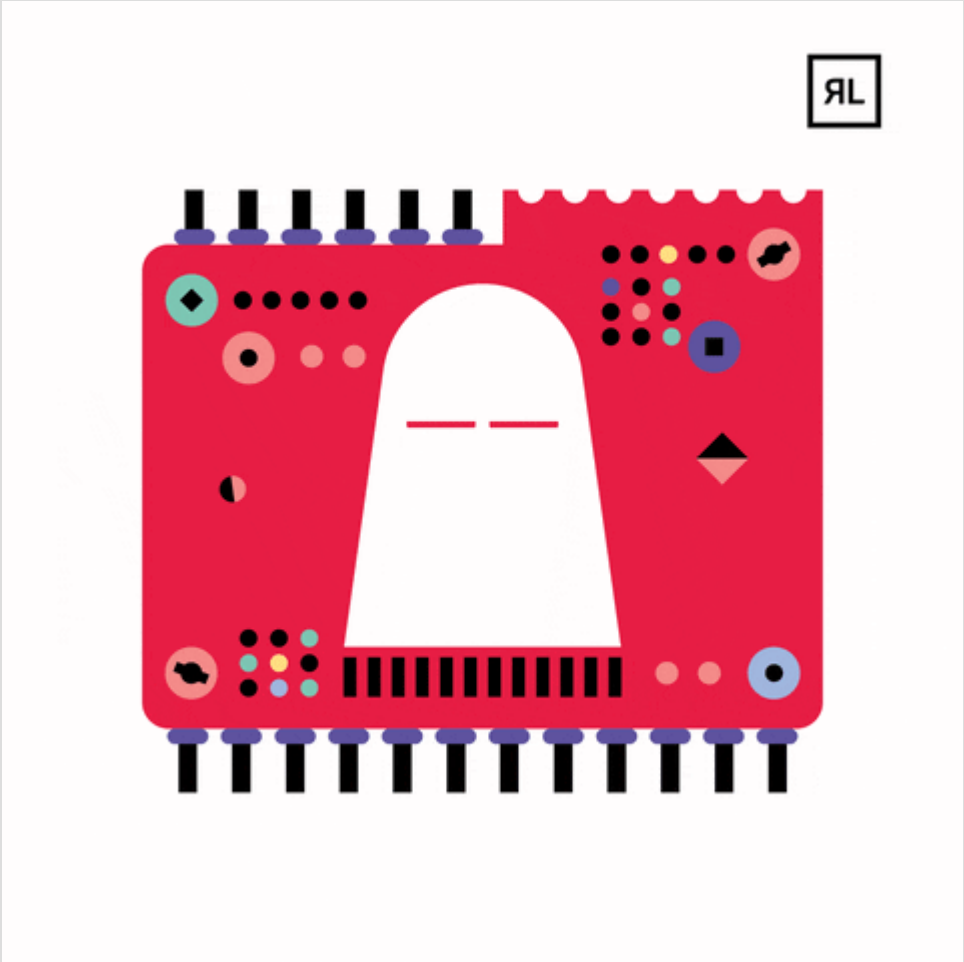
Introduction à l'analyse de malwares (analyse statique)



