le cnam

SEC102 - TP04
Comparaison des types d'analyse

Date	Version	Description	Auteurs
18/02/2023	1	Retranscription collective des informations présentées en cours	Bartel Cantin Bialas Alexis Dufourt Marvin Faussurier Marc N'Guessan Jérémy

Sommaire :

Contexte : **2**

1. Analyse statique : **3**

1.1. Avantages : 3

1.2. Inconvénients : 3

1.3. Outils utilisés pour l’analyse statique : 3

2. Analyse dynamique : **4**

2.1. Avantages : 4

2.2. Inconvénients : 4

2.3. Les outils pour l’analyse Dynamique : 4

3. Conclusion : **5**

4. Glossaire: **6**

Contexte :

L'analyse de codes malveillants est le processus d'inspection d'un programme informatique ou d'un ensemble de programmes pour détecter et évaluer les menaces potentielles qu'ils représentent pour les systèmes informatiques.

1. Analyse statique :

1.1. Avantages :

Rapidité : Peut-être plus rapide que l'analyse dynamique.

Détection : Peut détecter des vulnérabilités qui ne sont pas visibles lors de l'exécution.

Sans risques : Pas d'exécution du programme donc moins de risques d'infection.

1.2. Inconvénients :

Adaptabilité : Peut ne pas détecter des vulnérabilités qui nécessitent une exécution

Ressources : Nécessite un programme déchiffré ou décompresser

Accessibilité : Compliqué face à un code obfusqué

Connaissances : Nécessite une bonne connaissance du format d'exécutable du fichier en question

1.3. Outils utilisés pour l'analyse statique :

- **Ghidra** de la NSA
- **PExplorer** pour visualiser les imports / exports
- **HxD** pour visualiser l'exécutable en hexadécimal
- **JustDecompile** pour décompiler du .NET
- **JD** pour décompiler du Java
- **IDA** pour décompiler du C

2. Analyse dynamique :

2.1. Avantages :

Précision : permet de déterminer les comportements réels du logiciel malveillant en le faisant tourner sur un système.

Détection complète : permet de déceler les logiciels malveillants qui peuvent être passés au travers de l'analyse statique.

2.2. Inconvénients :

Risques de sécurité : expose le système en matière de sécurité en faisant tourner le code malveillant sur celui-ci. Le système où l'analyse est réalisée doit être isolé du système de production.

Temps et coûts: plus longue et coûteuse que l'analyse statique car elle nécessite la mise en place et l'utilisation d'un système.

2.3. Les outils pour l'analyse Dynamique :

- **CheatEngine** pour analyser notamment l'état des variables RAM et detecter des changements
- **OllyDBG** pour appliquer des modifications dans l'executable
- **StudPE** pour pour injecter un DLL dans l'executable
- **WireShark** pour analyser des paquets
- Librairie **Detours** de Microsoft pour faire des detours d'appel systeme

3. Conclusion :

En conclusion, l'analyse statique et dynamique sont deux approches complémentaires utilisées pour détecter et évaluer les menaces potentielles des codes malveillants. L'analyse statique est basée sur l'examen des caractéristiques du code sans l'exécuter, tandis que l'analyse dynamique implique l'exécution du code malveillant dans un environnement contrôlé pour observer son comportement.

4. Glossaire :

Code malveillant : également appelé "malware", c'est un programme informatique conçu pour nuire à un système informatique ou à des données. Les exemples courants de code malveillant incluent les virus, les chevaux de Troie, les logiciels espions et les ransomwares.

Paquets : en informatique, un paquet (ou "packet" en anglais) est une unité de données qui est transmise à travers un réseau informatique. Les paquets peuvent contenir différents types d'informations, tels que des données de courrier électronique, des fichiers, des images, des vidéos ou des instructions de commande.

DLL : une DLL (ou "Dynamic Link Library") est un fichier informatique contenant des routines de code qui peuvent être utilisées par plusieurs programmes simultanément.

RAM : La RAM (Random Access Memory) est une forme de mémoire vive qui est utilisée pour stocker temporairement des données et des programmes en cours d'exécution dans un ordinateur ou un autre appareil électronique.