

DE LA RECHERCHE À L'INDUSTRIE



Extraction et Analyse de malware

Du forensics au reverse

Commissariat à l'Énergie Atomique et
aux Énergies Alternatives |

18 décembre 2019

- 1** « Ecosystème » des malwares
 - Qui sont les attaquants ?
 - Définitions
 - Exploitation As A Service : les Exploits Kits
 - Les défenseurs
 - Monétisation

2 Forensics : extraction des malwares

- Objectifs
- Forensics réseau
- Forensics système
- Forensics applicatif

- 3 Analyse des malwares**
 - Généralités de l'analyse
 - Outils d'automatisation
 - Analyse « à la main »
 - Packers
 - Obscurcissement

N'hésitez pas à poser vos questions entre 2 slides

S'il n'y a pas de questions, autant regarder une vidéo :)

1 « Ecosystème » des malwares

- Qui sont les attaquants ?
 - Définitions
 - Exploitation As A Service : les Exploits Kits
 - Les défenseurs
 - Monétisation

Attaquants

Plusieurs types d'attaquants :

- 1 un hacker (dans son garage)
- 2 une mafia russe (dans sa vodka)
- 3 un état

Le cybermonde de la cybercriminalité

- les chercheurs de vulnérabilités → 0 day
- les auteurs d'exploits → *weaponisation*
- les auteurs de banker, miner, RAT, ... → charge utile
- les auteurs de packers → contournement des détections, retard dans l'analyse
- les auteurs d'obfuscateurs JS/HTML/Flash/... → idem
- les auteurs d'Exploit Kit → Exploitation As A Service
- les hosters "Bullet Proof" → réclamations > /dev/null
- les packagers → Platform As A Service
- les exploitants → Cartes bancaires As A Service

Démonstration

Phishing générique

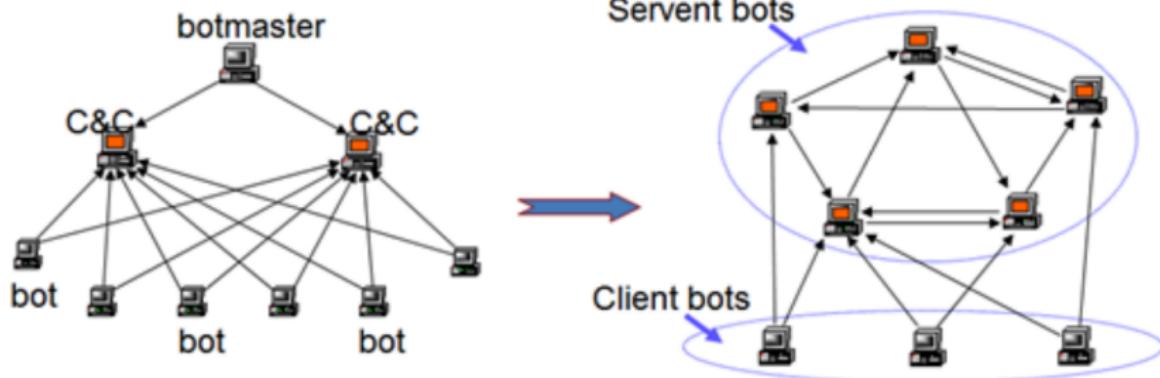
1 « Ecosystème » des malwares

- Qui sont les attaquants ?
- **Définitions**
- Exploitation As A Service : les Exploits Kits
- Les défenseurs
- Monétisation

Quelques définitions

- Malware
- Spyware
- Virus, Ver, Trojan, Rogue
- APT (*Advanced Persistent Threat*)
- SPT (*Simple Persistent Threat*)
- Cybercriminalité
- Anti-Virus - signatures et heuristiques
- RAT (*Remote Administration Tool*)
- Banker
- Miner
- Botnets
- *Drive by download*

Architecture de Botnets



From centralized botnet to hybrid peer-to-peer botnet

Démonstration

RAT : DarkComet

- 1** « Ecosystème » des malwares
 - Qui sont les attaquants ?
 - Définitions
 - Exploitation As A Service : les Exploits Kits**
 - Les défenseurs
 - Monétisation

Les composantes d'un Exploit Kit

- 1 Filets : site « pwnés » pour rediriger vers la *landing page*
- 2 Landing page : analyse du client, redirection vers les exploits suivant les versions des plugins détectées
- 3 Exploits : exploitation d'une vulnérabilité pour injecter un shellcode qui va télécharger le *payload*
- 4 Payload : banker, miner, RAT, ...

Exemples d'Exploit Kit

- NuclearPack
- SweetOrange EK
- Neutrino EK
- Goon EK
- Fiesta EK
- ...