Objectifs de l'analyse de malwares

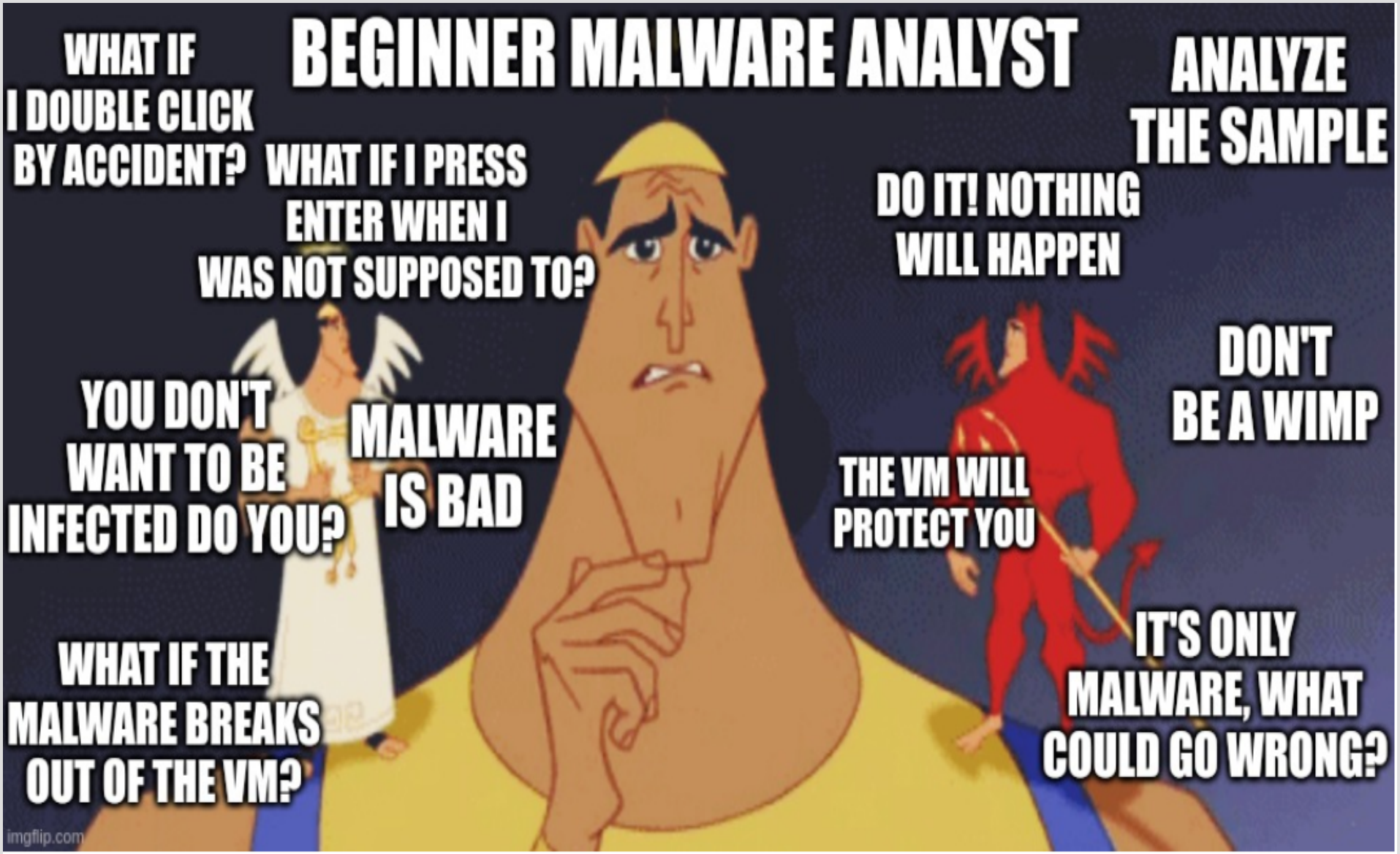
- Comprendre ce qu'un binaire de logiciel malveillant peut faire, comment le détecter sur les systèmes et le réseau :

Quels sont les indicateurs qu'un malware a été exécuté sur une machine ? Y a-t-il des fichiers, des processus ou peut-être une tentative de communication "inhabituelle" ?

Comment le logiciel malveillant se comporte-t-il ? Tente-t-il d'infecter d'autres appareils ? Chiffre-t-il les fichiers ou installe-t-il quelque chose comme une porte dérobée ou un outil d'accès à distance (RAT) ?

