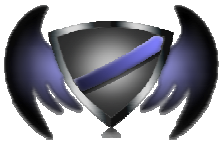
S E C U R I N E T S

Club de la sécurité informatique

I N S A T

# Dans le cadre de *SECURIDAY 2009*

SECURINETS



*Présente*

# Atelier : Analyse statique de malware

**Formateurs: 1. Ben Youssef Manel**

**2. Abdi Safa**

**3. Labidi Tarek**

**4. El Azrek Imen**

## 1. Présentation :

L'analyse de codes malveillants est devenue aujourd'hui une activité très importante dans le cadre de la lutte contre la cybercriminalité.

Voici quelques-uns des objectifs habituels de cette analyse :

- Vérifier si un fichier suspect est effectivement malveillant ou s'il s'agit d'un faux positif.
- Déterminer s'il s'agit d'un code malveillant générique (connu ou non) ou bien d'une attaque ciblée.
- Obtenir des informations sur la source de l'attaque.
- Déterminer toutes ses actions sur le réseau : connexions sortantes, réplication, serveur en écoute, ...
- Déterminer ses "fonctionnalités" malveillantes : virus, ver, cheval de Troie, bombe logique, relais de spam, Botnet, ...
- Déterminer quelles sont les machines ciblées et lesquelles pourraient être vulnérables ou déjà compromises sur le réseau.

… etc.

L'analyse statique est l'une des techniques utilisé afin d'atteindre ces objectifs, elle consiste à explorer le contenu des fichiers suspects à l'aide d'outils divers, dans le but d'en extraire le maximum d'informations sans pour autant l'exécuter. Ces informations nous permettent de déterminer le comportement et d'étudier sans risque le contenu de ce code.

## 2. Outils utilisés :

Dans cette partie, on présentera brièvement les outils utilisés :

**2.2- IDA Pro v 5.2**: celui la aussi est un désassembleur, c'est un logiciel commercial, il va nous permettre de passer du code binaire du malware vers son code assembleur. De plus, on peut lui ajouter le plugin **Hex-Ray** qui va nous permettre de faire la décompilation (c'est-à-dire revenir au code C).

**2.3- PEiD :** «c'est un logiciel de détection de protection de PE (programme exécutable).
PEiD permet de détecter plus de 475 protections et packers différents dans sa version de base. *(Le Packer est un "utilitaire" qui sert à compresser un programme exécutable (.exe, .dll, .ocx etc. ...) et à le crypter simultanément. Ce qui peut rendre un malware difficilement détectable par les anti-virus et les anti-trojans.).*

cela nous évite de passer des heures à chercher désespérément pourquoi le fichier traduit ne fonctionne pas.
Une fois que PEiD nous indique si un fichier est protégé, vous pouvez tenter de le décompressé directement à l'aide des plugins inclus ou de ceux disponibles sur le net.

On peut télécharger cet outil à partir de l'adresse suivante: http://www.trad-fr.com/telecharger/details.php?file=45

**2.4- PE. Explorer** : PE Explorer est un puissant utilitaire qui permet de désassembler des applications Windows afin de visualiser leur fonctionnement interne en assembleur.

Il facilite l'analyse du code source et corrige les sommes logiques destinées à la détection d'erreurs. Il accepte les formats EXE, DLL, MSSTYLES, BPL, DPL, SYS, CPL, OCX, SCR ainsi que ceux d'autres exécutables win32.

On peut télécharger PE Explorer à partir de l'adresse suivante: http://www.toocharger.com/fiches/logiciels/pe-explorer/7898.htm

## 3. Partie pratique :

Pour analyser un fichier suspect dont on n'a aucune connaissance préalable, nous devons passer par un certain nombre d'étapes, afin d'extraire un maximum nombre d'information utile par la suite pour déterminer si le code est malveillant ou pas :

***Remarques :*** *- Tout au long de cet atelier, on utilisera une machine équipée d'un système Windows XP.*
*-Nous sommes entrain de parler de code malveillant binaire ou exécutable et non pas de script (comme VBScript worm, JavaScript…)*
*-Il est à noter aussi que ce tutorial propose une démarche pour l'analyse d'un malware, cette démarche n'est pas forcément suivie de manière stricte, elle donne juste une idée sur un certain nombre de techniques utilisées dans ce domaine.*

### 3.1) Etape 1 :

Il faut tout d'abord préparer un environnement de travail sain, c'est-à-dire une machine isolé, non connecté à aucun réseau. C'est ce qu'on appel un **Sandbox**. Cette étape permet de sécuriser la machine en cas où le bot s'exécutera par erreur et d'éviter qu'il se propage à travers le réseau.

