Project CSC_5CS04_TP
Forensics des systèmes et des réseaux

Ransomware

Realisé par :

ABDELLAOUI Aicha

BEN AMOR Seifeddine

supervisé par :

M. Chamoun Maroun

Année académique : 2024/2025

Table des matières

Table des figures

Liste des tableaux

Introduction

Les ransomwares représentent l'une des menaces les plus redoutées dans le domaine de la cybersécurité. Ces logiciels malveillants cryptent les fichiers d'un utilisateur ou d'une organisation, rendant l'accès aux données vitales impossible, sauf si une rançon est payée. Les attaques par ransomware ont connu une croissance rapide en raison de leur rentabilité pour les cybercriminels et de l'impact dévastateur qu'elles peuvent avoir sur les entreprises et les particuliers. Elles peuvent perturber des opérations entières, entraîner des pertes financières importantes et nuire à la réputation des organisations.

Les ransomwares exploitent souvent des vulnérabilités dans les systèmes informatiques, ce qui rend crucial un processus de gestion proactive des risques pour protéger les données sensibles. L'augmentation des attaques par ransomware a conduit à un besoin croissant de solutions de cybersécurité avancées et d'audits de vulnérabilité réguliers pour identifier et atténuer ces menaces avant qu'elles ne causent des dommages irréparables.

Chapitre 1

Contexte du Projet

Introduction

Dans ce chapitre, nous allons faire une analyse du développement historique des ransomwares ce qui offre un aperçu précieux de l'évolution de cette menace cybernétique. Elle démontre la sophistication croissante et l'impact grandissant des attaques de ransomware, soulignant la nécessité d'une attention continue et de défenses robustes.

1.1 Les premiers ransomwares (avant 2010)

1.1.1 Formes initiales de ransomwares

Les origines des ransomwares remontent à la fin des années 1980. Le premier cas connu de ransomware, appelé le Trojan AIDS (ou PC Cyborg Trojan), est apparu en 1989 et a été distribué sur des disquettes par le Dr Joseph Popp. Joseph Popp a crypté les noms de fichiers et a exigé une rançon de 189 $ à envoyer à une boîte postale au Panama. Cette approche basique a été un moment charnière dans l'histoire des ransomwares, établissant le concept de cryptage des données de la victime dans le but d'extorquer de l'argent.

1.1.2 Exemples précoces et leurs caractéristiques

Aux débuts du phénomène, les attaques de ransomwares étaient relativement peu sophistiquées, tant en termes de méthodologie que de fonctionnalité. Le Trojan AIDS, par exemple, utilisait une méthode par laquelle les noms de fichiers étaient cryptés, les rendant ainsi inaccessibles. Les algorithmes de cryptage utilisés étaient relativement faibles, ce qui signifiait que les victimes pouvaient souvent trouver des

moyens de récupérer leurs données sans payer la rançon. Cette première forme de ransomware a posé les bases des développements futurs en montrant le potentiel de gain financier par le biais du cryptage de données et de l'extorsion.

1.2 Croissance des ransomwares (2010-2015)

1.2.1 Développement de ransomwares plus sophistiqués

La période entre 2010 et 2015 a connu des avancées significatives dans la sophistication des ransomwares. L'introduction d'algorithmes de cryptage plus puissants et de méthodes de distribution plus efficaces a marqué cette époque. Les attaquants ont commencé à utiliser le cryptage asymétrique, qui utilise une paire de clés (publique et privée) pour crypter et décrypter les données, rendant quasiment impossible pour les victimes de décrypter leurs fichiers sans la clé privée détenue par les attaquants.

1.2.2 Exemples clés tels que CryptoLocker

CryptoLocker, identifié pour la première fois en 2013, représente l'une des variantes de ransomwares les plus notoires de cette période. Le malware se propageait principalement via des pièces jointes d'e-mails, cryptant les fichiers de la victime et exigeant un paiement de rançon en Bitcoin. CryptoLocker utilisait le cryptage RSA, une méthode robuste qui rendait la récupération des données pratiquement impossible sans le paiement de la rançon. Cette attaque a démontré l'efficacité de la combinaison du cryptage fort et des méthodes de paiement anonymes, ouvrant la voie à une augmentation des attaques par ransomware.

1.3 Ransomwares modernes (2016 à aujourd'hui)

1.3.1 Attaques notables et tendances des dernières années

Depuis 2016, il y a eu une augmentation notable de la fréquence, de la sophistication et du potentiel destructeur des attaques de ransomwares. Cette période est caractérisée par un certain nombre d'attaques très médiatisées, dont WannaCry et NotPetya, qui ont causé des perturbations et des pertes financières importantes. Ces attaques ont exploité des vulnérabilités dans des logiciels largement utilisés, leur permettant de se propager rapidement dans les réseaux, touchant des milliers de systèmes dans le monde en quelques heures.

1.3.2 Évolution des tactiques et techniques des ransomwares

Les attaques modernes de ransomwares se caractérisent par plusieurs tendances et avancées clés :

— **Ransomware-as-a-Service (RaaS) :** Ce modèle permet même à ceux ayant peu de compétences cybercriminelles de lancer des attaques de ransomware. Les développeurs créent des ransomwares et les louent ensuite à des affiliés, qui les distribuent à leur tour. Les affiliés partagent les bénéfices générés par la distribution du ransomware. Cela a réduit le coût d'entrée et augmenté le nombre d'attaques.

— **Double extorsion :** En plus de crypter les données, les attaquants les exfiltrent fréquemment, menaçant de les publier ou de les vendre si la rançon n'est pas payée. Cela exerce une pression supplémentaire sur les victimes pour qu'elles se conforment.

— **Attaques ciblées :** Les groupes modernes de ransomware mènent souvent des recherches approfondies pour identifier des cibles de grande valeur, telles que les grandes entreprises, les agences gouvernementales et les infrastructures critiques. Ces attaques sont minutieusement conçues pour maximiser leur impact et les demandes de rançon.

1.4 Attaques de ransomwares notables

1.4.1 WannaCry (2017)

L'attaque par ransomware WannaCry est l'un des incidents de cybersécurité les plus significatifs de l'histoire. Le malware exploitait une vulnérabilité du système d'exploitation Windows, désignée sous le nom d'EternalBlue, pour se propager rapidement dans les réseaux. En quelques heures, WannaCry avait infecté plus de 200 000 ordinateurs dans 150 pays, perturbant les entreprises, les services de santé et les opérations gouvernementales. Les auteurs de l'attaque exigeaient des paiements de rançon en Bitcoin, mais un grand nombre de victimes n'ont pas pu récupérer leurs données même après avoir effectué les paiements requis. L'incident WannaCry a souligné la nécessité de mettre en œuvre rapidement les mises à jour logicielles et des mesures de cybersécurité robustes.

1.4.2 NotPetya (2017)

Bien que perçue initialement comme un ransomware, NotPetya était en réalité un wiper conçu pour causer un maximum de perturbations. Le malware s'est

propagé par le biais d'un mécanisme de mise à jour logiciel compromis d'un logiciel de comptabilité ukrainien populaire. NotPetya a chiffré le Master Boot Record (MBR), rendant les systèmes affectés inutilisables. Contrairement aux ransomwares conventionnels, il n'y avait aucun moyen de récupérer les données, même si la rançon était payée. L'attaque a causé des milliards de dollars de dommages, touchant des entreprises mondiales telles que Maersk et Merck. L'incident NotPetya a démontré le potentiel des cyber-guerres menées par des États sous le couvert de ransomware.