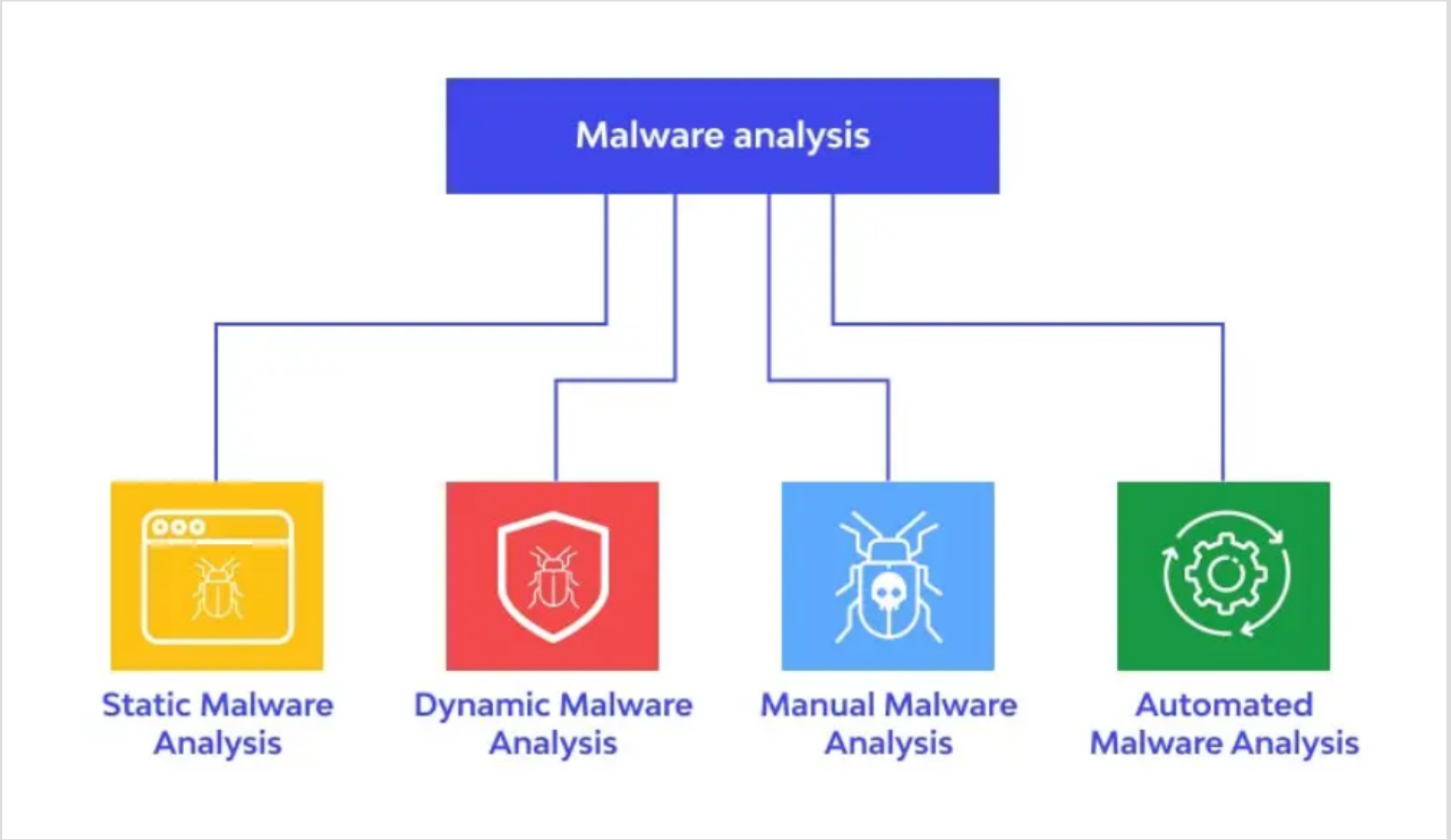
Le plus important : pouvons-nous finalement prévenir et/ou détecter une nouvelle infection ? !

Mesures de précaution :

