Je suis pleinement conscient(e) que le plagiat de documents ou d'une partie de document constitue une fraude caractérisée.
Nom, date et signature :

Robustesse des outils de sécurité

Monnet-Paquet Aurélien
·
Supervised by : Lauradoux Cédric

Juin 2016

Abstract

Keywords Bombe de compression · IDS · VirusTotal · Antivirus · Framework de détection de malwares

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

1 Introduction

2 Outils de sécurité

2.1 Antivirus

2.2 Agrégateurs

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.
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).

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 [12]). 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 et Viper

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, apres avoir réassembler le flux tcp, le fichier ne doit pas depasser 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.

3.2 Résultats et observations

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. virustotal.com, site agrégateurs d'antivirus, filière de Google, basé en espagne.
6. virusscan.jotti.org, site agrégateurs d'antivirus, basé aux Pays-Bas.
7. www.virscan.org, site agrégateurs d'antivirus, basé en chine.
8. `perlgems.blogspot.fr/2012/05/using-virustotal-api-v20.html`
9. www.korelogic.com
10. www.viper.li
11. irma.quarkslab.com
12. www.comodo.com
13. www.clamav.net