Éléments ciblés

- Navigateurs
- Plugins : PDF, Silverlight, Flash, Java, ...
- Naïveté de l'utilisateur

Démonstration

FinFisher

Démonstration

XyliBox

Point de vue réseau

```
11:13:02    text/javascript    http://sitexemple.com/jquery.js
Mozilla/5.0 (X11; Linux i586; rv:31.0) Gecko/20100101 Firefox/31.0
11:13:02    text/html      http://aze.rty.org/17856875623/head.html
Mozilla/5.0 (X11; Linux i586; rv:31.0) Gecko/20100101 Firefox/31.0
11:13:04    application/octet-stream  http://aze.rty.org/17856875623/765120987.jar
Java/1.6.0_24
11:13:06    application/octet-stream  http://aze.rty.org/wp-content.php?clid=98756
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 7.1; Trident/5.0)
```

1 « Ecosystème » des malwares

- Qui sont les attaquants ?
- Définitions
- Exploitation As A Service : les Exploits Kits
- Les défenseurs**
- Monétisation

Les entités

- ANSSI : Agence Nationale de la Sécurité des Systèmes d'Information
- CERT : *Computer Emergency Response Team*, aussi appelés CSIRT
- → CERT-FR (ex-CERTA)
- SOC : *Security Operation Center*

La prévention

- Recherche de vulnérabilités (analyse de code source, fuzzing, détection automatique, ...)
- Systèmes maintenus à jour
- Audits (infrastructure, organisationnel, logiciels, matériels)
- Pentests
- Certifications
- ...

La mitigation

- IDS (*Intrusion Detection System*)
 - NIDS (*Network*)
 - HIDS (*Host-based*)
- IPS (*Prevention*) → IDS actif
- Politiques de sécurité
- Pare-feu
- Antivirus
- ...

La réaction

- Qui ?
 - Provenance des connexions
 - Eléments d'identifications
 - "Preuves"
 - Analyse réseau
 - Mobile
 - Est-ce réellement la Corée du Nord ?
- Quoi ?
 - Analyse binaire
 - Analyse système
 - Quelle est / quelles sont les charges utiles ?
 - Analyse in-vivo
 - Analyse post-mortem

La réaction

- Comment ?

- Chemins d'infection
- Elevations de priviléges
- Vulnérabilités 0-day, 1-day, ...

- Pour quoi ?

- Vol de données métier
- Rebond vers une autre cible
- Manipulation, Rançon, Chantage
- Discrédition

La réaction

- Quelle réaction ?
 - Nettoyage de printemps
 - Aides extérieures
 - Armes juridiques
 - Contre-attaque
 - Divulgation dans la presse, au sein de l'entreprise
 - KDC compromis ? Il faut «basculer»...

1 « Ecosystème » des malwares

- Qui sont les attaquants ?
- Définitions
- Exploitation As A Service : les Exploits Kits
- Les défenseurs
- Monétisation

Prix de revente

Botnets par pays, pour 1000 machines infectées

- Canada : \$270
- France : \$200
- Russia : \$200
- United Kingdom : \$240
- United States : \$180
- Worldwide : \$35

Cartes de crédit

- Premium Card, Big Balance : \$250
- Credit Card + Social Security Number : \$5

Prix de revente

Doxxing

→ Recherche de renseignements sur une personne

- De \$25 à \$100

1000 adresses e-mails

- Gmail : \$200
- Hotmail : \$12
- Yahoo : \$10

Prix de revente

Réseaux sociaux

- 1000 “Likes” sur Facebook : \$15
- Page avec 30 000 fans : \$13
- 10 000 faux “followers” twitter : \$15

En ligne

- 1000 faux visiteurs : \$1
- Attaques de jeux en ligne chinois : \$16 000 / mois

Prix de revente

Divers

- Webcam (Homme) : \$0.01
- Webcam (Femme) : \$1
- Assurance vie : \$1200 à \$1300
- Cours de "Hacking" : \$75
- Achat d'un RAT : \$40
- Tueur à gage (France) : \$330 000
- Tueur à gage (Philippines) : \$110

Source : Havocscope

2 Forensics : extraction des malwares

■ Objectifs

- Forensics réseau
- Forensics système
- Forensics applicatif

- Les techniques de *computer forensics*, regroupée en français sous le terme **d'informatique légale**, recouvrent plusieurs aspects de la gestion d'une intrusion informatique. La discipline a été nommée en rapport avec la médecine légale.
- On parle aussi d'investigation numérique ou d'autopsie.
- On utilise plus couramment le terme de *forensics* pour désigner les analyses faites suite à la compromission d'un système informatique.

Les objectifs sont :

- Analyse du système « live » : relever un maximum d'indice sur l'activité du système compromis pendant que le malware ou l'attaquant est actif ;
- Analyse du système « offline » : diagnostiquer a posteriori le déroulement de la compromission à l'aide de traces prélevées sur les systèmes compromis ou dans leur environnement.