1.4.3 Autres attaques majeures

— **Ryuk :** est un groupe cybercriminel notoire impliqué dans plusieurs attaques de grande envergure contre des organisations importantes, notamment des hôpitaux, des écoles et des municipalités. Leur modus operandi consiste à cibler des entités de grande valeur et à exiger des rançons exorbitantes. Il est souvent observé que Ryuk suit une infection initiale par d'autres malwares, tels que TrickBot.

— **GandCrab :** GandCrab, qui fonctionnait sur un modèle RaaS, a été l'une des familles de ransomwares les plus prolifiques jusqu'à sa fermeture en 2019. Les développeurs de GandCrab ont affirmé avoir reçu plus de 2 milliards de dollars en paiements de rançons.

— **SamSam :** Ce ransomware ciblait des organisations spécifiques, notamment la ville d'Atlanta et le Département des Transports du Colorado. Les attaquants de SamSam déployaient le ransomware manuellement après avoir accédé aux réseaux, garantissant la plus grande perturbation possible.

— **REvil (Sodinokibi) :** a ciblé des organisations de premier plan, exigeant des rançons de plusieurs millions de dollars. Le groupe utilise une stratégie d'extorsion à deux volets et fonctionne sur un modèle RaaS.

— **LockBit :** est réputé pour sa vitesse de cryptage rapide et sa capacité à se propager dans les réseaux. La société a identifié la nécessité de fournir des mesures de sécurité interne robustes à plusieurs industries.

Conclusion

L'évolution des ransomwares illustre la nature dynamique et adaptable des cybermenaces. Chaque période a vu l'émergence de nouvelles techniques et tactiques qui ont repoussé les limites de la cybersécurité. Cette évolution souligne l'importance d'une vigilance continue et de solutions de sécurité avancées pour contrer l'impact croissant de ces attaques.

.

Chapitre 2

Analyse Technique des Attaques

Introduction

Ce chapitre se concentre sur l'analyse technique des ransomwares, en détaillant les différentes étapes qu'ils suivent pour infecter un système, crypter les données et extorquer une rançon. Nous examinerons les méthodes d'infection courantes, les mécanismes de cryptage utilisés, ainsi que les modalités de paiement de la rançon. Cette analyse permet de mieux comprendre les menaces posées par les ransomwares et d'adopter des stratégies de prévention adaptées.

2.1 Infection

Les ransomwares utilisent diverses méthodes pour infiltrer le système d'une victime. Les points d'entrée courants incluent :

- **Emails de Phishing :**
  - **Livraison du Payload :** Les payloads des ransomwares sont souvent déguisés en pièces jointes inoffensives (telles que des fichiers exécutables, des documents Office avec des macros, ou des archives compressées) ou intégrés dans des hyperliens dans les messages emails.
  - **Ingénierie Sociale :** Les emails de phishing utilisent des tactiques d'ingénierie sociale pour tromper les destinataires afin qu'ils ouvrent les pièces jointes malveillantes ou cliquent sur les liens. Cela peut impliquer de se faire passer pour des entités légitimes, de créer un sentiment d'urgence, ou d'exploiter des déclencheurs psychologiques pour manipuler le comportement de l'utilisateur.
- **Kits d'Exploitation :**

  - **Exploitation de Vulnérabilités :** Les kits d'exploitation exploitent les vulnérabilités connues dans les logiciels ou les systèmes d'exploitation pour livrer des payloads de ransomware silencieusement et sans interaction de l'utilisateur. Ces vulnérabilités peuvent exister dans des applications couramment utilisées, des navigateurs web ou des plugins.
  - **Téléchargements Automatiques :** Les kits d'exploitation sont souvent hébergés sur des sites web compromis ou malveillants. Lorsqu'un utilisateur visite un tel site, le kit d'exploitation analyse automatiquement le système du visiteur à la recherche de vulnérabilités et livre le payload de ransomware si une exploitation adaptée est trouvée.
- **Ports RDP Non Sécurisés :**
  - **Attaques par Force Brute :** Les attaquants peuvent tenter d'obtenir un accès non autorisé aux systèmes avec des ports RDP (Remote Desktop Protocol) exposés en devinant systématiquement les noms d'utilisateurs et les mots de passe. Cette méthode de force brute exploite les identifiants faibles ou par défaut pour franchir les défenses du système.
  - **Credential Stuffing :** Dans certains cas, les attaquants obtiennent des identifiants de connexion à partir de violations de données ou de marchés souterrains et utilisent des outils automatisés pour tester ces identifiants contre des services RDP exposés.

2.2 Cryptage

Une fois à l'intérieur du système de la victime, le ransomware initie le processus de cryptage. Il cible généralement une large gamme de types de fichiers, notamment les documents, les images, les vidéos et les bases de données. Pour crypter les fichiers, le ransomware utilise des algorithmes de cryptage robustes tels que l'AES (Advanced Encryption Standard) ou le RSA (Rivest-Shamir-Adleman). Ces algorithmes génèrent des clés cryptographiques utilisées pour crypter les fichiers de la victime, les rendant inaccessibles sans la clé de décryptage correspondante. Le processus de cryptage est souvent rapide et complet, garantissant qu'une grande partie des données de la victime est cryptée avant d'être détectée.

2.3 Demande de Rançon

Après avoir crypté les fichiers de la victime, le ransomware affiche une note de rançon à l'écran de l'utilisateur ou dans les répertoires cryptés. Cette note de rançon inclut généralement des instructions sur la manière de payer la rançon, ainsi que le montant demandé et l'adresse du portefeuille de cryptomonnaie à laquelle

le paiement doit être effectué. Les cybercriminels exigent souvent un paiement en cryptomonnaies comme le Bitcoin ou l'Ethereum en raison de l'anonymat et de l'irréversibilité des transactions. La note de rançon peut également inclure un délai d'expiration pour le paiement, accompagné de menaces de perte de données permanente ou d'autres conséquences si le paiement n'est pas effectué dans le délai spécifié.

2.4 Cryptomonnaies et le paiement de la Rançon

Les cryptomonnaies comme le Bitcoin sont privilégiées pour les paiements de rançon en raison de leur anonymat et de leur facilité de transfert. Les attaquants fournissent des instructions pour acheter et transférer de la cryptomonnaie, ce qui rend difficile pour les autorités de tracer les paiements.

Conclusion

Ce chapitre a exploré les étapes des attaques par ransomwares, de l'infection à la demande de rançon, en passant par le cryptage des fichiers. Les ransomwares exploitent des vulnérabilités pour pénétrer les systèmes et exigent un paiement en cryptomonnaies, rendant leur traçabilité difficile. Cette compréhension est essentielle pour se protéger contre ces menaces.

Chapitre 3

Réalisation

Introduction

Dans ce chapitre, nous aborderons les différentes phases de création de notre ransomware, depuis la planification jusqu'aux tests finaux. Nous détaillerons le processus de développement ainsi que les étapes de test mises en place pour assurer le bon fonctionnement de notre malware. Cette section permet de comprendre l'approche méthodologique adoptée pour la réalisation de notre projet.

3.1 Outils et Technologies

Ce projet utilise plusieurs outils et technologies pour implémenter le chiffrement et le déchiffrement des données, la gestion des transactions sur blockchain et l'intégration web.