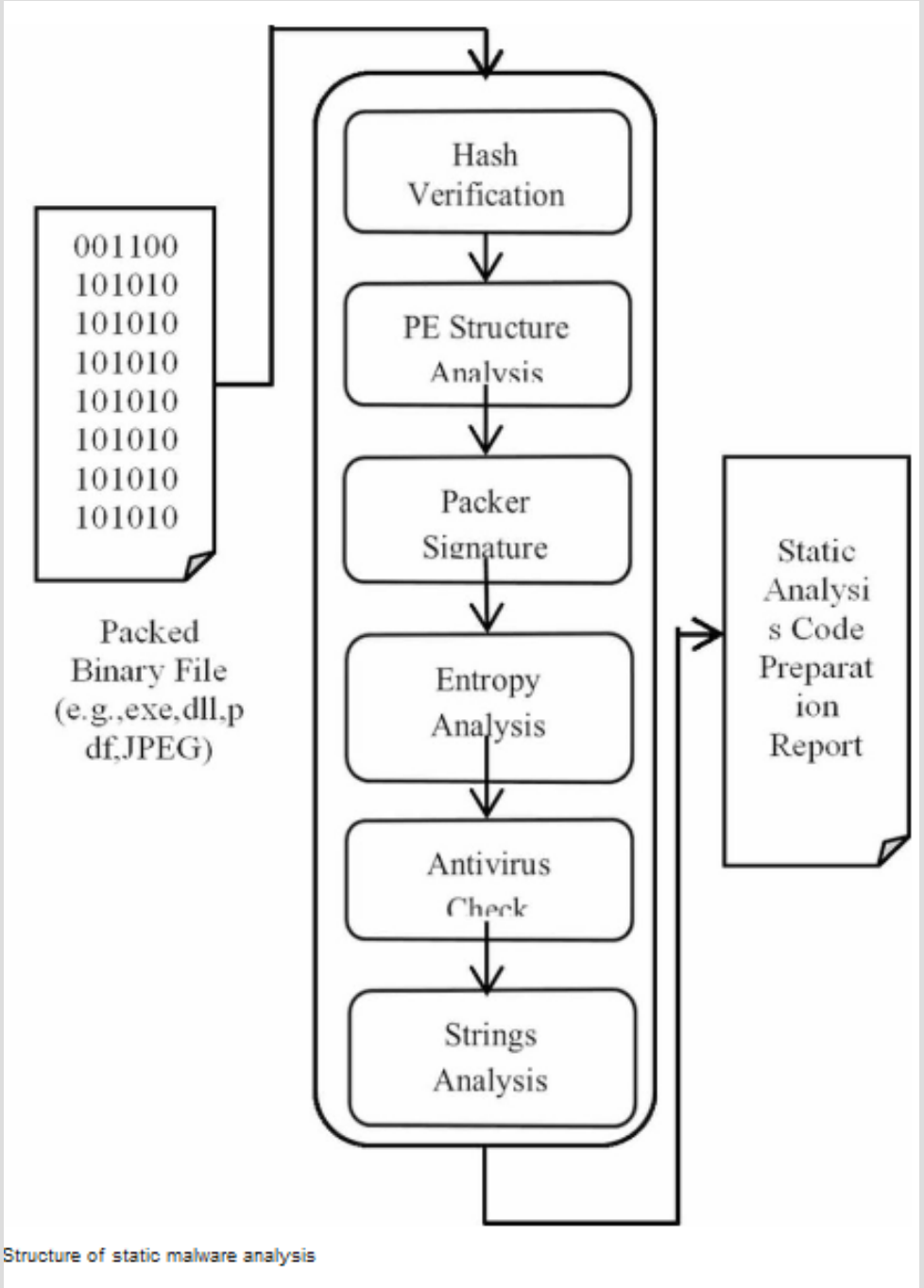


- N'analysez jamais un logiciel malveillant ou un logiciel malveillant présumé sur une machine qui n'a pas pour seul objectif d'analyser un logiciel malveillant.
- Lorsque vous n'analysez pas ou ne déplacez pas les échantillons de logiciels malveillants vers différents endroits, conservez-les toujours dans des archives zip/rar ou autres protégées par un mot de passe afin d'éviter toute détonation accidentelle.
- N'extrayez le logiciel malveillant de cette archive protégée par mot de passe qu'à l'intérieur de l'environnement isolé, et uniquement pour l'analyser.
- Créez une machine virtuelle isolée spécifiquement pour l'analyse du logiciel malveillant, qui peut être remise à zéro une fois que vous avez terminé.
- Assurez-vous que toutes les connexions Internet sont fermées ou au moins surveillées.
- Une fois l'analyse du logiciel malveillant terminée, remettez la machine virtuelle à l'état initial pour la prochaine session d'analyse du logiciel malveillant afin d'éviter que les résidus d'une exécution précédente du logiciel malveillant ne corrompent la prochaine.