Domaines de forensics

■ Forensics réseau

- Etude des traces collectées sur les réseaux informatiques, ce domaine cherche à retrouver des activités malicieuses dans des captures complètes de communications ou des données synthétiques (par exemple des données de flux);

■ Forensics système

- Etude des traces collectées sur l'activité d'un système d'exploitation ou sur les mémoires accessibles par ce système (mémoire vive, mémoire de masse type disque dur);

■ Forensics applicatif

- Etude des traces collectées au niveau des logs d'applications, de passerelles protocolaires (serveurs proxy) ou enfin analyse de documents transitant sur le réseau.

Le problème de la confiance

L'investigation numérique d'un système compromis pose différents problèmes de confiance (au sens informatique) :

- Confiance dans le système compromis : on ne sait pas a priori jusqu'où l'attaquant a compromis le système. Il peut aussi bien avoir simplement volé des données d'un utilisateur qu'avoir remplacé le noyau du système ou modifié les firmwares de l'ordinateur ;
- Confiance dans les outils utilisés : il est important d'avoir une bonne maîtrise des outils pour savoir quels indices ils peuvent extraire, ou au contraire savoir comment ils peuvent être trompés ;
- Confiance dans l'investigateur : en particulier si c'est un investigateur externe, il faut être sûr de sa compétence et de son intégrité (il peut compromettre les preuves ou ajouter des éléments incriminants qu'il a lui-même créés).

Reflections on trusting trust (Thompson 1984)

- Réflexion sur la dissimulation d'une backdoor dans un système
- L'attaquant veut ajouter une backdoor dans /bin/login (accès root caché) ;
- Il modifie les sources de login, le fichier login.c ;
- Tout le monde peut lire ce fichier et voir la backdoor, alors il modifie le compilateur pour qu'il ajoute la backdoor dans login.c au moment de la compilation ;
- Tout le monde peut lire les sources du compilateur, alors l'attaquant modifie le compilateur utilisé pour compiler le compilateur ;
- Conclusion : on fait toujours confiance à du logiciel compilé qui peut avoir été compromis, à moins d'écrire son propre système d'exploitation et compilateur en assembleur.

La réflexion sur la confiance nous montre qu'il est nécessaire de mettre en place un environnement d'analyse dans lequel on peut avoir une confiance maximale :

- Environnement d'analyse déconnecté des autres réseaux ;
- Machines jetables, en prenant en compte le fait que les moteurs de machines virtuelles peuvent être eux-mêmes compromis ;
- Systèmes d'acquisition de données avec une garantie de préservation des preuves (par exemple bloqueurs en écriture pour lire les disques durs de machines compromises) ;
- Limiter au maximum les analyses faites directement sur le système compromis (toute action d'analyse peut effacer des traces de l'attaque) ;
- Calculer des empreintes numériques, voire signer numériquement toutes les preuves.

2 Forensics : extraction des malwares

- Objectifs
- **Forensics réseau**
- Forensics système
- Forensics applicatif

L'investigation au niveau réseau vise à identifier les traces de la compromission dans les captures de communications.

Sources de données :

- Données de type paquets (typiquement pcap)
 - Capture exhaustive des communications réseau ;
 - Nécessite un grand espace de stockage ;
 - Utilisé pour extraire les données brutes correspondant à une connexion ;
 - Exemples d'outils : tcpdump, tshark, wireshark ou encore scapy.
- Données de type flux (par exemple NetFlow ou Argus)
 - Contient des statistiques sur les nombres de paquets et quantités d'octets échangés par connexion TCP ou échange UDP ;
 - Ne contient pas les données brutes ;
 - Utilisé pour défricher et repérer rapidement les flux intéressants, avant d'aller chercher dans des captures complètes ;
 - Exemples d'outils : ra pour le format argus, ou nfdump pour le format NetFlow ;

Démarche d'analyse :

- Repérage des connexions par adresse IP source ou destination dans les flux, avec éventuellement un raffinage par port source ;
- A partir des *timestamps*, sélection d'un fichier de capture de données pour obtenir les données brutes ;
- Enfin extraction des données brutes avec tshark ou scapy ;

Astuces :

- Les adresses IP intéressantes sont généralement celles des machines compromises, l'attaquant peut utiliser un botnet et donc attaquer depuis de multiples adresses ;
- Si on peut lier l'attaque à une adresse IP source alors c'est un bon critère ;
- Les ports source sont utilisés pour différencier les connexions liées à l'attaque ;
- Penser à corrélérer ces informations avec celles de l'analyse du système.

2 Forensics : extraction des malwares

- Objectifs
- Forensics réseau
- Forensics système**
- Forensics applicatif

Cette étape vise à mettre en évidence la présence d'un attaquant ou d'un malware sur un système compromis en cours de fonctionnement, en prenant soin de ne pas écraser les traces de l'activité malicieuse.

