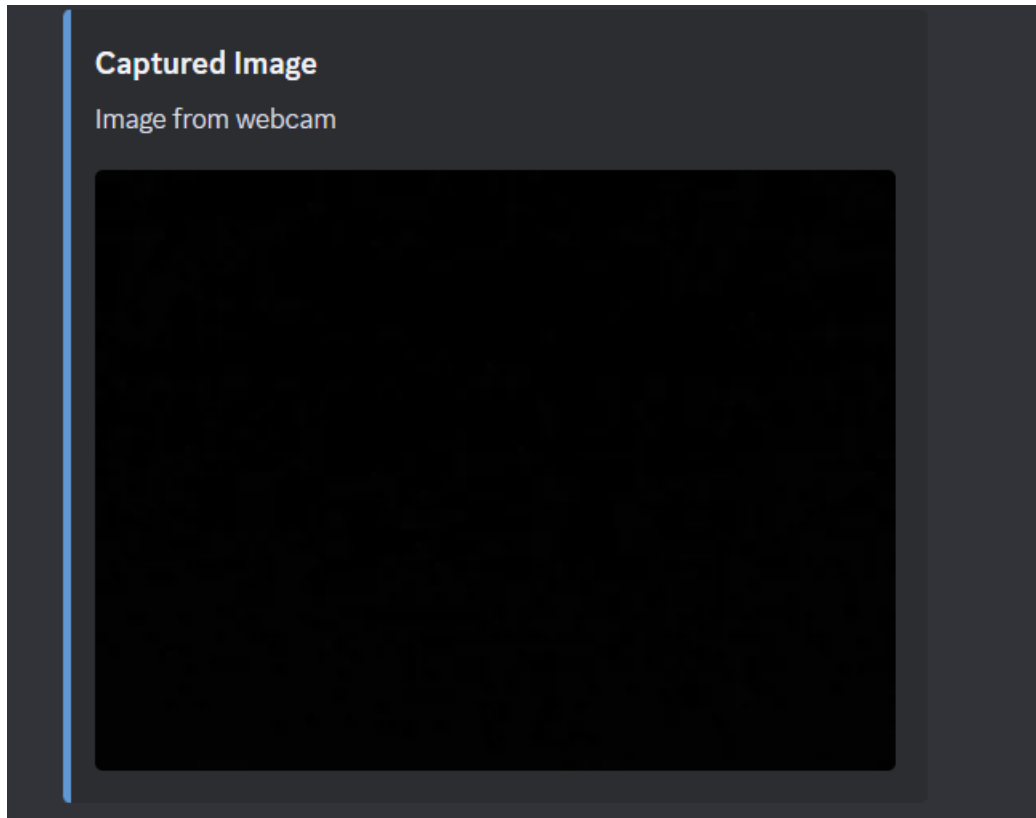
Dans le cadre de notre projet, nous avons porté notre choix sur deux technologies principales : Python et PowerShell. Ces choix ont été dictés par leur accessibilité, leur flexibilité et leur pertinence pour les tâches que nous envisagions.

Nous avons opté pour Python en raison de sa simplicité et de sa large adoption dans la communauté de développement, notamment pour des projets de sécurité informatique et de hacking éthique. Python est reconnu pour sa syntaxe claire et concise, ce qui en fait un excellent choix pour des programmeurs à notre niveau, permettant de se concentrer sur la logique sans se perdre dans la complexité du code. De plus, Python dispose d'une vaste bibliothèque de modules et d'outils tiers qui facilitent grandement

Quant à PowerShell, nous l'avons choisi pour rédiger le script d'installation que notre rubber ducky devait exécuter.

4

stabiliser la caméra, une image est capturée et enregistrée localement avant d'être également envoyée sur Discord. Ce processus permet une surveillance vidéo discrète sans interaction ou consentement de l'utilisateur.



