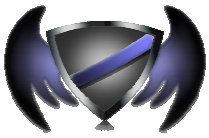
S E C U R I N E T S
Club de la sécurité informatique
I N S A T

Dans le cadre de *SECURIDAY 2009*

SECURINETS



Présente

Atelier : Analyse dynamique d'un BOTNET

Formateurs: 1. Akrimi Hayfa
2. Bel Hadj Aleya Atef
3. Guizeni Mejda
3. Hadj Taieb Maher
4. Hosni Aymen
5. Jemmali Hassan

1. Introduction :

L'Internet et les réseaux informatiques ont longtemps été infesté par du code malveillant et ses effets néfastes. Ce tutorial vous explique l'utilisation pratique et basique dans un environnement contrôlé de l'analyse dynamique des malwares.

1.1 Définition :

L'Analyse dynamique appelé aussi analyse comportemental est une méthode qui consiste à faire exécuter un échantillon de code malveillant sur une plate-forme conçue pour observer toutes ses actions.
Dans la plupart des cas, c'est une approche très efficace et rapide pour déterminer la nature et le comportement du code malveillant.
Certaines caractéristiques sont importantes pour l'analyse de code malveillant, notamment :

- La capacité à se répliquer
- Les techniques de camouflage
- Une connexion réseau pour le contrôle en local ou à distance
- Une protection contre le désassemblage et/ou le débogage
- Une détection de machine virtuelle
- La nature du code malveillant : fichier exécutable, document malformé contenant un exploit, script dans une page web, ...

1.2 Différents types de «Sandboxs»:

On peut réaliser tout seul un environnement d'analyse dynamique de malwares(SANDBOX) dans une machine virtuelle isolée des réseaux mais on risque toujours d'être contaminé soit par des malwares intelligents soit en faisant des manipulations risquées. Une des solutions qui s'offre à nous est d'utiliser des "Sandbox Online" tel qu'Anubis, CWSandbox, Norman Sanbox, JoeBox. Il existe aussi une distribution appelée ZeroWine, basée sur debian, qui génère de bon rapport (appel système, fichiers ouverts, contenu de l'exécutable, parties mémoires utilisés...) sur les exécutables qu'on lui soumet. Ces solutions ont l'avantage d'éviter que les malwares qu'on veut tester ne contaminent notre environnement de travail. Toutefois, leurs possibilités restent limitées.

2. Les outils utilisés :

2.1 Machine virtuelle :

Dans nos tests on a choisi de travailler sur des machines virtuelles pour les raisons suivantes :

- Sécurité : les machines virtuelles peuvent être connectées par un réseau virtuel totalement indépendant de tout réseau opérationnel, sans risque d'infecter d'autres machines.

- Rapidité et efficacité : une machine virtuelle peut être stoppée, restaurée et redémarrée en quelques secondes.
- Facilité d'emploi : n'importe quel état de fonctionnement peut être sauvegardé dans un « snapshot» en en quelques secondes. Il est ensuite très simple de restaurer tout snapshot précédent, et de comparer plusieurs états entre eux.
- Les machines virtuelles peuvent être facilement copiées, dupliquées et modifiées pour constituer une bibliothèque de toutes les versions d'un système d'exploitation ou d'une application.

2.2 RegMon :

Il affichera l'ensemble des entrées registre utilisées par un programme

2.3 FileMon :

Il affichera l'ensemble des fichiers utilisés par le code malveillant qui peut se modifier ou se répliquer en utilisant différents noms dans plusieurs endroits.
FileMon peut aussi étudier le comportement du code malveillant qui télécharge et exécute d'autres fichiers comme des : backdoors depuis un lieu distant et le place sur le système infecté.

2.4 Wireshark :

Capture et analyse des Paquets.
La plupart des malwares dans la nature actuellement essayaient de contaminer d'autres machines sur le réseau ou bien font partie des botnets et envoient ainsi beaucoup de spam depuis des machines infectées ou bien peuvent également envoyer beaucoup d'informations depuis des systèmes compromis comme les habitudes de navigation des utilisateurs, mots de passe, détails de comptes bancaires etc. Pour détecter cela on doit utiliser le Sniffer de paquets Wireshark qui peut capturer le trafic réseau passant par la machine infectée.