- **Programmation :**
  - **Le langage C :** C est utilisé pour la création du programme principal, y compris le chiffrement et déchiffrement des fichiers. C permet un contrôle fin de la mémoire et une efficacité élevée, ce qui est crucial pour des opérations cryptographiques avancées.
  - **Patchelf :** Patchelf est un outil permettant de modifier les informations relatives aux liens dynamiques d'un exécutable ou d'une bibliothèque. Il est utilisé pour ajouter de nouvelles bibliothèques de réimplémentation des fonctions dans les fichiers exécutables.
- **Algorithme cryptographique :**
  - **AES :**AES est un algorithme de chiffrement symétrique.Il est utilisé pour chiffrer des fichiers, chacun avec une clé unique, de manière rapide et sécurisée.

  - **RSA :** RSA est un algorithme de chiffrement asymétrique. Il est utilisé pour chiffrer les clés publiques et les vecteurs d'initialisation (IV) d'AES pour chaque fichier.
  - **OpenSSL :** OpenSSL est une bibliothèque cryptographique pour le langage C. Elle permet d'intégrer facilement des algorithmes de chiffrement tels que RSA et AES dans le code C.
- **Développement Web :**
  - **Frontend :** HTML, CSS et JavaScript utilisés pour développer un site de téléchargement de ransomwares et un site de paiement de rançon
  - **Backend :** Un serveur backend en PHP implémenté pour communiquer avec le réseau Ethereum, vérifier les transactions et envoyer le lien du fichier de clé privée.
- **Blockchain :**
  - **Frontend :** MetaMask est une extension de navigateur pour la gestion des portefeuilles Ethereum. Elle facilite la connexion au réseau blockchain, permettant aux utilisateurs d'effectuer des transactions sécurisées. MetaMask est utilisée pour effectuer des paiements de rançon dans un réseau de test.
  - **Sopelia :** Le réseau Sopelia est un réseau de test (testnet) qui permet aux développeurs et utilisateurs de tester des applications, des contrats intelligents et des transactions sur une blockchain. Il est utilisé pour simuler les transactions sans avoir recours à de vrais fonds.

3.2 Planification du Ransomware

3.2.1 organigrammes

Pour illustrer le processus de chiffrement et déchiffrement ansi que le fonctionnementde notre malware, nous avons créé un organigramme qui décrit les différentes étapes impliquées. L'organigramme a servi de guide visuel pour nous aider à comprendre le fonctionnement du programme et les étapes du processus. L'organigramme est divisé en 3 sections principales : le chiffrement, le déchiffrements et la logique du ransomware.

Chaque section comprend un ensemble d'étapes à suivre pour détecter et exploiter la vulnérabilité correspondante.

Mécanisme de fonctionnement du ransomware

Lorsqu'il est lancé, le programme commence par générer une nouvelle bibliothèque `libwrapper.so` qui intercepte les fonctions des bibliothèques standard

printf, puts et strcmp. Par la suite, il utilise le chemin donné en argument pour débuter le chiffrement du répertoire et de tous les fichiers qu'il contient d'une manière récursive. Ensuite, il affiche un message indiquant que le répertoire est chiffré et fournit un lien pour payer la rançon, laquelle augmente toutes les 48 heures. Si la rançon est payée, la clé de chiffrement sera téléchargée. Il suffit de fournir le chemin vers la clé au programme pour que tout revienne à l'état initial.

La figure 3.1 représente l'organigrame de mécanisme de fonctionnement du ransomware.

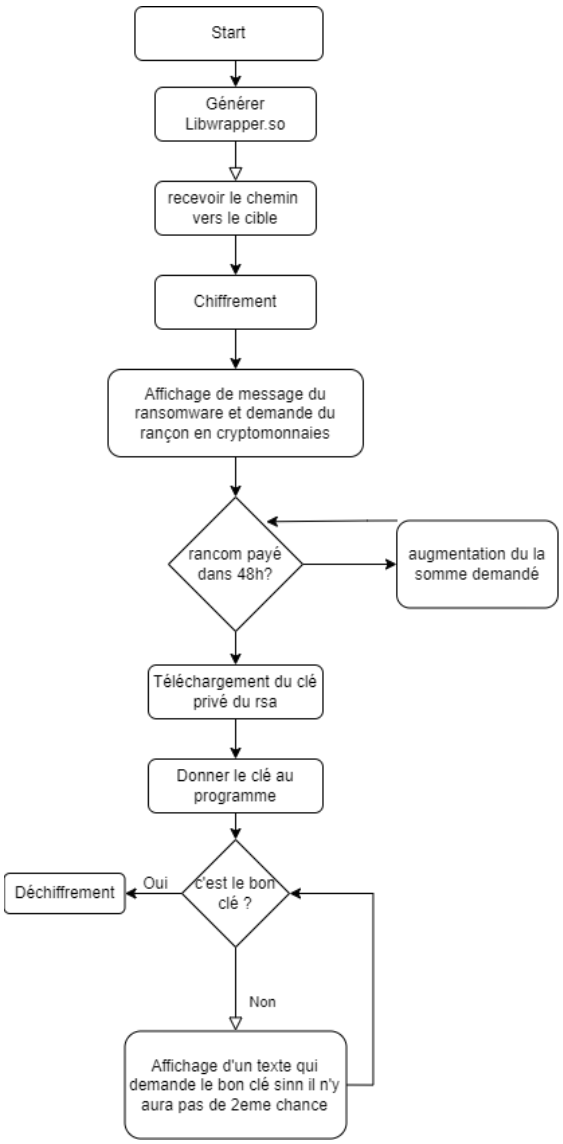

FIGURE 3.1 – Mécanisme de fonctionnement ransomware.

Chiffrement

Le chiffrement commence par le parcours des fichiers au sein du répertoire et l'obtention des chemins des éléments qu'il contient. Le programme vérifie si le chemin correspond à un exécutable, à un autre type de fichier ou à un répertoire. Dans le premier cas, le programme ajoute la bibliothèque partagée `libwrapper.so` comme dépendance au fichier exécutable. S'il s'agit d'un fichier ordinaire, le programme génère une nouvelle clé AES et un vecteur d'initialisation (VI) aléatoires, puis chiffre le fichier avec l'algorithme AES. La clé et le VI sont ensuite chiffrés avec la clé publique RSA et enregistrés dans le fichier `metadata.json`. Dans le dernier cas, si le chemin correspond à un répertoire, celui-ci est parcouru pour accéder aux fichiers qu'il contient.
La figure 3.2 représente l'organigrame de mécanisme de fonctionnement du chiffrement du ransomware.