Démarche d'analyse :

- Sauvegarder toutes les traces sur un media indépendant au fur et à mesure de l'investigation (clé USB, réseau) ;
- Identifier le système, obtenir un maximum d'informations sur la configuration (réseau, services activés, fichiers ou base de configuration) ;
- Mettre en évidence l'activité de l'attaquant, obtenir la liste des processus, des connexions établies ou des ports en écoute, sauvegarder les logs à chaud ;
- Effectuer des dumps en prévision de l'analyse offline, en particulier dump de la mémoire vive (si possible) de façon logicielle ou matérielle (dump via FireWire).

Démonstration

Analyse d'un dump de RAM avec Volatility

Forensics système : analyse « offline »

L'étape de l'analyse offline a pour objectif de faire une liste la plus exhaustive possible des actions qui ont été faites par l'attaquant sur le système.

Démarche :

- Collecte des informations brutes : dump de la mémoire vive, copie bit à bit des disques durs (pour avoir des copies de travail), si les logs système sont collectés sur un réseau séparé alors également obtenir les logs liés à la machine ;
- Reconstitution d'une *timeline* de l'activité du système compromis ;
- Recherche de fichiers effacés, modifiés ou cachés : soit par analyse du système de fichiers, soit par analyse brute du disque (*file carving*) ;
- Extraction des fichiers suspects pour analyse ultérieure.

Démonstration

Analyse d'un disque dur avec Autopsy

Démonstration

Analyse d'un disque dur avec DFF

Dans le cas de Linux, voici quelques pistes pour identifier la présence d'un rootkit dissimulé par l'attaquant :

- Détection de programmes modifiés : avec un outil de test d'intégrité, on compare les empreintes de référence avec les fichiers actuellement présents, cette action est faite de préférence en offline ;
- Détection de bibliothèques modifiées : idem avec outil de test d'intégrité, ou présence de variables PRELOAD dans les environnements des processus ;
- Détection de module noyau illicite : listing des modules noyau en live (commande `lsmod`) ou liste des modules présents sur le disque dur en offline ;
- Détection de détournement d'appels système : extraction des premiers octets et recherche de la présence d'instructions assembleur de type *jump* ;
- Détection de la modification de Grub ou des parties BIOS / UEFI : difficile, dépend de la distribution et du matériel.

Liste d'outils pour l'analyse offline d'un système :

Outils commerciaux, parfois reconnus par des tribunaux :

- Encase, FTK ;
- Orientés recherche de preuve, pas forcément adaptés pour reconstituer l'activité d'un attaquant ;

Outils Open Source

- le générique Sleuth Kit avec son interface web Autopsy ;
- les outils spécialisés en file carving Testdisk et Photorec ;
- l'outil graphique Digital Forensics Framework ;
- les distributions spécialisées comme Kali Linux ;
- Ces outils demandent généralement une certaine expertise.

2 Forensics : extraction des malwares

- Objectifs
- Forensics réseau
- Forensics système
- **Forensics applicatif**

Le forensics applicatif a pour objectif de mettre en évidence les traces d'une compromission dans les logs des applications ou des serveurs proxy, ainsi que d'extraire les fichiers contenant des malwares qui transitent sur les réseaux.

Démarche :

- Collecte des logs des applications déployées sur les stations de travail, en particulier des applications habituellement ciblées par les malwares comme les navigateurs web et les suites bureautiques ;
- Collecte des logs sur les serveurs proxy, en particulier proxy HTTP et mail ;
- Collecte des fichiers échangés à l'aide d'outils de détection d'intrusion comme Bro, ou des outils plus spécifiques comme mailsnarf ou urlsnarf ;

Analyse des traces collectées :

- Corrélation entre logs de serveurs proxy et listes de domaines connus pour héberger du malware, comme malwaredomainlist ou la base de Google Safe Browsing ;
- Recherche de *pattern* de logs de connexions correspondant à une redirection d'un navigateur vers un site hébergeant des malware (landing site) ;
- Recherche de *pattern* correspondant à une communication entre un système compromis et le C&C d'un botnet ;
- Analyse des fichiers extraits des communications réseau ;
- Analyse des logs des applications, ce travail étant probablement le plus fastidieux.

3 Analyse des malwares

- Généralités de l'analyse
 - Outils d'automatisation
 - Analyse « à la main »
 - Packers
 - Obscurcissement

Objectifs principaux de l'analyse

- Que fait le malware ?
- Création d'IoC (*Indicator of Compromise*)

Objectifs secondaires

- Attaque ciblée ?
- Appartenance à une campagne
- Appartenance à une famille connue
- Exploitation de vulnérabilités inconnues
- Niveau de l'attaquant (== niveau perçu du défenseur)
- Création d'outils automatisés (signature, unpacking, ...)

Protection contre l'analyse

- Détection d'un environnement virtualisé
 - Nom des interfaces et partages réseaux
 - Modèle du disque dur
 - *Hook* de fonctions systèmes
 - Faible taille de disque dur
 - Souris immobile
 - → cours sur la virtualisation !
- Masquage de la charge utile
 - Compression
 - Chiffrement
 - Obscurcissement
 - → packers !

