Robustesse des outils de sécurité

Monnet-Paquet Aurélien
·
Supervised by : Lauradoux Cédric

Juin 2016

Abstract

Keywords Bombe de compression · IDS · Antivirus · Framework

1 Introduction

2 Outils de sécurité

2.1 Antivirus

Les antivirus (AV) sont les logiciels de sécurité les plus utilisés par les particuliers. Ce sont des programmes installer sur les machines des utilisateurs. Les antivirus sont lancés avant l'initialisation du système de fichiers et du réseau par l'OS. Cet outil permet de détecter les malwares et les empêche de s'exécuter sur le système. Il existe différentes manières de détecter un malware :

- Scan par signature : L'AV va calculer la signature d'un fichier ou d'un morceau de code et va le comparer a une base de données. Méthode inefficace contre les malwares polymorphes ou capable de changer leur signature.
- Analyse heuristique : Méthode la plus puissante car elle permet de simuler l'exécution du code d'un programme dans une zone contrôlée. Ainsi, l'AV peut observer le comportement du code qui s'exécute et définir si il s'agit d'un malware ou non. Peut provoquer des fausses alertes.

Monnet-Paquet Aurelien
Université Grenoble Alpes, 38400 Saint Martin d'Hères
E-mail: monnetpa@e.ujf-grenoble.fr

Lauradoux Cedric
Inria, 38334 Montbonnot
E-mail: cedric.lauradoux@inria.fr

- Controle d'integrité : Méthode qui permet de vérifier qu'un fichier n'a pas été modifié au cours du temps. Les informations comme la taille, la date et l'heure de dernière modifications, la somme de contrôle éventuelle du fichier sont analysées lors de la demande d'ouverture du fichier par l'utilisateur (si analyse en temps réel) ou lors d'un scan de l'AV.
  Le but de notre expérience est de tester si il est possible de faire un deni de service lorsque l'AV analyse une bombe de compression.
  Nous avons tester deux AV pour Linux : ClamAV (`www.clamav.net`) et Comodo (`www.comodo.com`)

2.2 Agrégateurs d'antivirus

Les agregateurs d'AV sont des outils qui permettent d'analyser des fichiers avec differents AV en meme temps. Le résultat obtenu en analysant un fichier avec ce type d'outils est plus fiable qu'avec un seul AV. Cependant, cet outil ne doit pas remplacer un AV installer sur une machine. Il existe deux types d'agregateurs :

- Les sites web :
  - VirusTotal : `www.virustotal.com`, filière de Google, basée en Espagne, hébergé dans le cloud, comprenant 57 AV.
  - Jotti : `virusscan.jotti.org`, organisation basée aux Pays-Bas, hébergé dans le cloud, comprenant 19 AV.
  - Virscan : `www.virscan.org`, organisation basée en chine, comprenant 39 AV.

Pour utiliser ce service, il faut envoyer le fichier suspect vers le site pour que les moteurs des differents AV puisse l'analyser. Ce fichier n'est ensuite pas detruit, car il sert aux editeurs d'AV pour améliorer leurs programmes. Il n'est donc pas concevable pour une entreprise d'envoyer des fichiers avec des données sensibles vers des serveurs dont l'entreprise n'a pas de contrôle. C'est pourquoi il existe aussi

- les frameworks d'analyse de malwares :
  - Mastiff, `www.korelogic.com`,
  - Viper, `www.viper.li`,