2.5 Process Explorer :

Il est utilisé pour étudier les processus lancé par le bot ou le code malveillant.

2.6 SYSANALYSER:

C'est une application d'analyse des malwares en temps réel qui surveille les différents aspects du système et les états des processus.

3. Partie pratique:

3.1 Problèmes rencontrés par l'analyse Dynamique:

Comme vous pouvez l'imaginer, les Hackers qui créent les malwares (virus, bots, worms, trojans...) n'aiment pas qu'on surveille de prés leurs "progénitures" dans des environnements de test (SandBox). Ainsi, il existe des malwares qui vont chercher des régions mémoires spécifique aux machines virtuelles, chercher des drivers de matérielle qui leurs est propre ou

même chercher des débuggeurs connus ouvert dans la liste des processus. Si ces recherches sont fructueuses, alors le malware ne va pas s'exécuter.

Parfois même, des auteurs de malwares incluent des exploits qui, dans le cas des "SandBox" par exemple, essayeront d'attaquer la machine du chercheur en exploitant des failles, ou en exécutant des commandes dangereuses. D'autre fois, les malwares n'exécutent pas leurs instructions originales mais d'autre instruction pour faire diversion sur leurs vraies intentions.

3.2 Présentation des différentes étapes de l'analyse

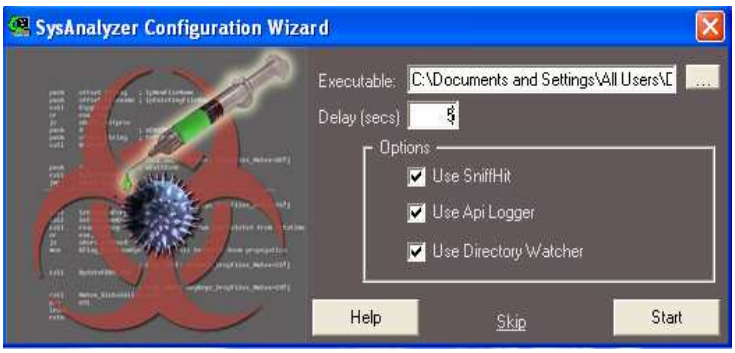
3.2.1 Préparation de l'environnement de l'analyse

- Installation de VMware.
- Création d'une nouvelle machine virtuelle Windows.
- Installation de SysAnalyser et wireshark.

3.2.2 Présentation des étapes de l'analyse :

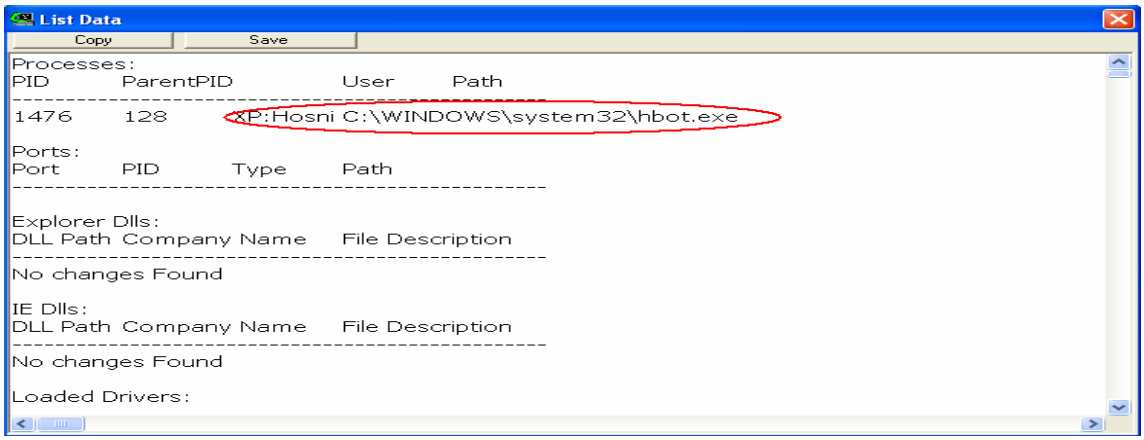
❖ Lancement de l'exécutable sur la machine virtuelle