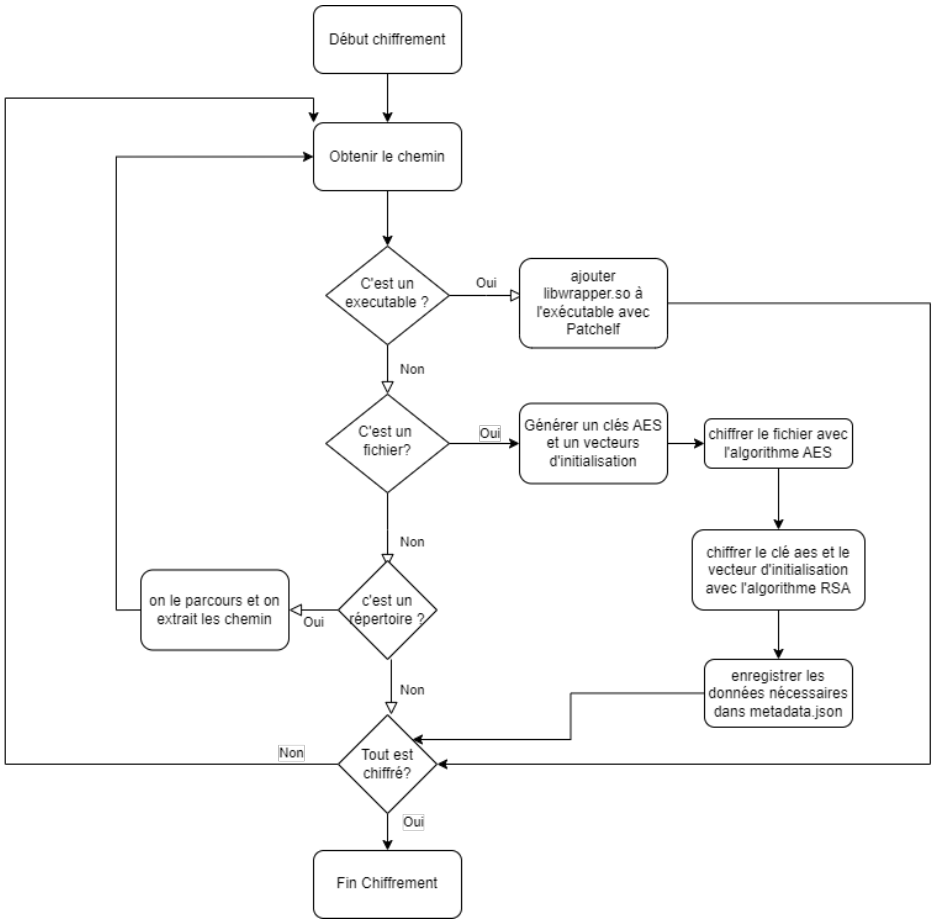

FIGURE 3.2 – Mécanisme de chiffrement.

Déchiffrement

Si la bonne clé privée est fournie au programme, le déchiffrement commence de la même manière que le chiffrement. Si un fichier rencontré est un exécutable, la dépendance à la bibliothèque partagée `libwrapper.so`. est supprimée. S'il s'agit d'un fichier ordinaire, on extrait la clé AES correspondante ainsi que l'IV, puis on les déchiffre à l'aide de la clé privée fournie. Ensuite, le fichier lui-même est déchiffré avec la clé déchiffrée. Le processus continue en vérifiant tous les répertoires jusqu'à ce qu'il n'y ait plus de fichiers à déchiffrer.
La figure 3.2 représente l'organigrame de mécanisme de fonctionnement du déchiffrement du ransomware.

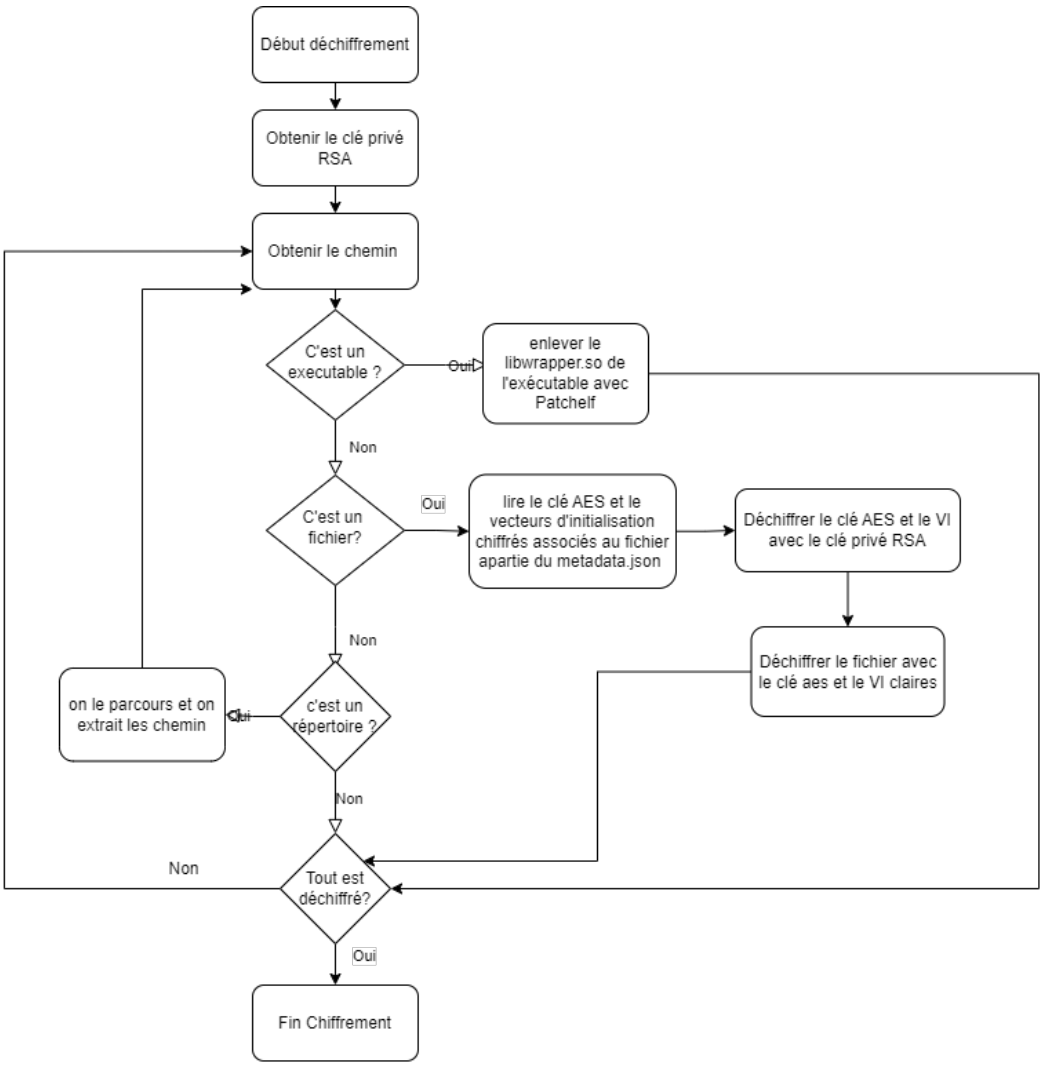

FIGURE 3.3 – Mécanisme de déchiffrement.

3.3 Développement de ransomware

3.3.1 Implémentation du Chiffrement

Le processus de chiffrement se compose de quatre étapes : le parcours des répertoires, le chiffrement en deux étapes des fichiers non exécutables, la modification du comportement des fichiers exécutables et l'enregistrement des clés AES chiffrées ainsi que des chemins des fichiers dans un fichier JSON.

Parcours de répertoire

La fonction de parcours explore récursivement tous les fichiers et sous-répertoires d'un répertoire donné pour appliquer un chiffrement ou une modification, en fonction du type de fichier. Elle commence par ouvrir le répertoire spécifié, puis parcourt chaque élément en ignorant le répertoire courant et le répertoire parent afin d'éviter les boucles infinies. Pour chaque fichier ou sous-répertoire trouvé, elle construit le chemin complet et récupère les informations du fichier. Si l'élément est un fichier exécutable, la fonction `patch` est appelée pour appliquer des modifications spécifiques. Si l'élément est un fichier régulier non exécutable, il est chiffré à l'aide de la fonction `encryptFile`. Dans le cas d'un sous-répertoire, la fonction appelle récursivement `encryptDirectory` pour traiter tout son contenu. Ce processus garantit que tous les éléments d'un répertoire, y compris les sous-répertoires, sont traités en fonction de leur type, assurant ainsi une couverture complète pour le chiffrement ou les modifications spécifiques, comme le montre la figure ci-dessous.