2.2.1 Sites web

2.2.2 Frameworks

2.3 IDS

3 Tests

3.1 Protocole de test

Dans un premier temps, nous avons générer un fichier (payload via metasploit) connu pour être analyser comme un malware par la plupart des outils disponible sur le marché. Ensuite, pour chacun des outils, nous avons effectuer une analyse témoin. Cette analyse se compose du fichier générer précédemment ainsi que d'un fichier contenant uniquement des zéros et qui est donc complètement inoffensif.
Dans un second temps, nous avons analyser les résultats de ses deux fichiers compressé dans différents formats : .gz, .lzma, .tar.bz2, .tar.gz, .tar.xz, .lz, .lzo, .zip
Et enfin, nous avons analyser l'impact des bombes de compression sur ses outils pour vérifier leur robustesse sur un éventuel déni de service.
Pour rappel : lorsqu'un fichier compressé de petite taille (42 Ko) se décompresse en un autre fichier d'une taille très largement supérieur (4.5 Po), c'est ce qu'on appelle une bombe de compression (42.zip).
Dans notre expérience, les fichiers décompressés sont de taille : 250 Mo, 500 Mo, 1 Go, 4.5 Po.
Nous avons réaliser les tests (et installer les outils nécessaires) sur une machine ayant un OS Linux (Ubuntu 12.04 LTS).
Après avoir installer ClamAV (via les dépôts Ubuntu/Debian) nous avons effectuer des tests dans sa configuration initiale. Pour lancer un scan dans le répertoire courant, il suffit de la commande "clamscan" (avec la configuration initiale). Cependant, deux paramètres sont à prendre en compte pour menée à bien cette expérience :

- "max-recursion= n" : La récursion lors de l'analyse des fichiers compressés, 16 par défaut.
- "max-filesize= n" : Extrait et analyse n octets de chaque archives. 25 Mo par défaut avec une limite de 4 Go.

Nous avons également installer un second antivirus : Comodo (via package du site officiel [**?**]). Il dispose d'une interface graphique, mais ne nous permet pas de modifier des paramètres comme la récursion, ou la taille des fichiers analysés. Pour lancer un scan, il faut suivre les indications sur l'interface graphique.

VirusTotal met à notre disposition une API permettant d'envoyer des fichiers

pour analyse via un script. Nous avons alors repris puis modifier un script existant pour effectuer nos tests. Ce script (écrit en Perl) effectue deux requêtes vers VirusTotal. Une première (HTTP POST) pour envoyer le fichier suspect. Et une seconde, pour récupérer le résultat sous forme d'un objet JSON. Ensuite le script génère un rapport mis en forme. Ce rapport contient le résultat de tous les antivirus associés au fichier envoyé.

En ce qui concerne Jotti et Virscan, nous avons utilisé l'interface web pour soumettre nos fichiers et récupérer les résultats.

Mastiff est un processus que l'on peut exécuter dans un docker. Dans ce cas, un dossier est partagée entre la VM et le docker. Dans ce dossier, nous avons au préalable placé les fichiers à analyser par Mastiff. Dans le cas présent, nous n'avons pas de fichier confidentiel, c'est pourquoi nous avons ajouter l'option d'envoyer les fichiers à analyser vers VirusTotal automatiquement. Lors de son analyse, Mastiff génère un fichier de résultat par moteur d'analyse.

Viper est aussi un processus que l'on peut lancer dans un docker. A la différence que Viper met à notre disposition une interface web, à partir de laquelle nous pouvons soumettre nos fichiers. Une fois un fichier soumis, une multitude de commande peuvent être lancer pour tester le fichier suspect. Les résultats de ses commandes se retrouvent sur cette interface web.

Après avoir installer Suricata, nous avons rédiger des règles ([4]) pour que les fichiers soit décompressés puis inspectés. Nous avons pris soin de modifier la configuration initiale de Suricata pour que l'extraction se passe de manière optimale :

- stream.reassembly.depth = 4gb, après avoir ré-assembler le flux TCP, le fichier ne doit pas dépasser une taille de 4Go.
- request_body_limit = 0 (infini), valeur que le corps de la requête HTTP ne peut pas dépasser.
- response_body_limit = 0 (infini), valeur que le corps de la réponse HTTP ne peut pas dépasser.

De plus, sur une autre machine nous avons installer un serveur web disposant des fichiers à tester par Suricata. Ainsi, avec une simple requête sur le serveur, Suricata reconstitue le fichier puis l'inspecte.

3.2 Observations et résultats

4 Conclusion

References

1. David J. Day et Benjamin M. Burns, "A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines"
2. oisf.net, The Open Information Security Foundation, organisation à but non lucrative qui développe et met à jour Suricata.
3. redmine.openinfosecfoundation.org/projects/suricata/wiki/, la documentation pour utilisateurs et développeurs de Suricata.
4. `blog.inliniac.net/2011/11/29/file-extraction-in-suricata/`
5. `perlgems.blogspot.fr/2012/05/using-virustotal-api-v20.html`
6. irma.quarkslab.com