Techniques d'analyse de malwares



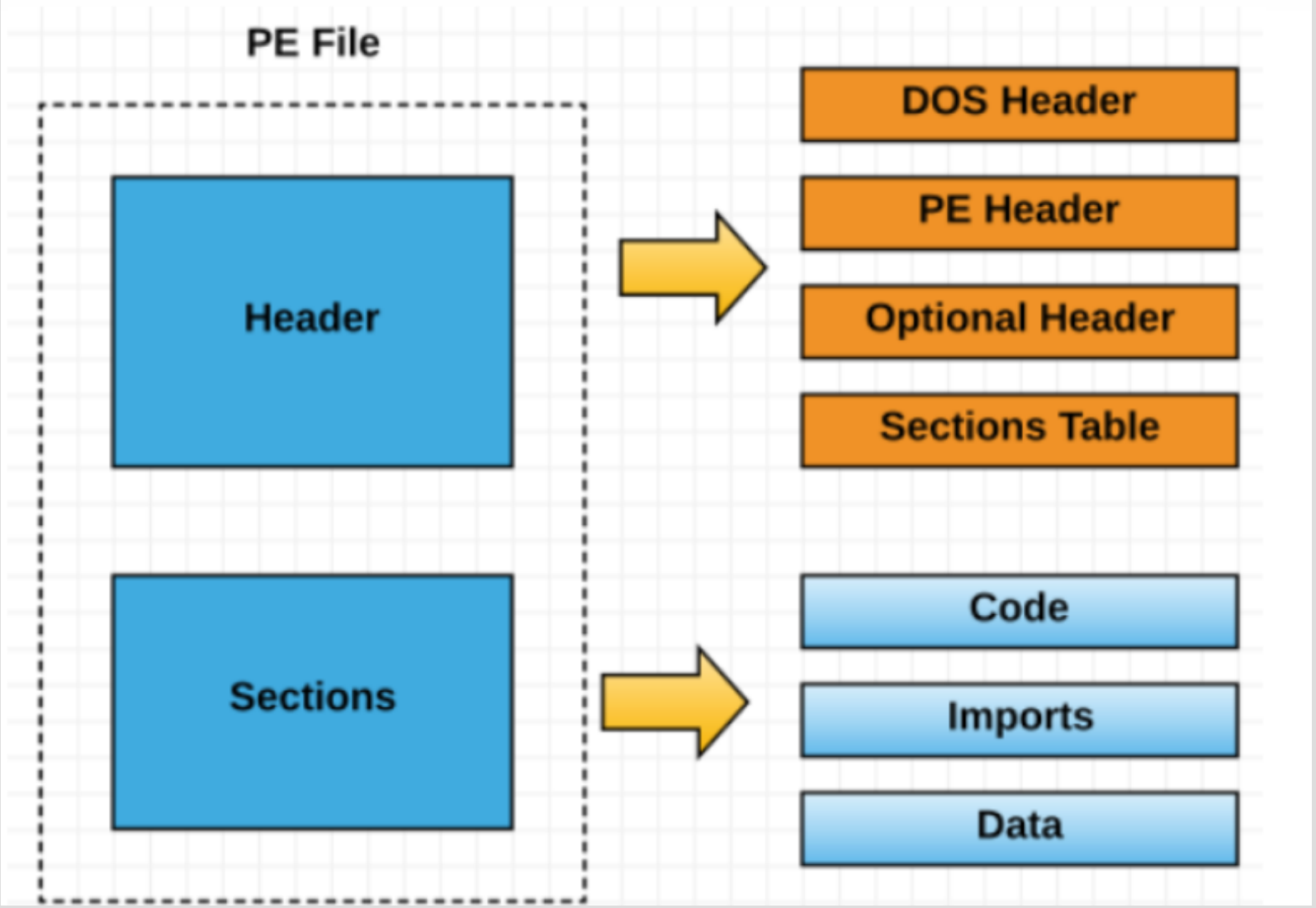
Analyse statique



Structure of static malware analysis

→ Analyse métadonnées + structure de l'en-tête PE ;