## Vol de mots de passe et de Wi-Fi

Vol de Wi-Fi : Le script exécute la commande netsh wlan export profile key=clear pour extraire les profils Wi-Fi avec les mots de passe en clair. Les données sont ensuite extraites des fichiers XML générés, compilées dans un fichier texte, et nettoyées pour ne laisser aucune trace sur le système hôte.

```
[ ] SSID: P4wnP1
[!] Password: MaMe82-P4wnP1

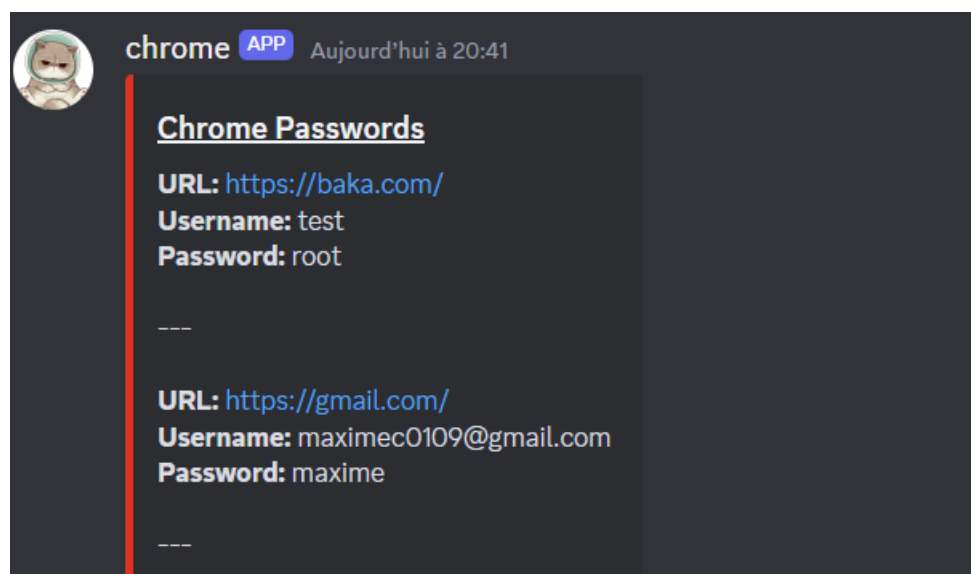
[ ] SSID: SFR_662F
[!] Password: 3ll4mp8r4pnys5hwbqn4

[ ] SSID: Téléphone-Diane
[!] Password: 7a40SEP...

[ ] SSID: WiFi@YNOV
[!] Password: none

[*] SSID: _SNCF_WIFI_INOUI
[!] Password: none
```

Vol de mots de passe : Pour les mots de passe stockés dans Chrome, nous avons adapté un script existant pour utiliser les fonctions cryptographiques de Chrome, récupérant la clé de chiffrement et déchiffrant les mots de passe stockés. Ces mots de passe sont ensuite envoyés via Discord pour une récupération à distance.



The screenshot shows a Discord message from the 'chrome' app, timestamped 'Aujourd'hui à 20:41'. The message content is as follows:

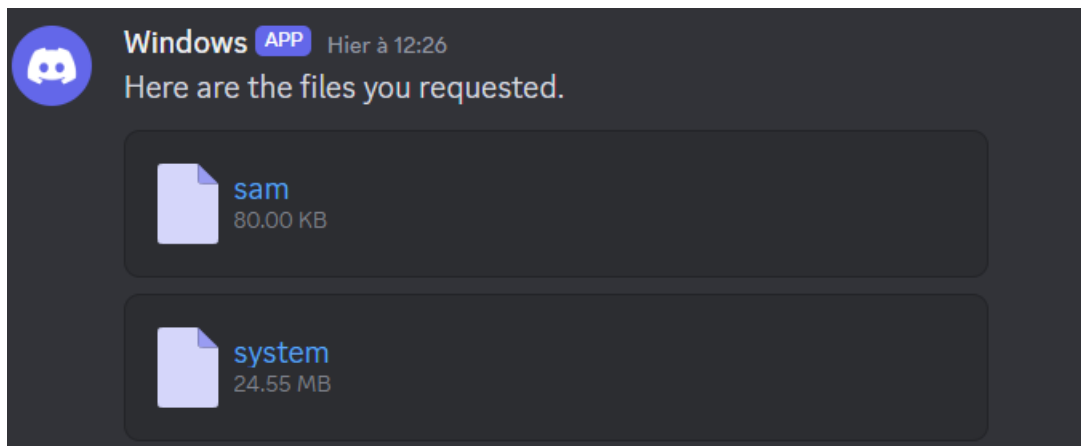
```
Chrome Passwords
URL: https://baka.com/
Username: test
Password: root

---

URL: https://gmail.com/
Username: maximec0109@gmail.com
Password: maxime

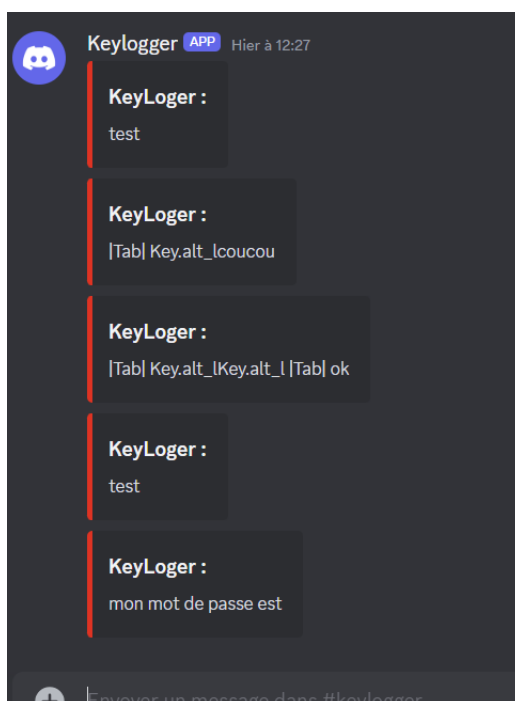
---
```