- Dans cette étape, on lance le fichier exécutable et on observe les différentes modifications réalisées au niveau du registre, ficher, API ...Pour réaliser cette étape on utilise **SysAnalyser.**

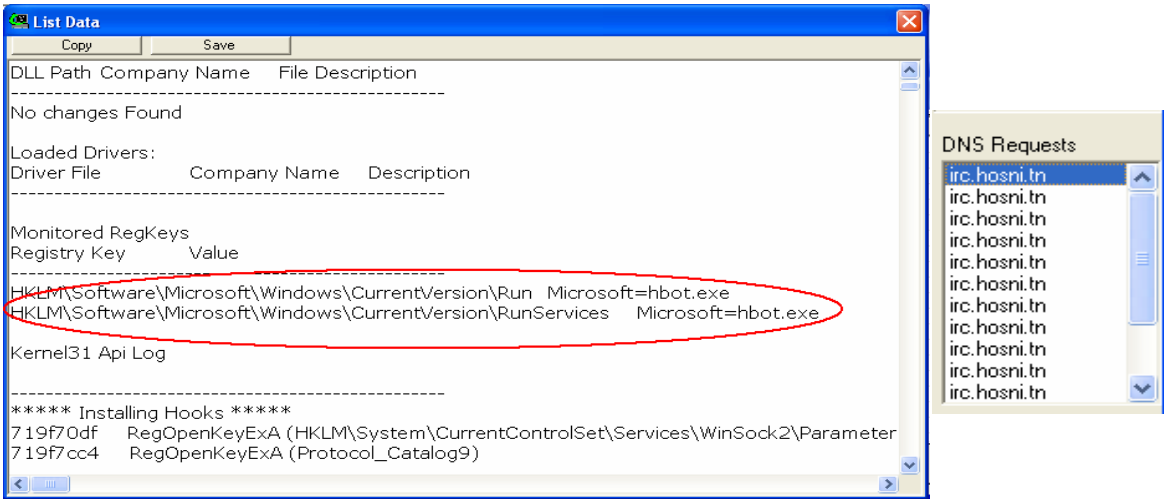


❖ Analyse du rapport

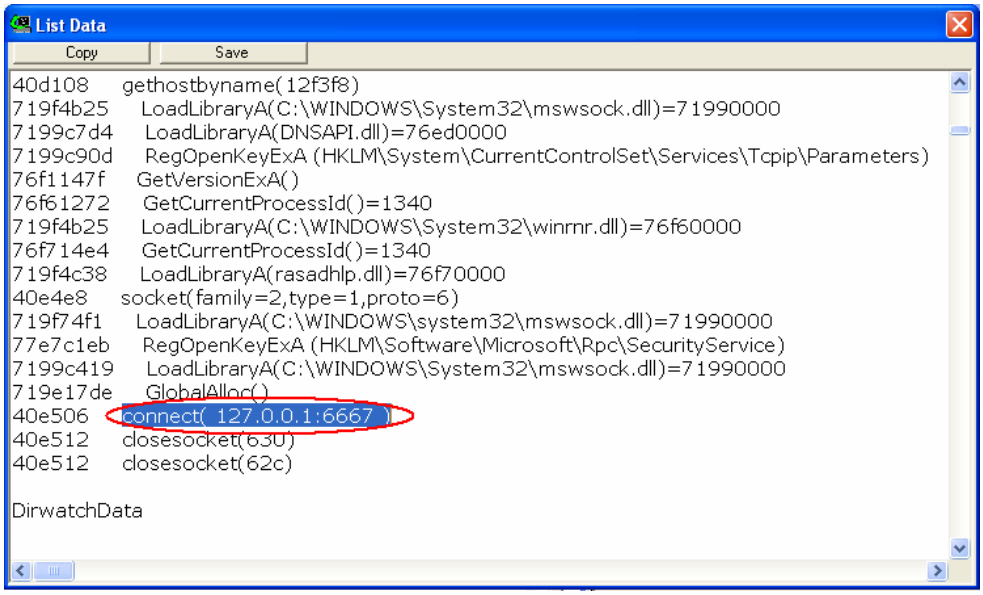
- SysAnalyser génère un rapport détaillé qui illustre les différentes modifications