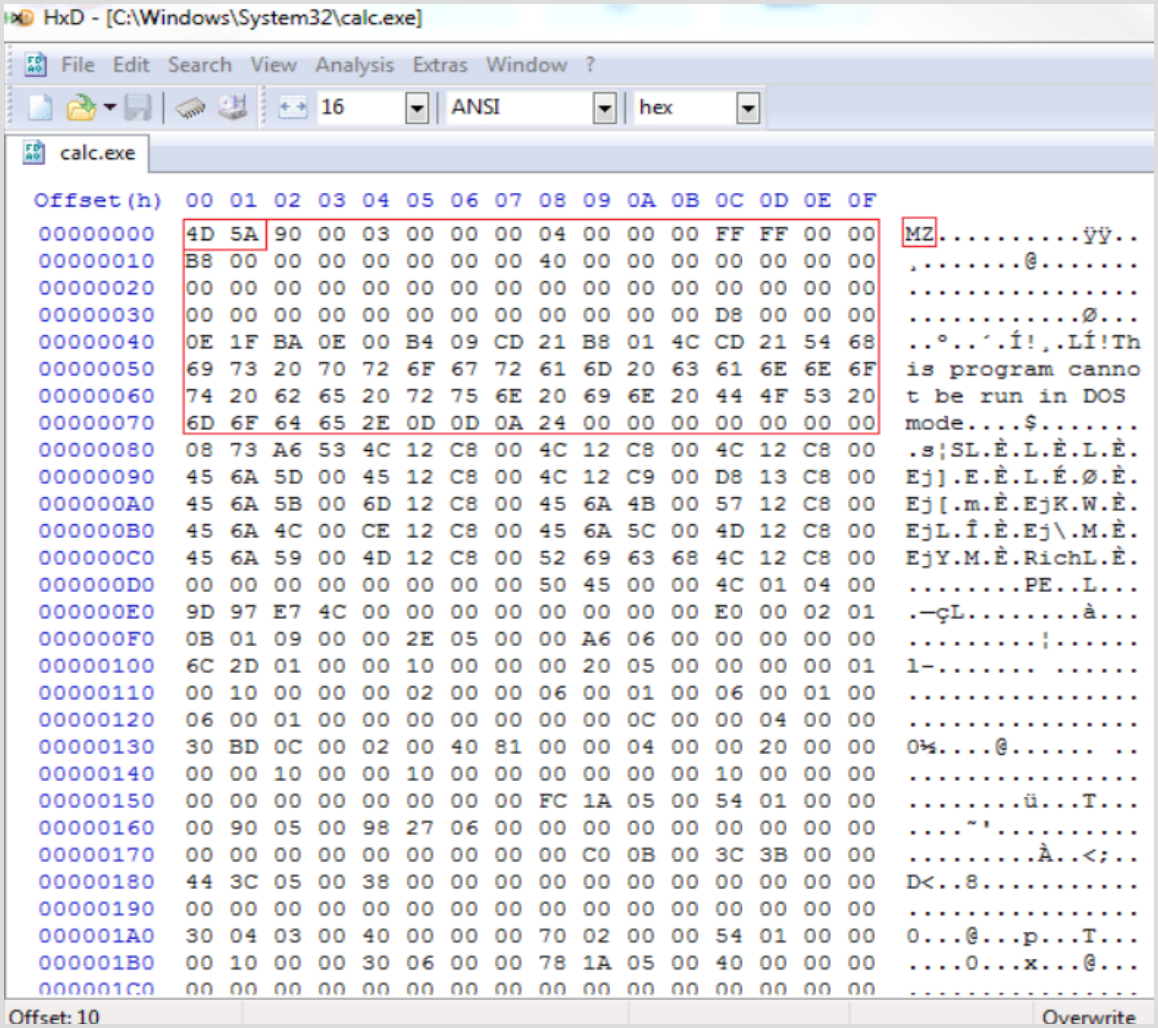
Petit point : format PE



Analyse statique : type de fichier

- File (Linux) : La première étape intervenant dans l'analyse statique va consister à récupérer des informations sur le type de fichier suspect ;

- Cette en-tête située au début de tout fichier Windows permet d'indiquer son format (4D 5A) ;



La sortie montre un fichier exécutable PE32 avec une interface utilisateur graphique, qui a été compilé pour un système fonctionnant sous Microsoft Windows avec un processeur Intel 80386;

Cela signifie que le "80386" dans le résultat ci-dessus nous indique que cette application a été conçue pour les processeurs Intel 32 bits.

```
ubuntu@ip-10-10-36-23:~/Desktop/Samples$ file wannacry
wannacry: PE32 executable (GUI) Intel 80386, for MS Windows
ubuntu@ip-10-10-36-23:~/Desktop/Samples$
```

Analyse statique : chaînes de caractères

Pour aperçu des fonctionnalités utilisées par le malware.

```
,}7z
 inflate 1.1.3 Copyright 1995-1998 Mark Adler
 n;^
Qkkbal
i]Wb
9a&g
MGiI
wn>Jj
#.zf
+o*7
- unzip 0.15 Copyright 1998 Gilles Vollant
CloseHandle
GetExitCodeProcess
TerminateProcess
WaitForSingleObject
CreateProcessA
GlobalFree
GetProcAddress
LoadLibraryA
GlobalAlloc
SetCurrentDirectoryA
GetCurrentDirectoryA
GetComputerNameW
SetFileTime
SetFilePointer
MultiByteToWideChar
GetFileAttributesW
GetFileSizeEx
```

Outils :

- HxD Hex Editor ;
- Strings (Sysinternals Suite) - utilitaire en ligne de commande, exécutable Windows 32bit/64bit ;
- Strings2 - utilitaire en ligne de commande, exécutable Windows 32bit/64bit (https://github.com/glmcdona/strings2) ;
- Flare-Floss (solutionneur de chaînes de caractères obfusquées, https://github.com/mandiant/flare-floss) - combine et automatise différentes techniques afin d'effectuer le décodage de chaînes de caractères.