Démonstration

Pafish

Protection contre l'analyse

- “Anti-debug”
 - IsDebuggerPresent
 - DebuggerMessage
 - Erreurs d'implémentation des debuggers
 - États indéterminés
 - Noms des fenêtres
 - Détection des INT 3
- Détection de l'environnement
 - Antivirus présents
 - Programmes d'analyse présents
 - Nom de l'utilisateur, de la machine
 - IP extérieure connue

Démonstration

Fonctions appelées dans un *payload* malveillant

Protection contre l'analyse

- Anti-émulation
 - Boucles d'attente
 - Boucles d'appels
 - Fonctions inconnues ou très peu documentées

Features

- Porte d'entrée
 - « Download & Execute »
 - Intégration d'outils (montage d'un disque dur virtuel)
- Reconnaissance
 - Mapping réseau
 - Observation de l'activité
 - Échantillonnage

Que fait le malware ?

Features

■ Fonctionnalités passives

- Vol des données de navigation
- Vol des mots de passe stockés (FTP, Mail, Steam, ...)
- Keylogger
- Vol de fichiers

■ Fonctionnalités actives

- Injection de script dans les pages webs
- Mécanismes de propagation (envoi de mail, exploitation de vulnérabilités, ...)
- Bitcoin

Démonstration

Chaînes de caractères dans NeverQuest

Type d'IoC

- Clés de registre
- MD5 de fichier
- FuzzyHash de malware
- IPs, noms de domaine de C&C

Utilisation des IoCs

- Recherche de l'étendue de la compromission (temps et nombre de postes touchés)
- Ajout de signatures dans les IDS, IPS, Anti-virus, ...
- Rapprochement avec les autres campagnes
- Échange avec les partenaires

3 Analyse des malwares

- Généralités de l'analyse
- **Outils d'automatisation**
- Analyse « à la main »
- Packers
- Obscurcissement

Pourquoi automatiser ?

- Nombreux *samples* à analyser
- Pour une même campagne, analyses potentiellement similaires
- Faible niveau technique requis
- Facilement parallélisable
- Business model

Mais ...

- Nécessité d'identifier la cible pour les analyses spécifiques
- Mise au point des analyses spécifiques
- Protections contre les analyses automatiques
- Tout ne peut pas être fait automatiquement!
- /!\ Est-on sûr de ne rien rater? (*Flame*)

Anti-virus

Pour pallier aux manques de signatures des AV → plusieurs AVs

- VirusTotal
- IRMA

Analyse comportementale

La méthode est la suivante :

- 1 Restauration d'une machine (le plus souvent virtuelle via un snapshot)
- 2 Ajout et lancement de l'élément ciblé (.doc, .exe, URL, ...)
- 3 Instrumentation de la machine
 - Fonctions systèmes appelées
 - Fichiers déposés sur le disque
 - Interactions réseau
- 4 Crédit et récupération d'un rapport

Analyse comportementale

- Cuckoo Sandbox
- Malwr .com -> en cours de réouverture
- Hybrid Analysis (Crowdstrike)
- ANY.RUN

Détection de signature

- Yara
- Snort
- LordPE

Les outils du commerce intègre les différents éléments dans leur interface

Démonstration

Hybrid Analysis

Lancements automatiques

L'automatisation permet aussi de lancer des tâches en permanence.

- Sur les PJ d'un e-mail lors de sa réception
- Sur l'ensemble des binaires d'un parc
- Sur une passerelle USB

Toolchain

Lancement d'une suite d'outils « maison » ou commerciaux, jusqu'à l'obtention d'un point fixe

- REbus

3 Analyse des malwares

- Généralités de l'analyse
- Outils d'automatisation
- **Analyse « à la main »**
- Packers
- Obscurcissement

Intérêts et inconvénients

Intérêts

- Permet d'être complet dans l'analyse
- Découverte de nouveaux éléments
- Des tâches ne peuvent pas être automatisées

Inconvénients

- Prend (beaucoup) de temps
- Résultats fortement liés aux compétences de l'équipe d'analyste

Extraction des documents (pdf, doc, ...)

Les documents ne sont pas directement exécutables. Il faut donc s'intéresser aux macros et éléments externes inclus !