- Création et lancement d'un exécutable hbot.exe



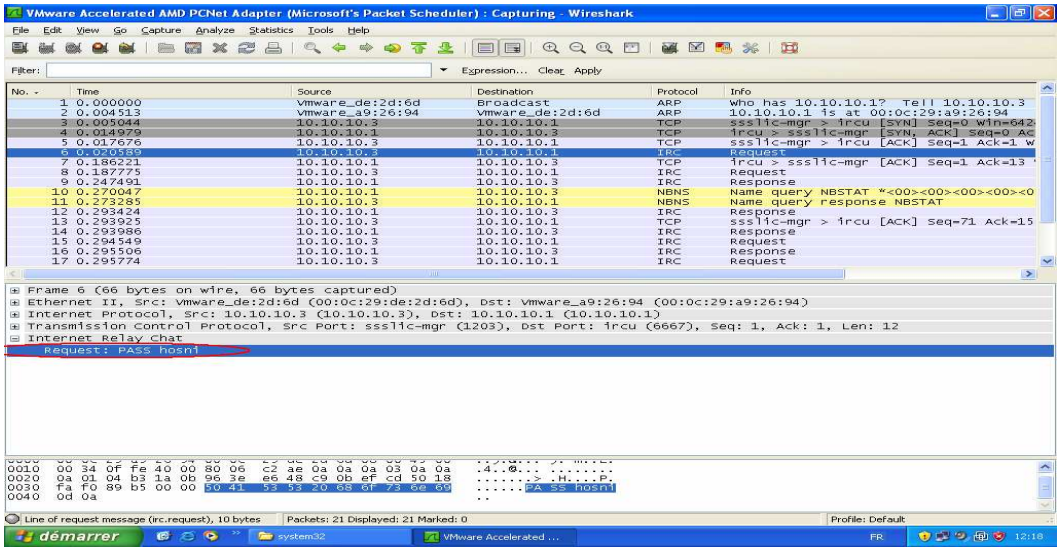
- Modification des clés au niveau des registres et demande de résolution DNS



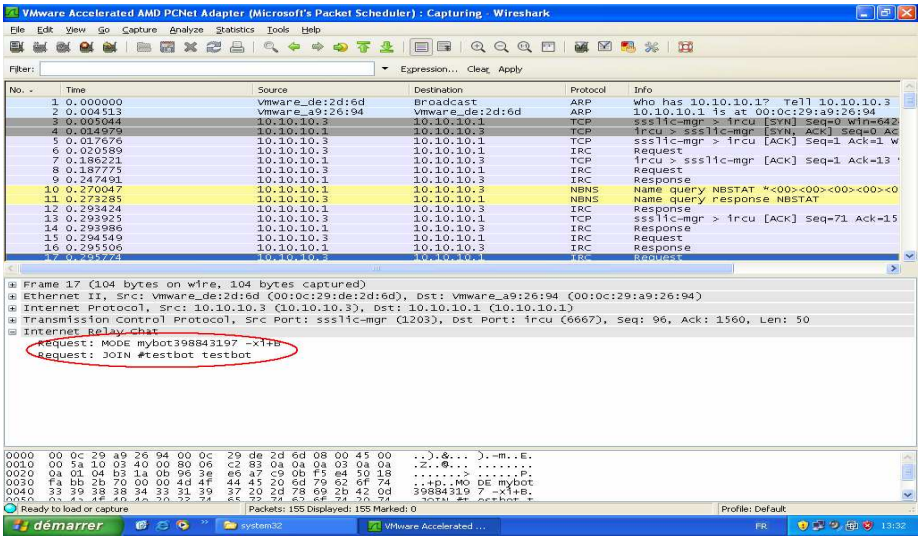
- Après avoir configuré le fichier host pour rediriger la demande de connexion à notre machine local on remarque que le programme veut se connecter au port 6667 qui est un port par défaut d'un serveur IRC

→ Donc on lui prépare un serveur IRC sur une machine virtuelle.