Chiffrement des clé AES

La fonction `rsa_encrypt` chiffre le clé et IV de AES fournies à l'aide de l'algorithme RSA et d'une clé publique. Elle ouvre le fichier contenant la clé publique, puis lit la clé au format PEM. Un contexte de chiffrement est créé et initialisé avec le mode de remplissage `RSA_PKCS1_OAEP_PADDING`. La fonction calcule ensuite la taille du message chiffré et effectue le chiffrement des données. Le résultat chiffré est stocké dans le tableau `encrypted`, et la fonction retourne la longueur des données chiffrées. En cas d'erreur à n'importe quelle étape, un gestionnaire d'erreur est appelé.

Chiffrement des fichiers

La fonction `encryptFile` permet de chiffrer le contenu d'un fichier en utilisant l'algorithme AES-256 en mode CFB (Cipher Feedback). Le processus commence par la génération aléatoire d'une clé AES de 256 bits et d'un vecteur d'initialisation (IV) à l'aide de la fonction `RAND_bytes`, qui sont essentiels pour garantir un chiffrement

sécurisé. Ensuite, le contenu du fichier est lu et stocké dans une variable `plaintext`, la fonction `readFile` retournant à la fois le contenu du fichier et sa taille. Le chiffrement proprement dit est effectué par la fonction `aes_encrypt`, qui utilise la bibliothèque OpenSSL pour appliquer l'algorithme AES-256 en mode CFB. Une fois le fichier chiffré, une nouvelle version du fichier est créée avec l'extension `.enc` pour stocker les données sécurisées. Les informations nécessaires au déchiffrement, telles que le chemin d'origine, la clé AES et l'IV, sont enregistrées dans un fichier de métadonnées afin de permettre un éventuel déchiffrement ultérieur. Enfin le fichier original non chiffré est supprimé après la création du fichier chiffré.

Sauvegarde des Métadonnées de Chiffrement

La fonction `save_metadata` permet de sauvegarder les métadonnées relatives à un fichier chiffré, incluant son chemin d'origine, son chemin chiffré, ainsi que la clé AES et le vecteur d'initialisation (IV) utilisés pour le chiffrement. D'abord, la clé AES et l'IV sont chiffrés à l'aide de RSA avec la clé publique (stockée dans `public.pem`), et leurs versions chiffrées sont stockées dans des tableaux. Ensuite, la fonction lit ou crée un fichier JSON (`metadata.json`) pour stocker ces informations. Un nouvel objet JSON est créé pour chaque fichier. Ces données sont ensuite ajoutées au fichier JSON sous un identifiant unique basé sur le chemin d'origine du fichier. Enfin, le fichier JSON est mis à jour avec ces nouvelles métadonnées, et la mémoire est libérée. Ce processus permet de maintenir un enregistrement des fichiers chiffrés et des informations nécessaires pour leur restauration ultérieure.

Modification du comportement des fichiers exécutables

La fonction `gen_libwrapper` génère un fichier source C (`wrapper.c`) qui crée une bibliothèque partagée permettant de modifier le comportement des fonctions `printf`, `puts`, et `strcmp` dans les exécutables. Elle écrit du code C dans le fichier pour redéfinir ces fonctions en utilisant `dlsym` pour récupérer leurs versions originales. La fonction `printf` et `puts` sont modifiées pour afficher un message personnalisé, et `strcmp` retourne un nombre aléatoire au lieu de comparer les chaînes. Une fois le code écrit, le fichier est compilé en une bibliothèque partagée (`libwrapper.so`) avec `gcc`, puis le fichier source est supprimé.

La fonction `patch` permet d'ajouter ou de retirer cette bibliothèque partagée (`libwrapper.so`) à un programme exécutable via l'outil `patchelf`. Selon la valeur du paramètre `i`, elle construit une commande pour ajouter ou retirer la bibliothèque du programme cible, puis l'exécute.

3.3.2 Implémentation du Déchiffrement

Cette section explique le processus de déchiffrement des fichiers chiffrés par AES. En utilisant la clé privée RSA fournie, les métadonnées (contenant la clé AES et l'IV) sont récupérées et déchiffrées. Ces éléments sont ensuite utilisés pour restaurer les fichiers à leur état d'origine. Enfin, les bibliothèques modifiant le comportement des exécutables sont supprimées pour revenir à la configuration initiale.

Parcours de répertoire chiffré

La fonction `decryptDirectory` parcourt récursivement un répertoire donné, décryptant les fichiers chiffrés et supprimant les modifications apportées aux exécutables. Elle commence par ouvrir le répertoire spécifié. Pour chaque fichier ou sous-répertoire rencontré, elle vérifie s'il s'agit d'un fichier exécutable. Si c'est le cas, elle retire les modifications comportementales via la fonction `patch` en passant le paramètre `-1`. Si le fichier est un fichier chiffré (identifié par l'extension `.enc`), elle appelle la fonction `load_metadata` pour récupérer la clé privée RSA et déchiffrer le fichier. Si un sous-répertoire est rencontré, la fonction s'appelle récursivement pour traiter les fichiers dans ce sous-répertoire. Enfin, le répertoire est fermé une fois toutes les opérations terminées.

Déchiffrement des Fichiers

La fonction `load_metadata` lit un fichier JSON contenant les métadonnées des fichiers chiffrés, récupère la clé privée RSA et l'IV nécessaires pour le déchiffrement, et effectue le processus de déchiffrement des fichiers. Elle commence par ouvrir et lire le fichier `metadata.json`. Pour chaque fichier listé dans les métadonnées, elle extrait les chemins du fichier original, du fichier chiffré, ainsi que la clé AES et l'IV qui ont été chiffrés par RSA. Ces éléments sont ensuite déchiffrés à l'aide de la clé privée RSA fournie en paramètre.

Une fois que la clé AES et l'IV sont récupérés, le fichier chiffré est ouvert, et le contenu est déchiffré avec AES à l'aide de la clé et de l'IV. Si le déchiffrement réussit, le contenu déchiffré est sauvegardé dans le fichier original, et le fichier chiffré est supprimé. La fonction continue ainsi pour chaque fichier spécifié dans les métadonnées. Enfin, le fichier JSON contenant les métadonnées est fermé.

3.3.3 Implémentation d'un Système de Paiement de la Rançon

Cette section décrit l'implémentation d'un mécanisme de paiement sécurisé, utilisé pour le paiement de rançons. L'objectif principal de ce système est de

permettre aux victimes de récupérer leurs fichiers en payant une rançon via une transaction Ethereum. Le système se compose de deux parties distinctes : un frontend interactif et un backend en PHP qui gère les transactions et la génération du lien pour la clé de déchiffrement..

Frontend

Le frontend de l'application se compose d'une interface web qui permet à l'utilisateur d'interagir facilement avec le système. Lorsqu'une victime de ransomware souhaite récupérer ses fichiers, elle est invitée à effectuer une transaction Ethereum vers un compte spécifique contrôlé par l'attaquant. Cette transaction est facilitée par l'utilisation du portefeuille Ethereum MetaMask, qui permet à l'utilisateur de gérer ses fonds Ethereum directement depuis son navigateur.