Vol du fichier SYSTEM et SAM : Pour ces 2 fichiers stockés dans Windows, nous avons utilisé regsave pour les dump et les envoyer via Discord.



## Keylogger

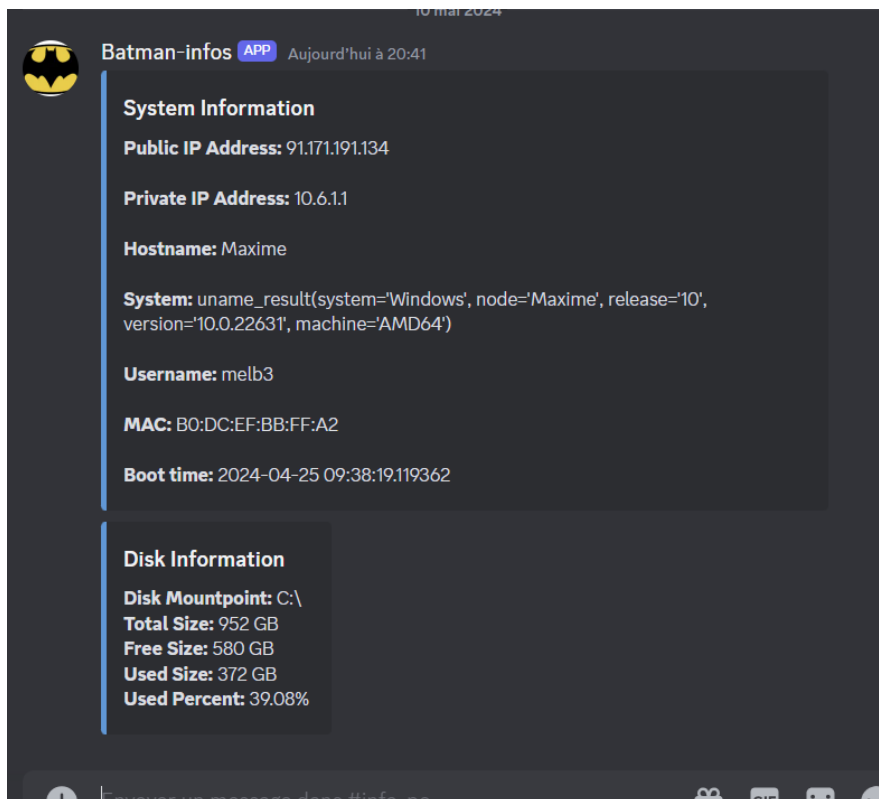
Le keylogger intégré capture toutes les frappes clavier, stockant les données localement avant de les envoyer périodiquement via Discord. Nous avons utilisé la bibliothèque pynput pour écouter les frappes clavier et gérer les touches spéciales comme la touche Entrée ou Espace. Cet outil présente à la fois des défis techniques, pour assurer une capture fluide et discrète.





## Informations sur le PC

Notre spyware récupère également des informations détaillées sur le système, comme l'adresse IP publique, le nom du processeur, le nom du GPU, et d'autres détails système via des commandes wmic. Ces informations sont cruciales pour un profilage détaillé de la machine cible et sont envoyées au serveur de contrôle pour analyse.



## III. Méthodes d'Injection

### Le spyware peut être déployé de plusieurs manières

Via Rubber Ducky : Un script PowerShell est exécuté lorsque la Rubber Ducky est connectée à un système, installant le spyware de manière furtive.