➔ Dans cette étape, nous pouvons soit utiliser une machine non connectée au réseau, soit utiliser VMware.

### 3.2) Etape 2 :

Une fois l'environnement convenablement préparé, on commence par déterminer la signature du malware à l'aide d'une fonction de hachage qui va générer une empreinte unique de ce fichier. Cette empreinte va nous permettre de savoir si ce malware a été déjà analysé par une autre personne. Pour ce faire, on peut trouver des sites qui permettent d'accomplir cette tâche. Il suffit d'entrer le fichier et il va vérifier s'il s'agit d'un malware connu ou non. On peut citer comme exemple le site suivant www.virustotal.com qui permet d'analyser le fichier avec 31 antivirus différents. Cette étape est très importante dans le but de ne pas perdre du temps dans l'analyse d'un malware déjà connu.


***Figure 1 : aperçu sur l'interface du site www.virustotal.com***

### 3.3) Etape 3 :

Voila, une fois qu'on a vérifié que ce malware n'a pas été analysé par une autre personne, on peut passer à l'étape suivante qui consiste à vérifier si ce malware est compressé et crypté ou pas et si c'est le cas, vérifier par quel outil et par quel version d'outil il a été compressé et crypté. Là, on peut utiliser le logiciel **PEiD** qui va nous permettre d'identifier la signature du Packer. Une fois que ce dernier est identifié, on peut l'utiliser pour décompresser le fichier suspect. Cette étape nous facilitera par la suite la lecture du code et du coup d'éviter les fausses lectures.

- On lance PEiD, puis on donne le chemin d'accès au fichier suspect :

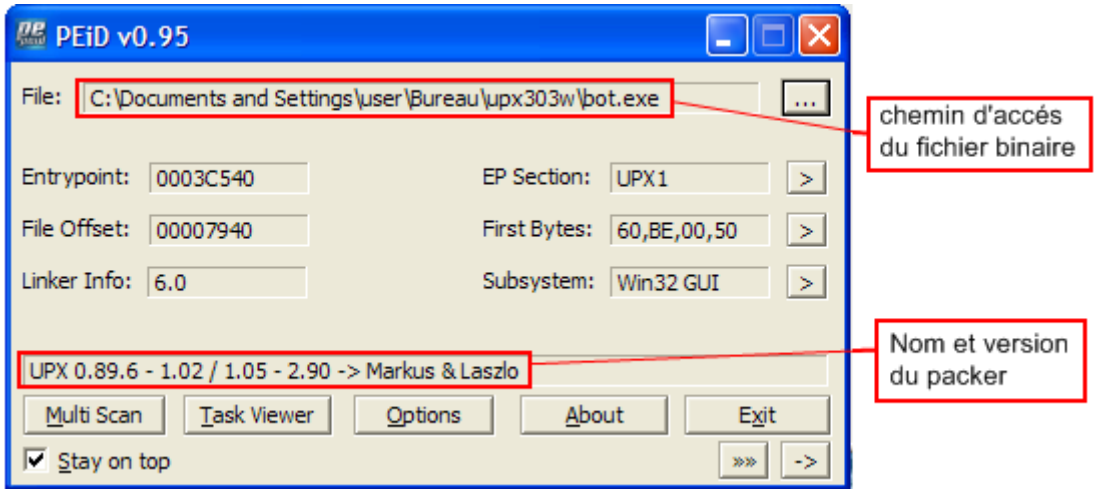



***Figure 2 : décompression d'un fichier avec PEID***

### 3.4) Etape 4 :

Voila, notre code est bien décompressé, on peut maintenant essayer d'analyser les chaines de caractères visibles au niveau du code à l'aide par exemple du programme « **Strings** » qui sert à extraire des chaînes de caractères d'un fichier binaire. Il est à noter que les développeurs de malwares, dans la plupart des cas, signent leurs œuvres en ajoutant des informations personnelles au niveau du code, tout simplement, pour se montrer et pour avoir un moyen de copyright. Donc, cette étape va nous aider à identifier cette personne et ensuite de la suivre malgré que trouver une telle information ne confirme pas qu'il s'agit bien d'un bot.

Dans cette étape, on ouvre le fichier suspect avec l'outil **PE Explorer**.