Extraction de ces éléments :

- PDFToolkit
- JSUnpack
- Origami
- OfficeMalScanner
- Unzip

Stratégie

- 1 Utiliser des outils automatiques
- 2 Extraire les éléments exécutables
- 3 Les désobscurer si besoin
- 4 Analyser le code (souvent très simpliste)

Exemple de macro Word

```
Sub Run_Cmd(command, visibility, wait_on_execute)
Dim WshShell As Variant
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run "%COMSPEC% /c " & command, visibility, wait_on_execute
End Sub

Sub Run_Program(program, arguments, visibility, wait_on_execute)
Dim WshShell As Variant
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run program & " " & arguments & " ", visibility, wait_on_execute
End Sub

Sub Workbook_Open()
Const VISIBLE = 1, INVISIBLE = 0
Const WAIT = True, NOWAIT = False
Run_Cmd "ping 127.0.0.1", VISIBLE, WAIT
Run_Program "notepad.exe", "", VISIBLE, NOWAIT
End Sub
```

Analyse d'élément scriptés

1 Récupération du code

- JS : immédiat
- Flash : JPEXS FFDec
- Java : Jad
- .NET : DotPeek
- VB : voir précédemment

2 Désobscurcissement si nécessaire

- Suppression des lignes inutiles
- Ré-indentation
- Renommage des fonctions, des variables et des alias

3 Analyse statique / dynamique si la ré-interprétation est possible

Démonstration

FFDec : extracted2.swf

Exemple de méthode

- 1 Utiliser les outils automatiques
- 2 Identifier le type d'exécutable
 - PE / ELF / Mach-O / Shellcode
 - .NET / VB / AutoIT / ...
- 3 Extraire les informations associées
 - Ressources
 - Binaires accolés
 - Signatures connues
 - Entropie

Démonstration

BinWalk

Exemple de méthode

- 4 Ingénierie inverse (*Reverse engineering, RE*) : couche(s) de packing
- 5 RE : payload finale
- 6 Extraction des C&C et des différents IoCs
- 7 Si possible, capitalisation
- 8 Pour l'analyse, on peut aussi reconstruire un binaire final «propre» (ImpRec, ...)

3 Analyse des malwares

- Généralités de l'analyse
- Outils d'automatisation
- Analyse « à la main »
- **Packers**
- Obscurcissement

Besoin

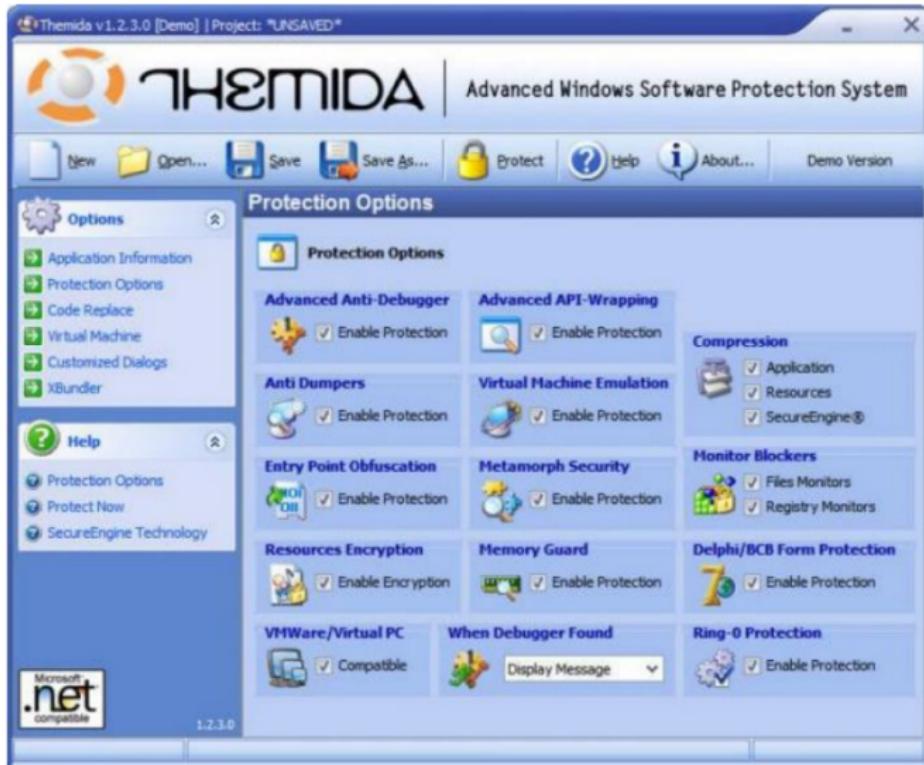
- Éviter les détections par signature
- Ajouter des mesures anti-émulation, anti-instrumentation, anti-antivirus
- Plusieurs couches pour augmenter la consommation de ressources des défenseurs
- Unpackers automatiques peu efficaces
- Fréquence de renouvellement des packers (empirique) : 2 fois / jour

Caractéristiques

- S'appliquent après la finalisation d'un binaire
- Fonctionnent comme les poupées russes
- Externalisable, des packers commerciaux existent