Via un lien de téléchargement : Le spyware peut aussi être téléchargé à partir d'un lien malveillant, souvent masqué derrière une façade légitime.

Via Reverse Shell : Si un shell inversé est déjà en place, le spyware peut être injecté et exécuté à distance.

L'emploi d'une Rubber Ducky pour déployer notre spyware s'est avéré être une méthode extrêmement efficace pour une exécution discrète et rapide. La Rubber Ducky, a été chargée avec un script PowerShell personnalisé conçu pour automatiser l'installation du spyware dès que le périphérique USB est connecté à un ordinateur.

### Script PowerShell

Le script utilisé dans notre Rubber Ducky est essentiellement conçu pour télécharger discrètement le spyware depuis un serveur distant et l'exécuter sans interaction utilisateur.

## IV. Défis et Solutions

Lors de notre première tentative avec la rubber ducky utilisant le Raspberry Pi Zero, nous avons rencontré des difficultés dues à l'utilisation d'un framework désuet. Ce framework n'était plus maintenu, ce qui rendait sa mise à jour et son adaptation à nos besoins particulièrement compliquée, surtout avec nos compétences limitées à ce stade. Ce premier échec nous a contraint à reconsidérer notre approche et à pivoter vers le développement d'un spyware, un domaine dans lequel nous étions également novices.

Le démarrage du projet de spyware a été marqué par une incertitude sur la direction à prendre et la manière d'implémenter les fonctionnalités souhaitées. Sans une compréhension claire de la logique sous-jacente ou des techniques de développement de malwares, nous étions bloqués. C'est grâce à l'intervention d'un ami, Julien Chamouveau, qui nous a fourni des explications sur les bases du développement de malware, que nous avons pu commencer à structurer notre projet.

Un autre défi technique fut le déchiffrement des mots de passe stockés localement par les navigateurs et le système d'exploitation. Les mots de passe dans Chrome, par exemple, sont chiffrés, et déchiffrés, ces données sans comprendre les mécanismes de cryptographie appropriés fut initialement un obstacle majeur. Après de nombreuses recherches et l'exploration de divers dépôts GitHub, nous avons adapté des scripts existants pour déchiffrer ces mots de passe en utilisant les clés de chiffrement extraites du système de l'utilisateur.

Un des défis les plus significatif que nous avons rencontré lors du développement de notre spyware a été la récupération des mots de passe des sessions utilisateurs de Windows. Bien que nous ayons réussi à copier et à envoyer les fichiers SAM et SYSTEM via Discord, qui sont essentiels pour l'accès aux informations d'authentification, le déchiffrement de ces fichiers pour extraire réellement les mots de passe a constitué un obstacle majeur.

La complexité du déchiffrement des fichiers SAM et SYSTEM repose sur le fait que ces fichiers utilisent des mécanismes de sécurité robustes conçus spécifiquement pour protéger les informations d'identification des utilisateurs contre les accès non autorisés. Cela inclut l'utilisation de hachages irréversibles pour stocker les mots de passe, ce qui complique grandement leur extraction directe.

Pour surmonter ce défi, des solutions alternatives telles que l'utilisation de Mimikatz ou Pypykatz, nous ont été suggérées. Cependant, même avec ces outils à notre disposition, le niveau de compétence technique requis pour les utiliser efficacement était au-delà de notre expérience en tant que débutants. Malgré plusieurs tentatives, nous n'avons pas réussi à maîtriser ces outils pour déchiffrer les mots de passe à partir des fichiers extraits

## **V. Impacts et Réflexions**

### **Sur le Plan Technique**

- **Compétences en programmation :** Nous avons significativement amélioré nos compétences en programmation et en utilisation de divers outils et langages, notamment Python et PowerShell. La nécessité de naviguer entre différents frameworks et bibliothèques nous a aidé à devenir plus agiles et adaptatifs dans notre approche du développement logiciel.
- **Connaissance en sécurité :** Le projet nous a permis de comprendre en profondeur les techniques utilisées par les malwares et les spywares, de la collecte de données au camouflage des activités du système.

### **Sur le Plan Personnel**

- **Conscience des risques :** Ce projet a élevé notre niveau de conscience concernant les risques associés aux logiciels malveillants. Nous avons réalisé l'importance de la sécurité informatique et l'impact potentiel de ces technologies sur la vie privée des individus.

- Développement de la logique : Nous avons développé une logique de résolution de problèmes beaucoup plus robuste. Face à des défis complexes, nous avons appris à décomposer les problèmes, à explorer des solutions alternatives, et à persévérer malgré les difficultés.