NB : Les chaînes de caractères sont au format ASCII et Unicode (pour certains outils, il faut spécifier le type de chaîne à extraire lors de l'analyse, car certains outils n'extraient pas les deux formats)

Analyse basique : signature numérique, scans anti-virus et VirusTotal

- https://www.virustotal.com/gui/home/upload
- L'analyse d'un fichier à l'aide d'un antivirus ou la recherche d'un hachage sur VirusTotal peut fournir des informations utiles sur la classification des logiciels malveillants effectuée par les chercheurs en sécurité :

- détails sur l'historique de l'échantillon, la première soumission, la dernière soumission et les métadonnées de l'échantillon (SHA256, MD5, taille du fichier, infos sur la signature, détails de la section, importations, etc.) ;

- Des informations sur le comportement d'un échantillon et de ses relations dans différents environnements en ligne ;

- Des commentaires sur l'échantillon par la communauté sur VirusTotal, qui peuvent parfois fournir un contexte supplémentaire sur l'échantillon.

Analyse statique basique : En-tête du fichier PE : Imports/exports

- Contient les métadonnées d'un fichier Portable Executable :

Date de compilation

Fonctions importées par l'exécutable

Fonctions exportées par l'exécutable

Ressources utilisées (Icon, chaines de caractères, version...)

- Étant donné que la plupart des fichiers PE utilisent l'API Windows pour effectuer la majeure partie de leurs tâches, les importations d'un fichier PE nous fournissent des informations cruciales sur ce que fera ce fichier PE.

→ un fichier PE qui importe la fonction InternetOpen communiquera avec internet, une fonction URLDownloadToFile montre qu'un fichier PE téléchargera quelque chose depuis internet, et ainsi de suite.

En-tête du fichier PE : Sections

Un fichier PE est divisé en différentes sections qui ont des objectifs différents. Bien que les sections d'un fichier PE dépendent du compilateur ou de l'empaqueteur utilisé pour compiler ou empaqueter le binaire, les sections suivantes sont les plus courantes dans un fichier PE.

.text : Cette section contient généralement les instructions du CPU exécutées lors de l'exécution du fichier PE. Cette section est marquée comme exécutable.

.data : Cette section contient les variables globales et autres données globales utilisées par le fichier PE.

.rsrc : Cette section contient les ressources utilisées par le fichier PE, par exemple, les images, les icônes, etc.

Windows :

- PEiD Tool ;
- CFF Explorer ;
- Resource Hacker ;
- PeStudio ;
- IDA free ;
- Ghidra ;

Linux :

- Pecheck ;
- Manalyze ;
- StringSifter ;
- Peframe ;
- PE Tree ;