Lorsque l'utilisateur clique sur le bouton "Envoyer 1 ETH", une demande est envoyée à MetaMask pour autoriser la transaction. Les informations essentielles, telles que le montant à envoyer et l'adresse du destinataire, sont prédéfinies dans le code, simplifiant ainsi l'interface pour l'utilisateur. MetaMask prend en charge la signature de la transaction et l'envoie ensuite sur la blockchain Ethereum. Cependant, pour garantir un environnement sécurisé et sans risque financier, la transaction est effectuée sur le réseau de test Sopelia, un réseau de test Ethereum conçu spécifiquement pour simuler des transactions sans utiliser de fonds réels. Cela permet de tester et valider l'ensemble du processus sans aucune implication financière réelle.

Backend

Une fois la transaction envoyée, un hash de celle-ci est généré et transmis au serveur backend via une requête HTTP. Ce hash permet de vérifier que la transaction a bien été enregistrée sur la blockchain, garantissant ainsi son intégrité et sa confirmation.

Dès que le backend reçoit la confirmation que la transaction a été validée avec succès, il envoie un lien de téléchargement vers un fichier contenant la clé de déchiffrement. Cette clé permet à la victime de récupérer ses fichiers après avoir effectué le paiement de la rançon.

Chapitre 4

Scénario du ransomware

Introduction

Dans ce chapitre, nous allons explorer en détail le déroulement d'une attaque par ransomware. Nous examinerons les différentes étapes de l'attaque, depuis le phishing jusqu'à la le paiement de rançon, en passant par les techniques de propagation et de chiffrement et de déchiffrement des données.

4.1 L'arborescence repertoire cible

Dans ce scénario, le répertoire `Folder` est la cible du ransomware, où nous allons chiffrer plusieurs fichiers : `Password.txt`, `user.txt`, `parcours.c`, l'image `enf.jpg` et l'exécutable `hello`, qui se trouvent à différents niveaux dans cette arborescence..

La figure 4.1 montre l'emplacement des fichiers cibles.

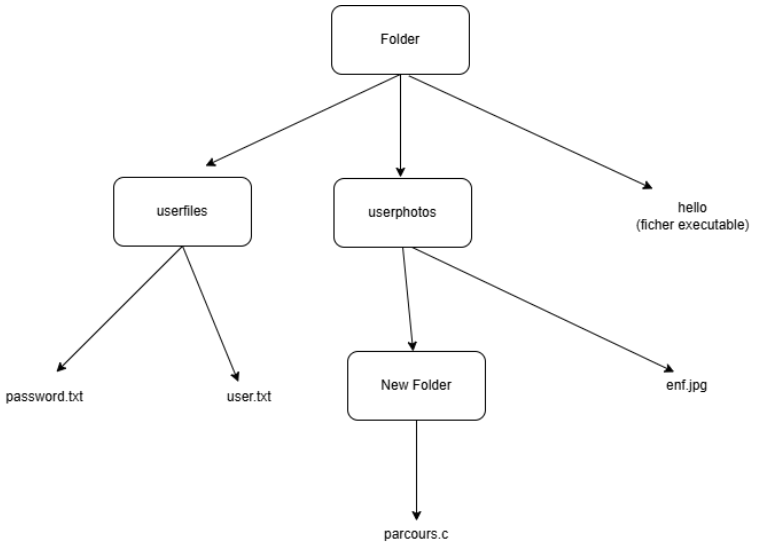

FIGURE 4.1 – Représentation de la réparation des fichiers.