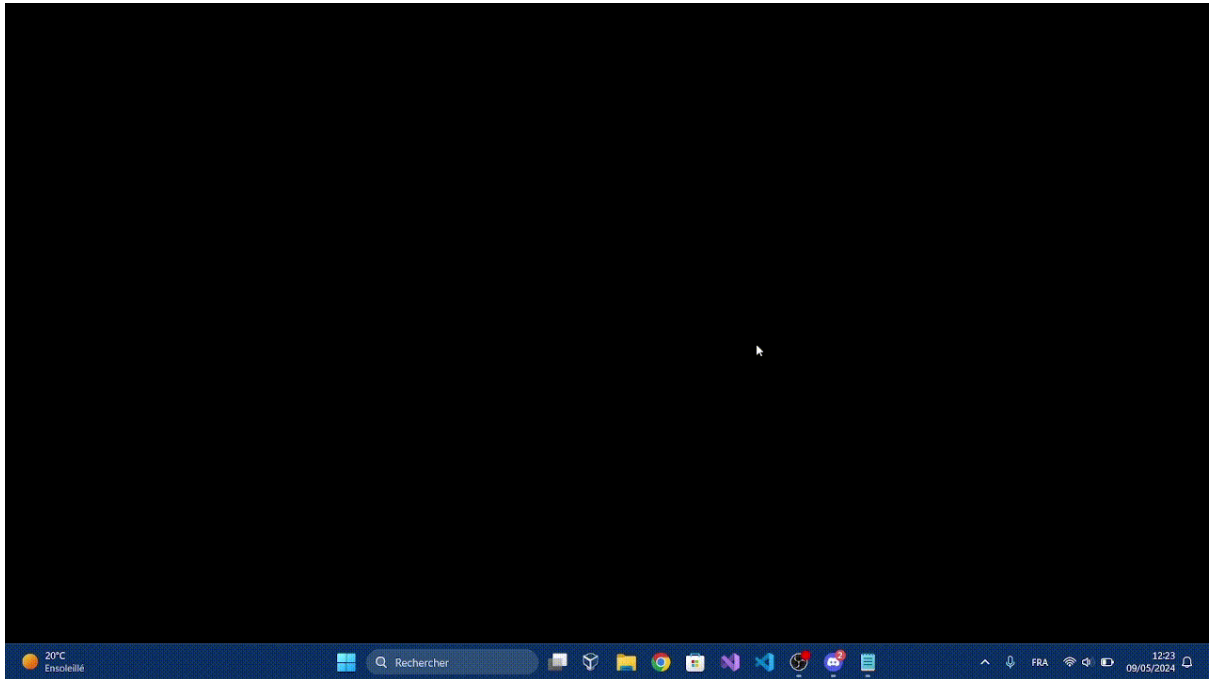
## VI. Conclusion

Notre projet spyware nous a permis de nous introduire dans le monde de la cybersécurité, nous dotant d'une compréhension pratique des méthodes et des techniques utilisées dans le développement de spyware. Tout au long du projet, nous avons été confrontés à des défis techniques qui ont stimulé notre capacité à résoudre des problèmes complexes et à penser de manière critique, tout en nous sensibilisant aux implications éthiques du déploiement de telles technologies. Les compétences apprises au cours de ce projet enrichissent notre arsenal professionnel. Ce projet a été non seulement une opportunité d'apprentissage, mais aussi un catalyseur pour une réflexion approfondie sur la manière dont nous pouvons contribuer à créer un environnement numérique plus sûr.

## Annexes

- **Code source** : Voici le lien de notre Github : [Spyware](#). Il est possible que notre spyware ne fonctionne pas correctement sur tous les PC, principalement à cause de problèmes liés à l'envoi du fichier system via Discord, qui peut ne pas être compatible avec les webhooks. Si cela se produit, vous pouvez désactiver cette fonction en mettant en commentaire la ligne 338 du code. Par ailleurs, l'extraction des mots de passe Wi-Fi peut également échouer sur certains ordinateurs, bien que nous ne soyons pas certains de la cause de ce problème. Pour contourner ce problème, mettez simplement la ligne 337 en commentaire. Enfin, pour une utilisation optimale de notre spyware, assurez-vous de mettre à jour le lien des webhooks dans les lignes 324 à 331 du script.

- **Démonstration**



Ou via ce [lien](#)

Ou via [Youtube](#)