

## Formation « Rétroingénierie de logiciels malveillants »

**Réf : REVERSE1**

Comprendre le fonctionnement des logiciels malveillants est un élément clé nécessaire auprès des entreprises afin de pouvoir répondre de manière plus efficace à vos incidents de sécurité. L'objectif de cette information est de fournir les éléments clés permettant de comprendre le fonctionnement des logiciels afin de pouvoir créer des "Indicateurs de Compromission" ainsi que des signatures permettant de détecter des versions modifiées des outils malveillants afin de détecter les mises à jour de ceux-ci sans avoir besoin de mettre à jour vos signatures. La formation vous permettra alors de pouvoir analyser tout type de menace, du client lourd à l'application "Flash" en passant par les documents malicieux (office, PDF) en passant par les sites web malveillants et les applications mobiles.

### Objectifs

- Qualifier la menace d'un logiciel malveillant
- Savoir mettre en place d'un laboratoire d'analyse des logiciels malveillants et préparer l'outillage d'analyse
- Analyser de manière statique et dynamique le comportement de logiciels malveillants
- Apprendre l'architecture x86
- Savoir identifier les structures logiques (boucles, branchement...)
- Savoir identifier des motifs utilisés par les logiciels malveillants en analysant le code
- Analyser la mémoire
- Savoir contourner les techniques d'autoprotection

### Durée & horaires

- 5 jours soit 35 heures
- Horaires : de 9h30 à 12h et de 13h30 à 18h00/18h30.

### Nombre de participant

- Minimum 8 participants – Maximum 24 participants

### Public visé

- Membres d'un SOC ou d'un CSIRT
- Équipes de réponse aux incidents
- Toute personne souhaitant réaliser des analyses avancées des menaces
- Toute personne intéressée par l'analyse des logiciels malveillants
- Professionnel de la sécurité souhaitant acquérir des connaissances en analyse de codes malveillants
- Analystes
- Responsables sécurité

### Pré-requis

- Connaître le système Windows
- Savoir programmer
- Avoir les bases en réseau
- Connaître l'assembleur

## Méthode pédagogique

- Cours magistral illustré par des travaux pratiques réguliers

## Supports

- Support de cours au format papier en français
- Ordinateur portable mis à disposition du stagiaire
- Cahier d'exercices et corrections des exercices
- Certificat attestant de la participation à la formation

## Modalité d'évaluation de la formation

- Fiche d'évaluation remise aux stagiaires à l'issue de la formation afin de recueillir leurs impressions et identifier d'éventuels axes d'amélioration

## Certification

- A l'issue de cette formation, le stagiaire a la possibilité de passer un examen ayant pour but de valider les connaissances acquises. Cet examen de type QCM dure 1h30 et a lieu durant la dernière après-midi de formation. La réussite à l'examen donne droit à la certification FORENSIC2 par HS2.

## Programme

### Section 1 : Introduction aux bases de l'analyse de logiciels malveillants

- Processus et méthodologie générique
- Analyse statique :
  - Analyse des métadonnées
  - Analyse statique
- Analyse dynamique
  - Comportemental
  - Débugger
- Construire son laboratoire d'analyse
  - Simuler internet
  - Utilisation de la virtualisation
    - Contournement des mécanismes de protection anti-VM
    - Simulation d'architecture "exotique" (IOT)
    - Construction du laboratoire et boîte à outils
  - Sandbox

### Cas d'analyse

- Introduction au langage assembleur
  - Guide de survie des instructions de bases
  - Instruction modifiant le flux d'exécution
  - Présentation des registres
- Conventions d'appels
  - Spécificités des langages objets
- IDA Pro:
  - Introduction
  - Prise en main de l'outil (création de scripts)
- Chaîne de compilation et binaires
  - Fuite d'informations possibles
  - Imports d'information dans IDA

## Section 2 : Système d'exploitation

- Introduction aux systèmes d'exploitation
  - Processus vs thread
  - Scheduler
  - Syscall
  - Différence processus vs thread
- Format d'exécutable
  - Format PE
    - Présentation des informations
- Structures internes
  - SEH
  - TEB
  - PEB
  - SSDT
- Introduction au "kernel debugging"

## Section 3 : Mécanismes de protection (DRM ou packer)

- Introduction aux outils de DRM/Protection de code
  - Comment les identifier ?
    - Quels sont les impacts ?
- -- Introductions aux différentes techniques de protection :
  - Anti-désassemblage
  - Anti-debogage
  - Obscurcissement du CFG
  - Machine virtuelle
  - Évasion (détection de sandbox/Virtualisation)
- Analyse de packer
  - Présentation de la méthode générique d'unpacking
  - Découverte de l'OEP
  - Reconstruction de la table d'imports
    - Miasm2 :
      - Unpacking automatique

## Section 4 : Malwares

- Catégoriser les logiciels malveillants en fonction de leurs API
- Keyloggers
- Rootkits (userland et kerneland)
- Sniffers
- Ransomwares
- Bots et C2
- Injection de code
  - Technique de contournement de flux d'exécution (ie: detour)
- Shellcode
  - Techniques et outils d'analyses
  - Miasm2
  - Unicorn Engine

## Section 5 : Autres types de malwares

- Malware "Web" (JavaScript/VBScript)
  - Analyse statique et dynamique
  - Limitation des navigateurs
- Malwares Flash
- Applications mobiles Android
- Documents malveillants

- Suite Office
- PDF
- RTF
- Malwares .Net

### **Section 6 : Threat Intelligence**

- Création de signatures Yara
- Communication et base de connaissances
  - MISP
  - Yeti

### **Section 7 : Avantage de l'analyse mémoire**