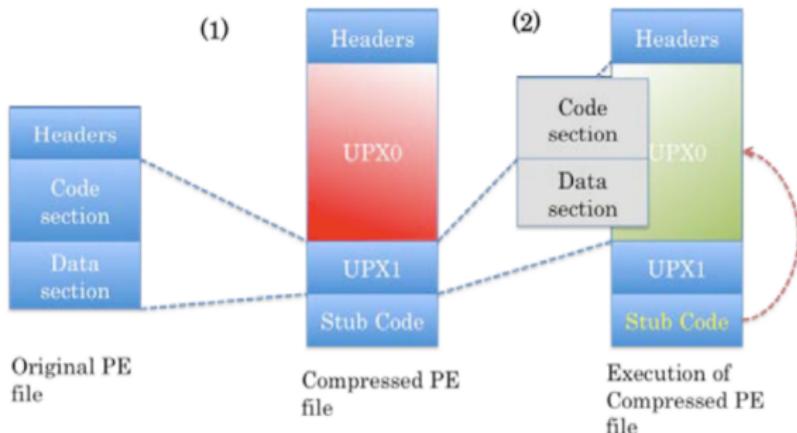
Exemples

- ASPack
- UPX
- Armadillo
- ASProtect
- Themida
- Code Virtualizer



Compression

- 1 Le binaire embarque des éléments compressés
- 2 Un «stub» de décompression est appelé
- 3 Il alloue une zone mémoire, décomprime les différents éléments
- 4 Il saute alors dans cette zone mémoire, sur l'OEP (*Original Entry Point*)



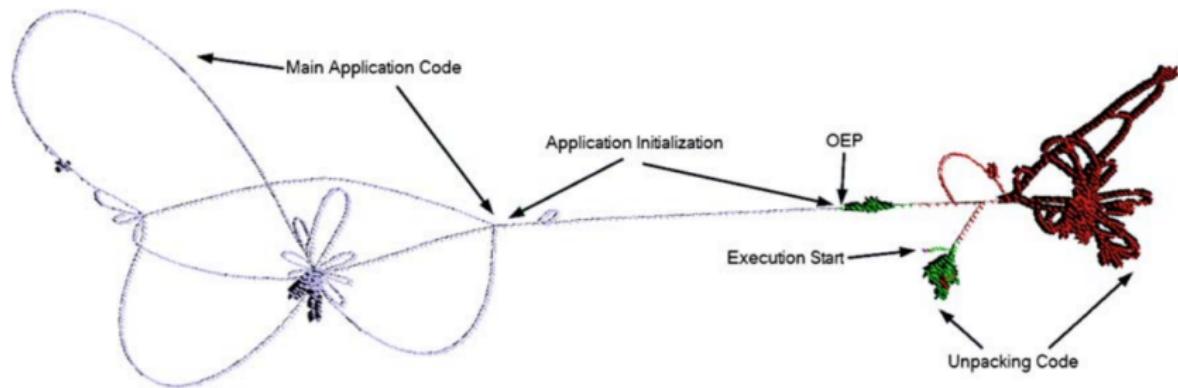
Chiffrement

- Même méthode que pour la compression
- Clés calculées à la volée, dépendantes du binaire
- Auto-bruteforce
- MD5 d'un fichier attendu → Duqu, Gauss

Améliorations

- Déchiffrement partiel et re-chiffrement
- Contrôle d'intégrité
- Surveillance mutuelle inter-threads
- “Proxy” pour les appels de fonctions

Visualisation VERA



3 Analyse des malwares

- Généralités de l'analyse
- Outils d'automatisation
- Analyse « à la main »
- Packers
- Obscurcissement

Définition

L'obscurcissement (obfuscation) consiste à rendre l'analyse de code la plus complexe et la plus coûteuse possible, tout en conservant le comportement original (la *sémantique observable*) du programme.

Motivations

- Protection du code sensible : dissimuler des clés embarquées dans le code par exemple
- Protection d'algorithme : question de propriété intellectuelle

Différents niveaux

- Code source
- Langage intermédiaire (IR)
- Code assembleur

Problèmes posés

- Obscurcissement de façon automatique.
- Compromis performances / obfuscation.

Plusieurs types d'obfuscation

- **Obfuscation du flux de contrôle (*control flow*)**
- **Obfuscation du flux de données (*data flow*)**
- Réécriture des symboles : nom de variable, de fonction...
- Chiffrement du code, packing...