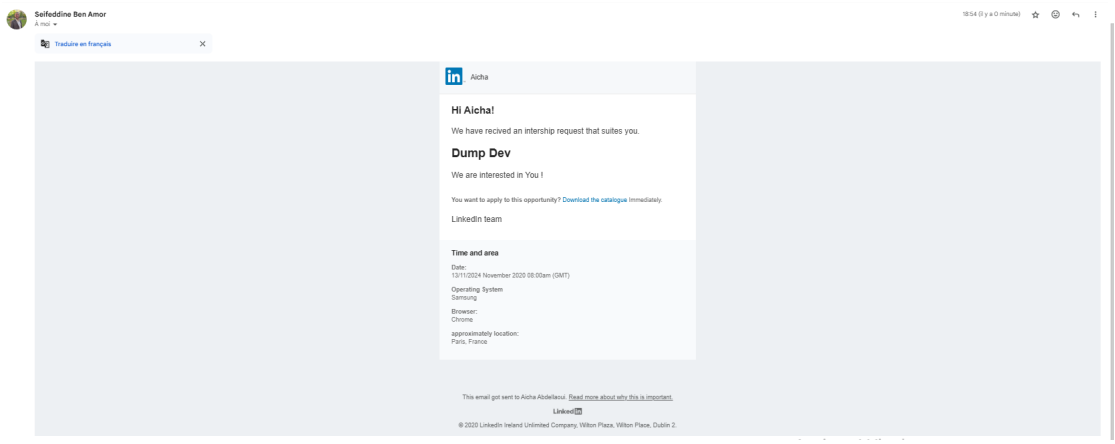

FIGURE 4.2 – Enter Caption

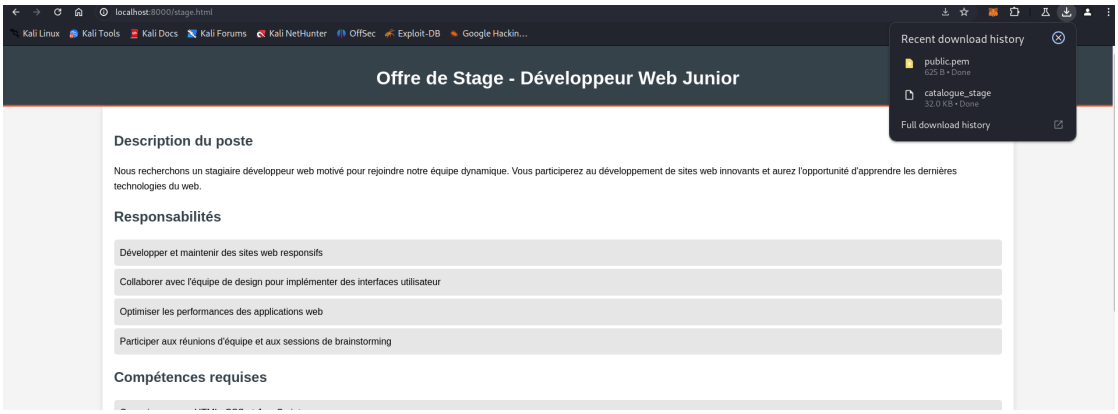

FIGURE 4.3 – Enter Caption

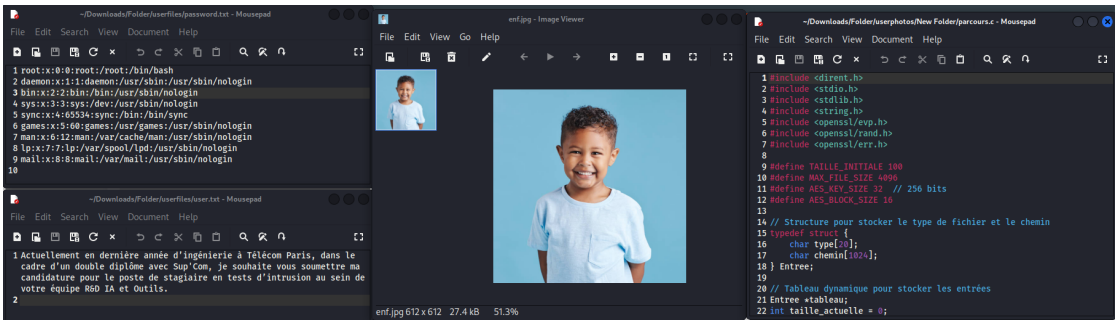

FIGURE 4.4 – Enter Caption

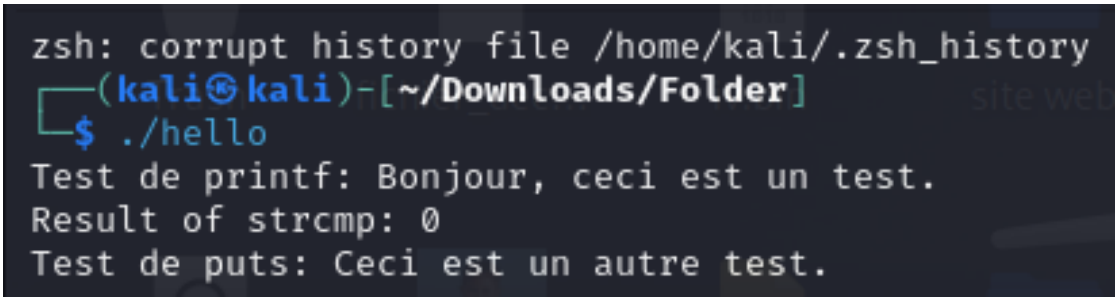

FIGURE 4.5 – Enter Caption

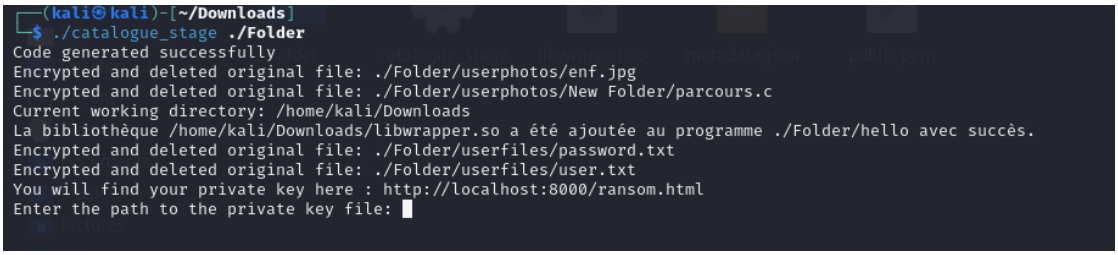

FIGURE 4.6 – Enter Caption

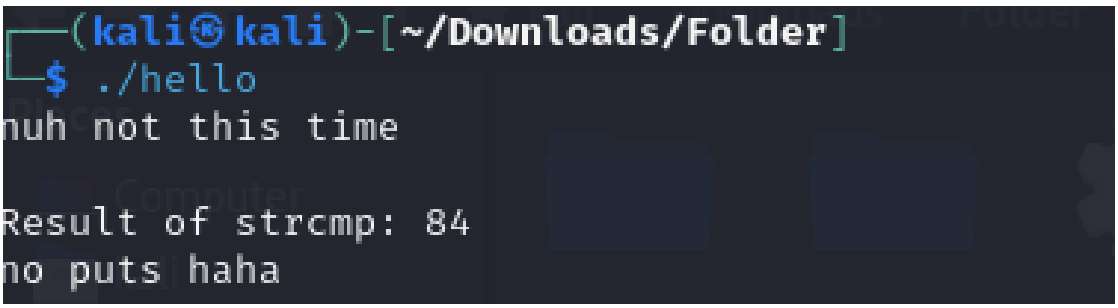

FIGURE 4.7 – Enter Caption

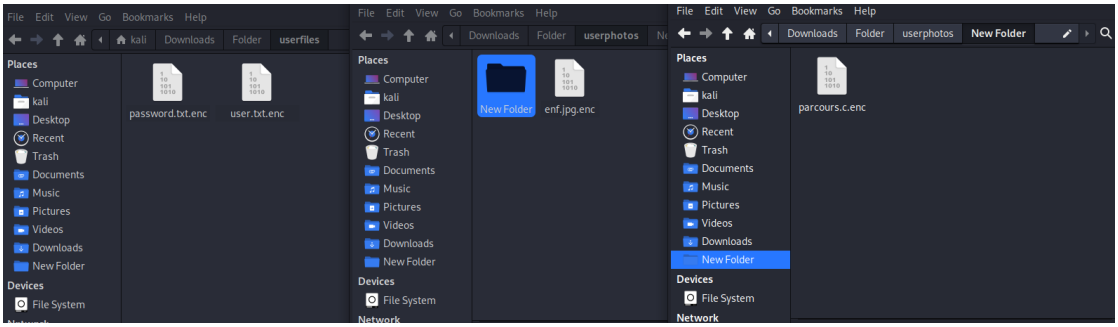

FIGURE 4.8 – Enter Caption

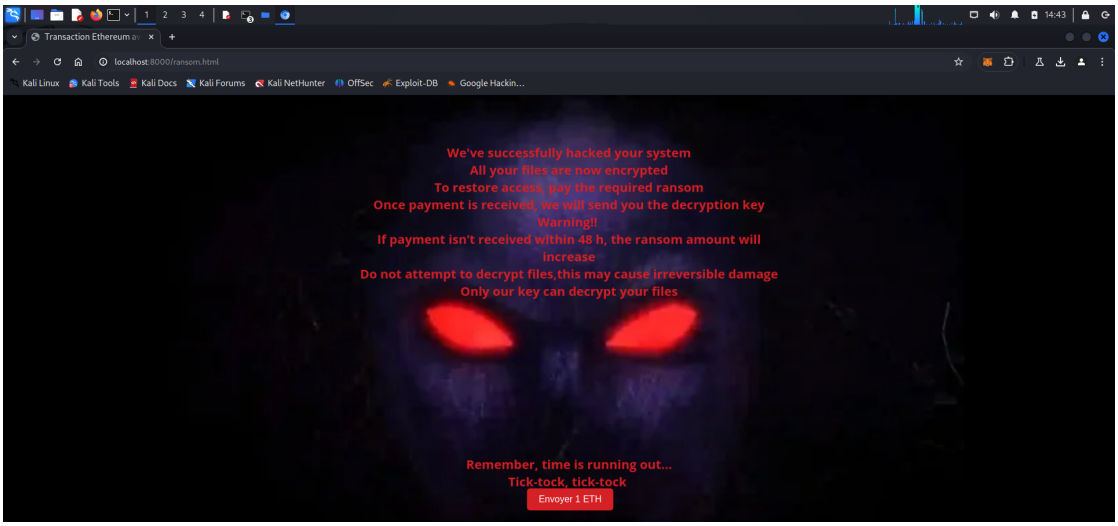