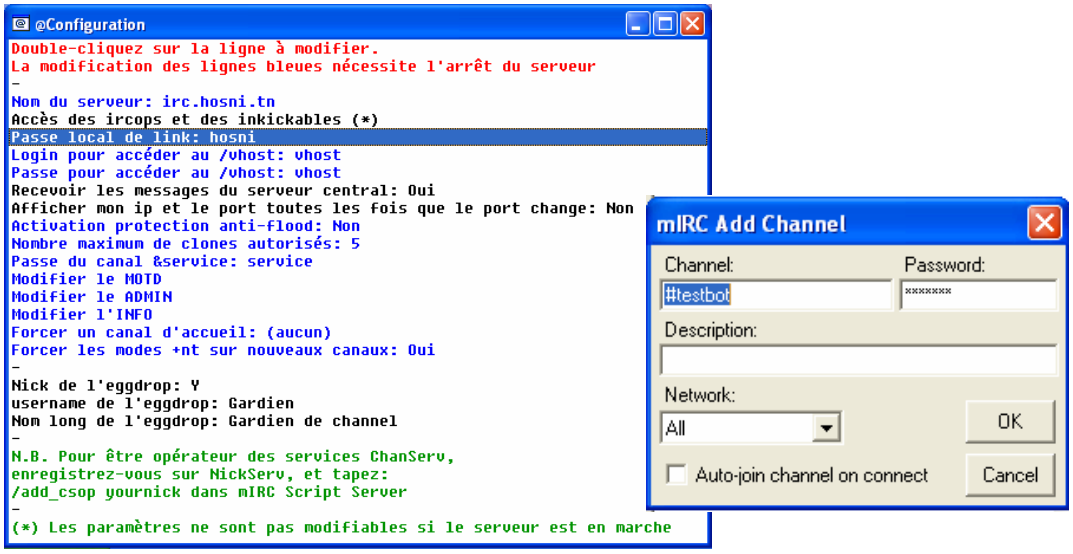
❖ Analyse du trafic réseau :



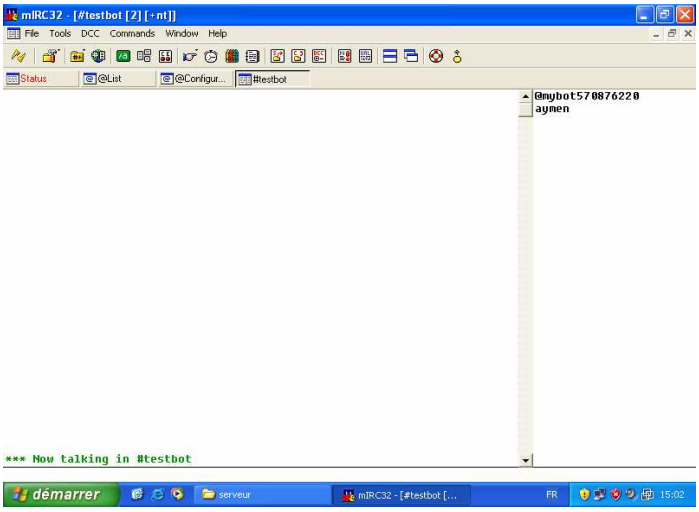
- Détection du mot de passe de connexion au serveur IRC.



- Détection de login et de mot de passe de connexion au canal.



- Configuration des paramètres nécessaires pour que le bot puisse se connecter au serveur et au canal.



- Connexion réussite. Maintenant on sait que le bot testé est un bot IRC qui essaye de se connecter à un serveur IRC pour recevoir les ordres d'un hacker qui contrôle le salon sur lequel notre malware essaye de se connecter. L'étape suivant, consiste, pour les autorités concernées, de prendre les mesures nécessaires pour trouver ou se trouve le serveur IRC originale et essayer d'arrêter son fonctionnement.

4. Conclusion :

L'analyse dynamique combinée à l'analyse statique permet une étude en profondeurs des actions et des parades utilisés par les malwares pour contrôler nos pc, voler nos données, les détruire....Cette combinaison sera plus redoutable si elle devient automatisée.