Nous remarquons qu'il y a différentes fenêtre qui s'ouvrent, on s'intéresse a l'onglet **Strings** (voir figure ci-dessous)

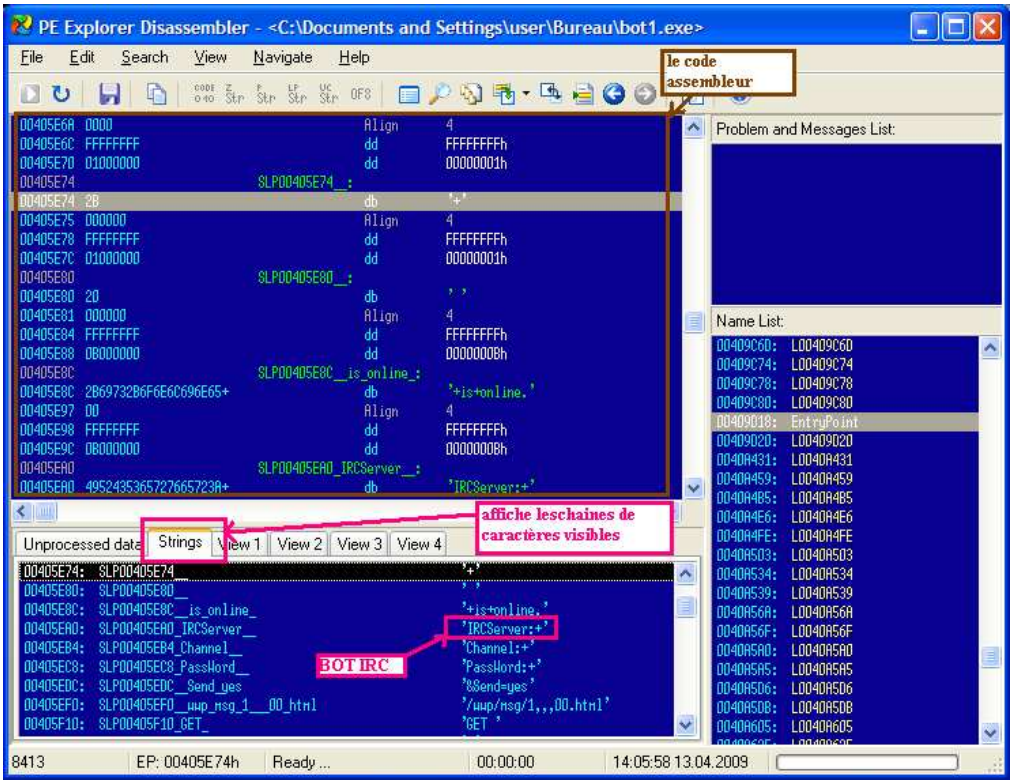



***Figure 3 : analyse d'un fichier exécutable avec PE Explore***

- **Explication du résultat obtenu :** on remarque au niveau des chaines de caractères détectées l'existence de la chaine « IRCServer », donc on peut déduire qu'il s'agit d'un malware qui essaye de se connecter à un serveur IRC. Donc, il est très probable qu'il s'agit d'un bot IRC. On peut ainsi trouver les instructions que utilise ce bot pour communiquer avec son maitre.

### 3.5) Etape 5 :

Nous arrivons maintenant à l'étape de conversion du code. Une première conversion pourra être effectuée qui consiste au désassemblage c'est-à-dire convertir le code binaire en un code assembleur. Pour ce faire, on peut utiliser divers outils comme **IDA PRO, PE Explorer** ou **OllyDbg.** Mais à ce niveau là, il faut avoir une bonne connaissance du langage assembleur pour bien le comprendre et une bonne expérience pour pouvoir détecter les instructions qui nous permettent d'identifier le fonctionnement du malware. Par exemple, détecter les ports qu'il utilise, voir s'il essaye de modifier des entrées au niveau du registre, etc.…

### 3.5.1) Désassemblage à l'aide de l'outil PE Explorer :

(Voir figure 3)