FIGURE 4.9 – Enter Caption

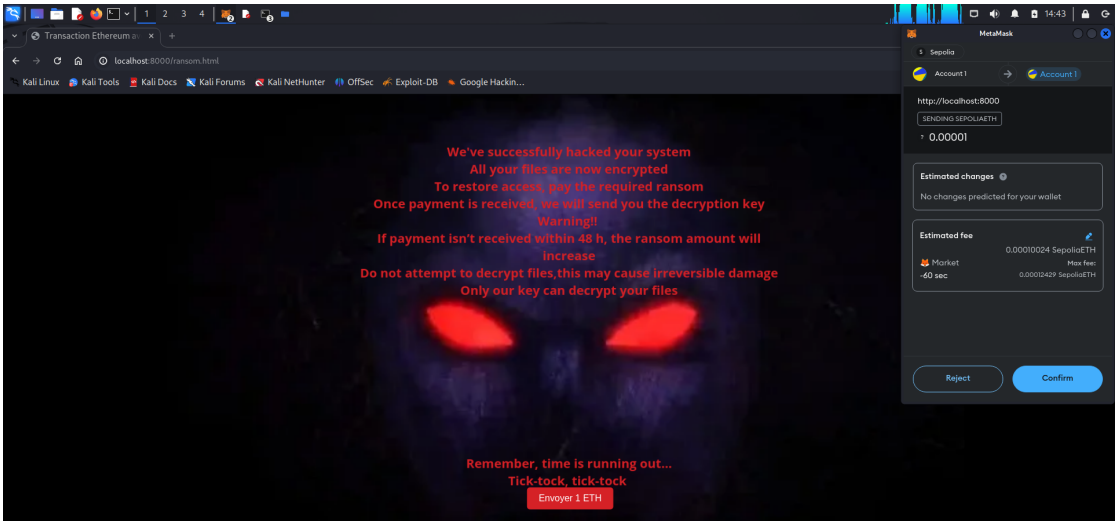

FIGURE 4.10 – Enter Caption

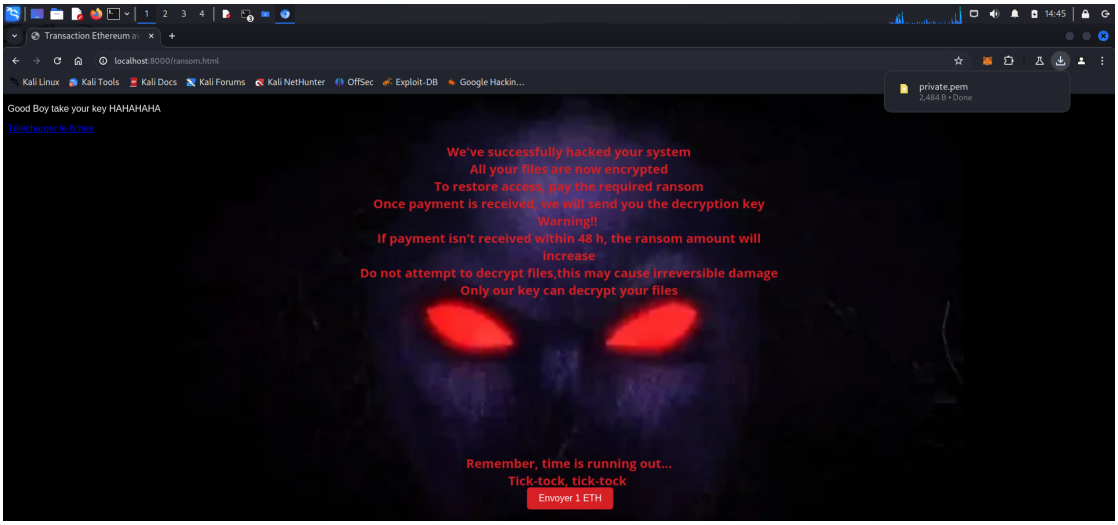

FIGURE 4.11 – Enter Caption

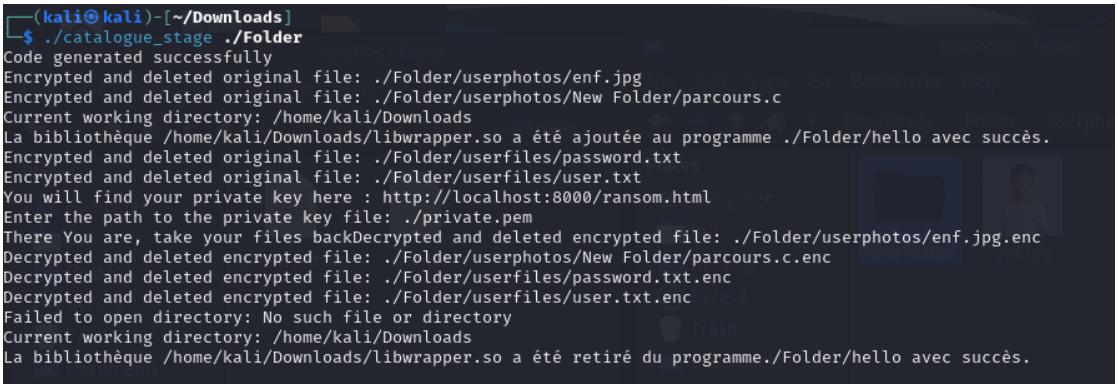

FIGURE 4.12 – Enter Caption

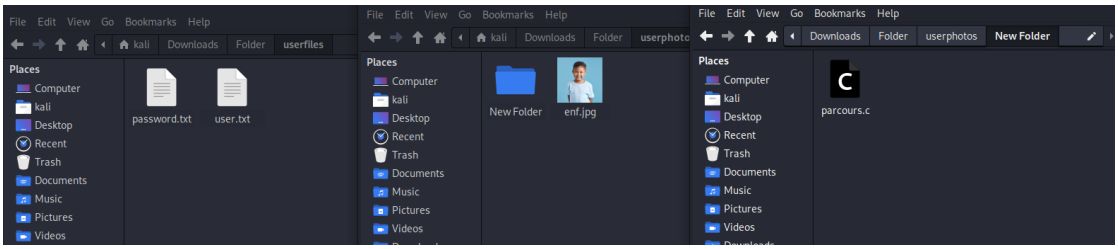

FIGURE 4.13 – Enter Caption

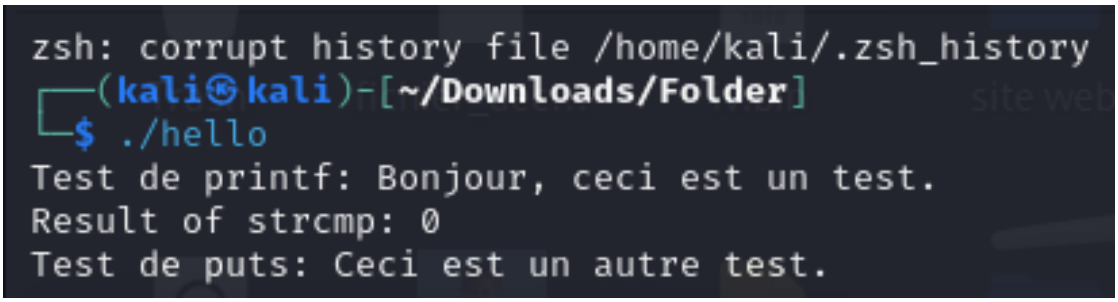

FIGURE 4.14 – Enter Caption

Annexe

En scannant le QR code 4.15, Vous trouverez l'implimentation du Ransomware.

FIGURE 4.15 – Lien vers le travaille réalisé