Le flot de contrôle (*control flow*)

- Représente le flot d'exécution d'un programme : les différents chemins possibles lors de l'exécution
- On retrouve les boucles (for, while), les conditions (if), les appels à d'autres fonctions...
- On le modélise grâce à un *Control Flow Graph* (CFG), qui représente les *basic blocks* et les liaisons entre eux

Le flot de contrôle (*control flow*)

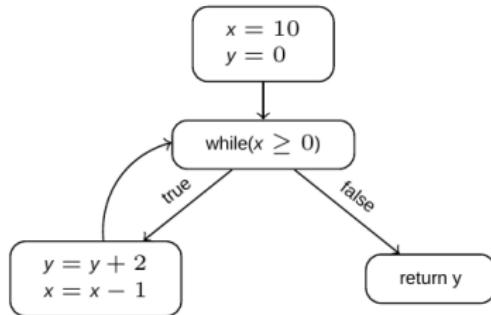


Figure – CFG de pseudo-code source

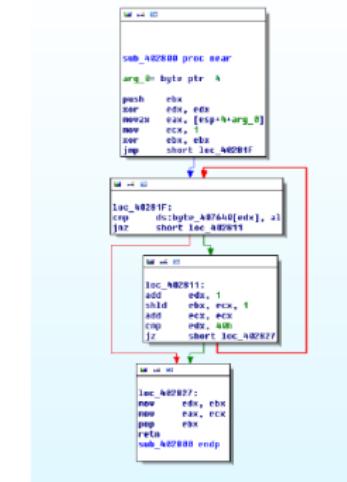


Figure – CFG de code assembleur

On cherche à transformer le CFG :

- déroulage des boucles `for` ;
- *inlining* de fonctions ;
- insertion de *junk code* ;
- prédictats opaques ;
- aplatissement de flot de contrôle (ou *control flow flattening*).

Applatissement de flot de contrôle

- Transformer la structure du programme pour rendre la reconstruction du CFG plus difficile
- Coder les informations du flot de contrôle et cacher le résultat dans le flot de données

Implémentation

- Les basic blocks sont numérotés
- Un *dispatcher* gère l'exécution
- Une variable détermine quel bloc doit être exécuté après le bloc courant
- À chaque fin de bloc, cette variable est remise à jour, et on va vers le *dispatcher* qui redirige vers le bloc suivant.

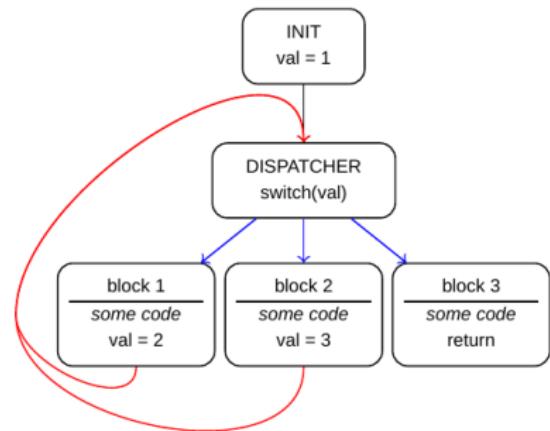


Figure – Principe de l'applatissement de code

Exemple

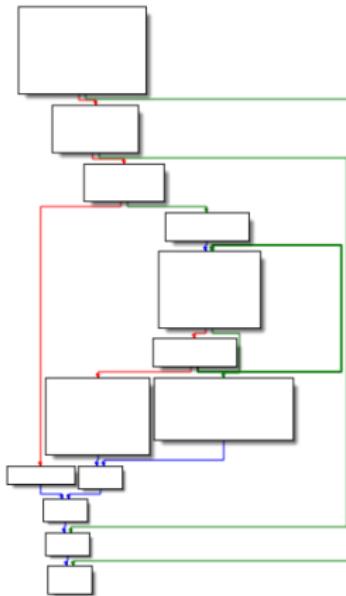


Figure – CFG original

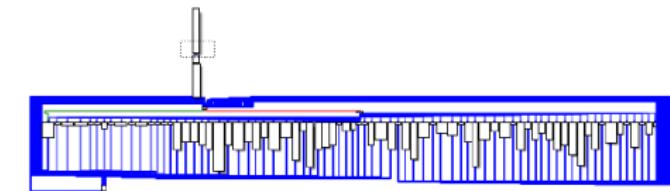


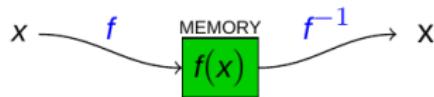
Figure – CFG après aplatissement du flot de contrôle

Plusieurs axes

- Informations fournies par les données du programme : chaînes de caractères, entiers...
- Relations entre les données ou entre les entrées et sorties (du programme, d'une fonction, d'un basic block)
- Interactions entre le programme et les données : lectures, écritures, emplacement des données dans la mémoire...

Pour complexifier l'analyse des données :

- encoder les constantes (chaînes de caractères par exemple);
- complexification des opérations arithmétiques sur les données;
 $x + y \Leftrightarrow (x \oplus y) + 2 * (x \wedge y)$
- modifier la façon dont sont stockées / manipulées les données : éclater les tableaux, changer la convention d'appel des fonctions, etc;
- encoder les données lors des lectures et écritures.



Des questions ?

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Bruyères-le-Châtel | 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00 | F. +33 (0)1 69 26 40 00
Etablissement public à caractère industriel et commercial
RCS Paris B 775 685 019

CEA DAM
DSSI
CTSI