Prise d'empreinte passive de trafic réseau pour l'aide à la configuration de systèmes de supervision réseau

Pierre Chifflier, Gilles Guette, Johan Mazel

pierre.chifflier@ssi.gouv.fr, johan.mazel@ssi.gouv.fr
ANSSI – Paris, France

gilles.guette@univ-rennes1.fr
Équipe CIDRE (Inria / CentraleSupélec / Univ. Rennes 1 / CNRS)
Rennes, France

1 Contexte

Les protocoles IP et TCP utilisent des mécanismes spécifiques comme la fragmentation et la segmentation afin de transmettre des quantités arbitraires de données tout en palliant les contraintes du medium physique ou de la couche protocolaire inférieure. Les systèmes d'exploitation utilisent des politiques variées afin de remédier aux ambiguités de recouvrement liées à ces deux mécanismes.

Un des enjeux de sécurité informatique est la manière dont les équipements de détection tels que les IDS gèrent les différentes politiques de ré-assemblage implémentées dans les entités protocolaire en bout de chemin. En effet, si les équipements de détection ou de prévention d'intrusion ne reconstruisent pas les flux d'octets de la même manière que les entités protocolaires en bout de chemin, il devient possible pour un attaquant de se rendre invisible (attaque par évasion) ou d'insérer des données dans le flux visible par l'équipement de détection (attaque par insertion). Les systèmes de détection ou de prévention d'intrusion (IDS/IPS) doivent ainsi tenir compte de ces politiques pour reconstruire un flux d'octets cohérent avec ce que les applications observent en bout de chemin [1–3]. Les systèmes de détection ou de protection implémentent donc les politiques de piles réseaux connues, et s'appuient pour cela sur un fichier de configuration qui associe une politique de reconstruction à chaque machine supervisée (par exemple `host-os-policy` dans Suricata [4]).

Cette configuration manuelle est compliquée et fastidieuse car elle requiert d'identifier toute les machines d'un parc ainsi que les versions d'OS. Cette configuration requiert également une mise à jour constante afin de suivre l'évolution du parc supervisé.

Or, il existe des approches pour identifier les OS de machines présentes sur un réseau : la prise d'empreinte. Cette pris d'empreinte peut s'opérer de manière passive (ex. p0f [5] ou nPrintML [6]) ou active (ex. Herschel [7]). L'approche peut-être basée sur des signatures [5, 7], ou utiliser des techniques d'apprentissage automatique pour construire un classifieur [6]. Ces signatures sont aussi parfois non-maintenues (i.e. la dernière version de p0f a été publiée en 2014). De plus, il n'existe pas de publication qui fournisse une comparaison entre ces approches.

2 Objectifs

Le but de ce stage est de définir les différentes étapes de trouver les solutions menant à l'implémentation d'un outil de prise d'empreinte de traffic réseau pour l'aide à la configuration d'IDS. Cette approche est similaire à ce qui est proposé dans [8].

- Le premier objectif du stage est d'identifier les OS dont des politiques de reconstruction de flux sont ciblés par les IDS (par exemple Snort ou Suricata). Ce travail s'appuiera sur les documentations des IDS existants (ex. [4]) et les résultats de travaux en cours des encadrants du stage.

- Le deuxième objectif est l'analyse des approches existantes en terme de prise d'empreinte passive et active de trafic réseau pour la classification d'OS.

  Des travaux sur la prise d'empreinte active [7] ont montré de bonnes performances sur un large panel d'OS. Ces travaux ont aussi décrit des situations pour lesquels il nous semble que les approches passives existantes sont faillibles.

  Les sous-objectifs seront donc les suivants : 1) l'analyse du comportement des approches passives existantes quant à ces situations, 2) l'extension éventuelle des approches passives existantes pour corriger les problèmes identifiés, et 3) la réalisation d'une comparaison de performances entre les méthodes passives et actives.

  Il sera porté une attention particulière à la capacité de la méthode de prise d'empreinte à discerner des OS ou versions d'OS dont les politiques sont différentes.

- Le dernier objectif est la génération automatique de configuration pour les IDS. Cette étape s'appuiera sur les besoins des IDS documentés lors de la réalisation du premier objectif et des capacités de classification construites au cours de l'accomplissement du deuxième objectif. Dans le cas où la prise d'empreinte n'est pas suffisamment fiable (i.e. certains OS et/ou certaines versions d'OS ne peuvent pas être différenciés), il faudra être capable de fournir cette information à l'utilisateur et de la motiver. En particulier, si la solution retenue lors de l'accomplissement du deuxième objectif utilise de l'apprentissage, il faudra pouvoir interpréter les modèles utilisés.

3 Compétence requises

Le candidat, présentant des compétences solides en informatique, doit maîtriser au moins un langage de programmation (idéalement C et/ou Python) et avoir des connaissances raisonnables sur le fonctionnement des réseaux TCP/IP.

Une spécialisation en sécurité ou en réseaux informatiques serait appréciée.

Des compétences en virtualisation (Vagrant/VirtualBox) seraient un plus.

4 Logistique

Le stagiaire sera sous convention avec Inria et rattaché à l'équipe CIDRE. En fonction des conditions, le stage pourra se dérouler sur les deux sites rennais de CIDRE (IRISA et CentraleSupélec), ou bien en majeure partie à distance.

- Durée du stage : 5/6 mois.
- Rémunération : gratification de stage.
- Niveau : M2, troisième année d'école d'ingénieurs ou équivalent.

References

[1] U. Shankar and V. Paxson, "Active mapping: Resisting nids evasion without altering traffic," in *2003 Symposium on Security and Privacy, 2003.* IEEE, 2003, pp. 44–61.

[2] J. Novak, "Target-based fragmentation reassembly," 2005.

[3] J. Novak and S. Sturges, "Target-based tcp stream reassembly," 2007.

[4] OISF. (2020) Suricata documentation on host-os-policy. [Online]. Available: https://suricata.readthedocs.io/en/suricata-4.0.2/configuration/suricata-yaml.html#host-os-policy

[5] M. Zalewski. (2014) p0f. [Online]. Available: https://lcamtuf.coredump.cx/p0f3/

[6] J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "nprint: A standard data representation for network traffic analysis," *arXiv preprint arXiv:2008.02695*, 2020.

[7] Z. Shamsi, A. Nandwani, D. Leonard, and D. Loguinov, "Hershel: Single-packet os fingerprinting," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2196–2209, 2015.

[8] G. Taleck, "Ambiguity resolution via passive os fingerprinting," in *International Workshop on Recent Advances in Intrusion Detection.* Springer, 2003, pp. 192–206.