### 3.5.1) Désassemblage à l'aide de l'outil IDA Pro:

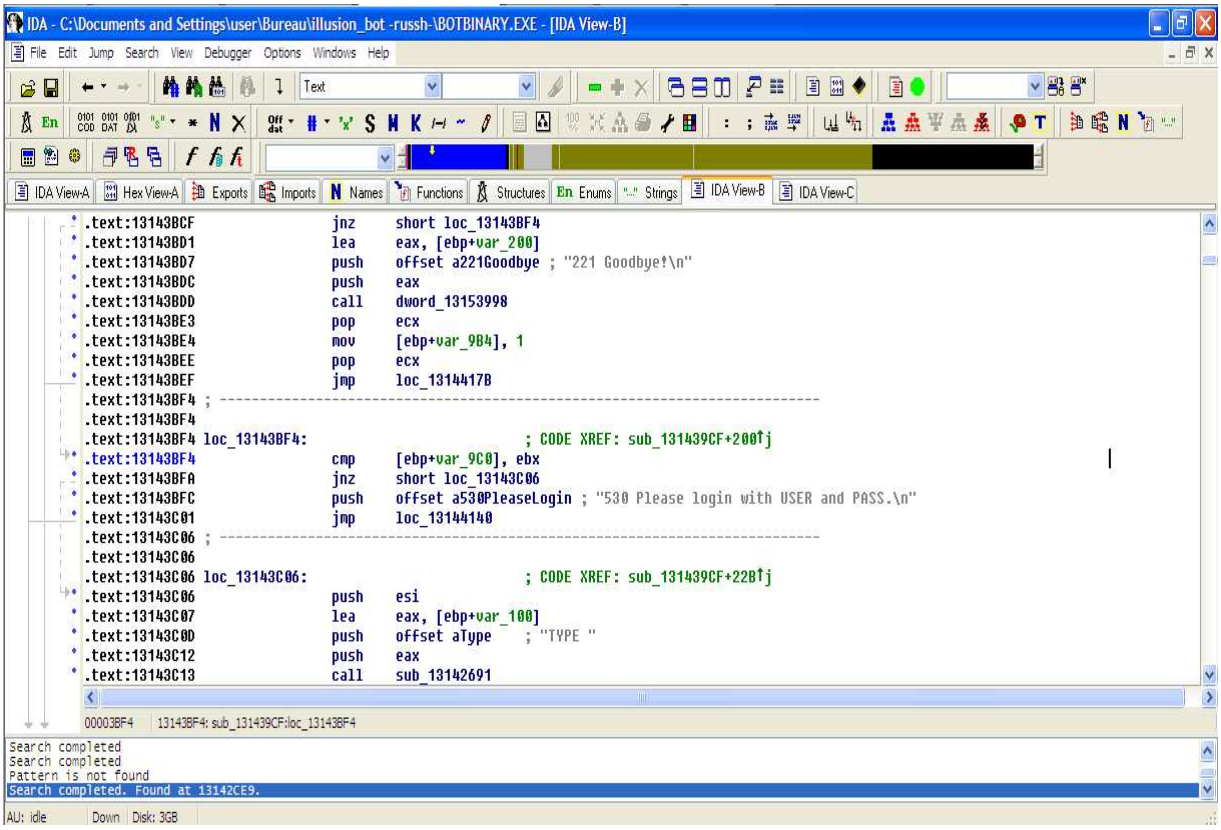



***Figure 4 : analyse d'un fichier exécutable avec IDA PRO***

Il est possible d'analyser le code assembleur afin de comprendre le fonctionnement de ce code exécutable, on peut par exemple trouver les instructions qui permettent d'ouvrir une connexion, de modifier le registre Windows, de créer des fichiers sur le disque …

Mais vu la longueur du code assembleur, cette étape s'avère difficile et requiert beaucoup d'expérience et une grande maitrise en ce langage.

### 3.6) <u>Etape 5:</u>

Cette dernière étape pourra être suivie d'une deuxième étape de conversion qui consiste à la décompilation qui nous permet de passer au code source C ou C++ par exemple. Mais cette conversion génère généralement des erreurs c'est-à-dire que c'est presque très difficile de

retrouver le code initial. Pour faire la décompilation, on peut utiliser l'outil **IDA Pro** avec le plugin **Hex-Rays**.

## 4. Conclusion

Pour conclure, cette technique d'analyse est très puissante avec un minimum de risque, mais ça demande beaucoup de temps et de connaissances en matière de programmation, de systèmes d'exploitation et surtout en langage assembleur.

C'est pourquoi, cette analyse est complétée par une phase d'analyse dynamique du malware, qui consiste à l'exécuter tout en surveillant son comportement à savoir les modifications qu'il apporte au système, les connexions qu'il tente d'ouvrir a travers le réseau, etc …

Nous espérons que cet atelier vous ait apporté le plus en matière de sécurité. Nous sommes donc ouverts à vos questions et à vos participations au sein de notre club **SECURINETS à l'INSAT**.