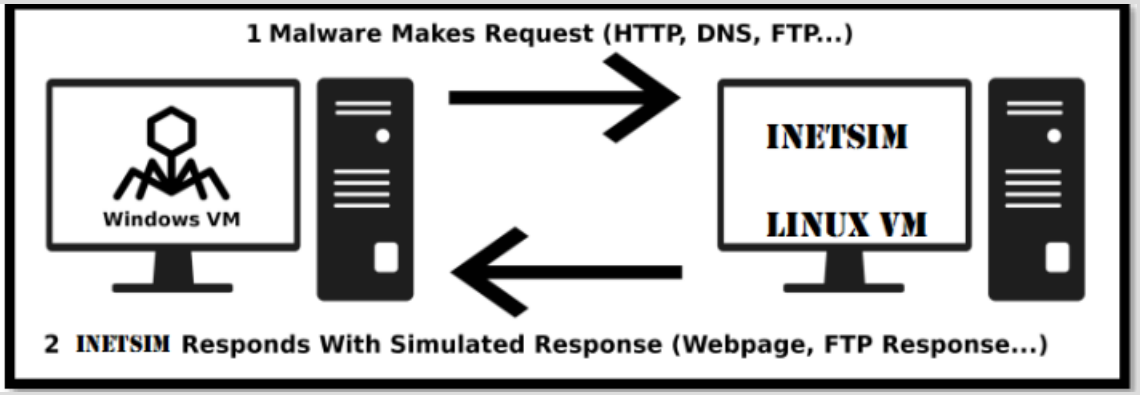
```
ubuntu@ip-10-10-22-38:~/Desktop/Samples$ pecheck wannacry
PE check for 'wannacry':
Entropy: 7.995471 (Min=0.0, Max=8.0)
MD5     hash: 84c82835a5d21bbcf75a61706d8ab549
SHA-1   hash: 5ff465afaabcbf0150d1a3ab2c2e74f3a4426467
SHA-256 hash: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
SHA-512 hash: 90723a50c20ba3643d625595fd6be8dcf88d70ff7f4b4719a88f055d5b3149a4231018ea30d375171507a147e59f73478c0c27948590794554d031e7
.text entropy: 6.404235 (Min=0.0, Max=8.0)
.rdata entropy: 6.663571 (Min=0.0, Max=8.0)
.data entropy: 4.455750 (Min=0.0, Max=8.0)
.rsrc entropy: 7.999868 (Min=0.0, Max=8.0)
Dump Info:
----------DOS_HEADER----------

[IMAGE_DOS_HEADER]
0x0         0x0   e_magic:                      0x5A4D
0x2         0x2   e_cblp:                       0x90
0x4         0x4   e_cp:                         0x3
0x6         0x6   e_crlc:                       0x0
0x8         0x8   e_cparhdr:                    0x4
0xA         0xA   e_minalloc:                   0x0
0xC         0xC   e_maxalloc:                   0xFFFF
0xE         0xE   e_ss:                         0x0
0x10        0x10  e_sp:                         0xB8
0x12        0x12  e_csum:                       0x0
0x14        0x14  e_ip:                         0x0
0x16        0x16  e_cs:                         0x0
0x18        0x18  e_lfarlc:                     0x40
0x1A        0x1A  e_ovno:                       0x0
0x1C        0x1C  e_res:
0x24        0x24  e_oemid:                      0x0
0x26        0x26  e_oeminfo:                    0x0
0x28        0x28  e_res2:
0x3C        0x3C  e_lfanew:                     0xF8
```

analyse dynamique et avancée de malwares : les sandboxes

Pour l'analyse des logiciels malveillants avec des sandboxes, les considérations suivantes rendent l'analyse efficace :

- Une machine virtuelle imitant l'environnement cible réel de l'échantillon de logiciels malveillants.
- Possibilité de prendre des snapshots de la VM et de revenir à un état « propre ».
- Logiciel de surveillance du système d'exploitation (Procmon, ProcExplorer ou Regshot, etc).
- Logiciel de surveillance du réseau (Wireshark, tcpdump, etc).
- Contrôle du réseau par le biais d'un serveur DNS et d'un serveur Web factices (REMnux : INetSim, Fake DNS, ...)



Liste sandboxes :

- Cuckoo's Sandbox ;
- CAPE Sandbox :
- En ligne :

Online Cuckoo Sandbox ;

Online CAPE Sandbox ;

Any.run ;

Intezer ;

Hybrid Analysis

Sources

https://www.youtube.com/watch?v=BMFCdAGxVN4

https://www.avira.com/en/blog/malware-threat-report-q2-2020-statistics-and-trends

https://www.atomicmatryoshka.com/post/what-is-fuzzy-hashing

https://www.sans.org/blog/how-you-can-start-learning-malware-analysis/

https://zeltser.com/malware-analysis-cheat-sheet/

https://0xinfection.github.io/reversing/

https://www.youtube.com/playlist?list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io

https://ccdcoe.org/uploads/2020/07/Malware_Reverse_Engineering_Handbook.pdf

https://fsec404.github.io/blog/Shanon-entropy/

https://www.microsoft.com/en-us/security/blog/2021/07/27/combing-through-the-fuzz-using-fuzzy-hashing-and-deep-l

https://whiteheart0.medium.com/dynamic-malware-analysis-lab-setup-613075f9423f

https://www.inetsim.org/features.html

https://tryhackme.com/room/intromalwareanalysis

https://tryhackme.com/room/malmalintroductory

https://www.securiteinfo.com/attaques/malwares-virus-spam-logiciels-indesirables/techniques-detection-malware.shtml