Projet : implémentation d'un IPS

30 mars 2020

Certains d'entre vous ont eu l'occasion de jouer avec Scapy afin d'implémenter un protocole. Je vous propose d'aller plus loin et d'implémenter une solution IPS en python via Scapy en travail individuel ou binôme.
Pour ceux qui n'ont pas encore rencontré cette librairie, voici sa documentation
-> https://scapy.readthedocs.io/en/latest/

1 Objectif de votre application

Du fait du peu de temps que vous aurez, votre application sera une version simplifiée et testée. Votre IPS doit pouvoir permettre de protéger un réseau d'une attaque (une si vous êtes en individuel, 2 attaques si vous êtes en binôme) que vous allez sélectionner. Cela peut-être par exemple :

— Une attaque sur TCP
— Une attaque sur ARP
— Une attaque sur HTTP
— Un scan NMAP ?
— Une attaque sur le protocole flipper ?
— Une autre attaque ?

Votre programme doit détecter la ou les attaques et réagir en conséquence.

Rappel :
Le rôle d'un IPS est de détecter via des patterns (donc ici un ou deux) une tentative de contournement ou d'attaque sur un protocole particulier. Ensuite une fois la détection effectuée, il y a une réaction, soit par rejet/log simple soit par un paquet particulier en réponse. Par exemple, dans le cas d'un paquet http, l'envoi d'une réponse 403 forbidden.

2 Rendu

Vous devrez me fournir votre code source accompagnées d'un fichier readme pour son lancement ainsi qu'un fichier contenant une liste des paquets ou librairies nécessaires. Il faudra également un rapport montrant une démonstration de votre outil sur un scénario, une explication de la ou des attaques qui seront évitées ainsi qu'une mesure des impacts avec et sans votre application.

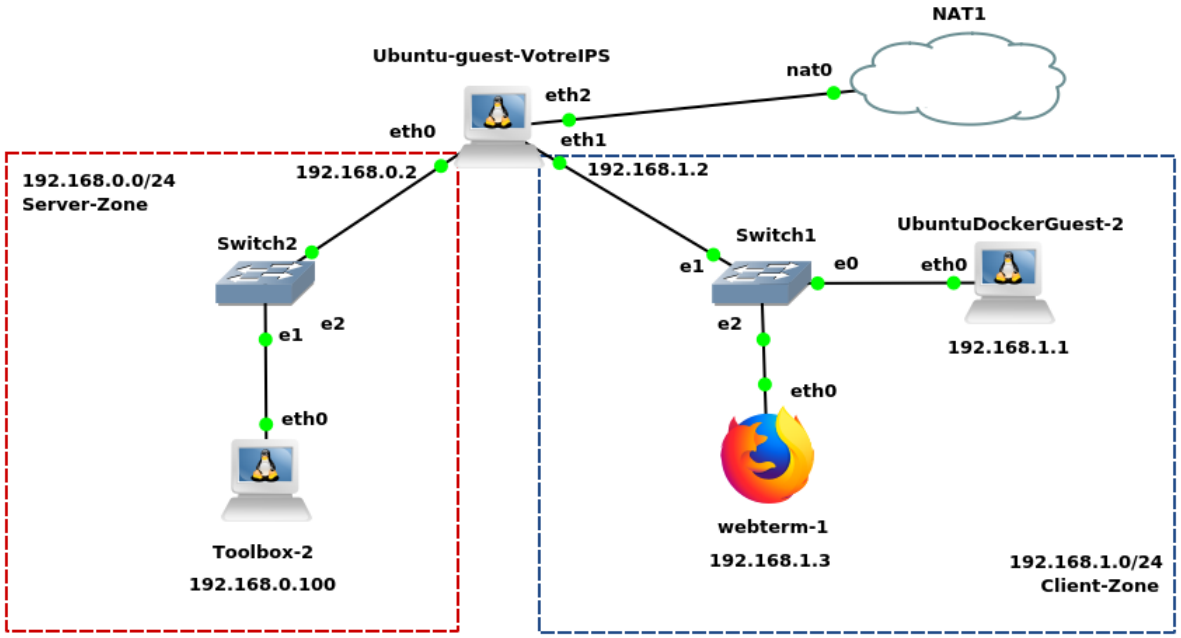
Exemple de mesure d'impact avec le cas d'un blocage de ping (Bien sûr, l'exemple ne peut pas être choisi... trop facile) :
Mon application a pour rôle d'empêcher un réseau de pinger un équipement d'un autre réseau. Sans mon application, les réponses permettrait à un pirate d'avoir une vision de la cartographie du réseau distant. Mon test consistait à envoyer 100 ping provenant d'une machine d'un réseau vers un autre réseau distant et de voir le résultat. 80 ping sur 100 ont été détecté et rejeté en réponse. Il a été noté que mon programme rajoute une latence supplémentaire moyenne de 3ms.

Dans cet exemple, j'ai mis deux métriques, une pour l'efficacité de mon application, une autre pour un effet de bord de mon système. Dans le rapport, il me faudra au minimum une seule métrique de validation par scénario.

Le rapport peut être fait sous forme d'un document pdf avec des screenshots et des explications ou sous forme de vidéo de démonstration voire de diaporama. Du moment que les informations sont présentes et que j'ai une vision de l'action de votre application ainsi que son impact, cela m'ira très bien. Dans votre rapport vous pouvez également expliquer si un élément de votre programme ne fonctionne pas et ce que vous avez essayé de faire pour y remédier (cela m'évitera de passer une heure à tester...)

3 Architecture

Pour vous faire gagner du temps, j'ai réalisé une petite architecture de départ sur GNS3 que vous pourrez utiliser afin que vous puissiez vous concentrer les parties développement et validation. Il suffit de l'importer via file > import portable project.



Voici une description des différents éléments :

— NAT1 : C'est votre accès Internet, normalement cet élément fonctionne comme une interface réseau en NAT sur virtualbox, elle diffuse un DHCP et redirige tous les paquets

— Ubuntu-guest-VotreIPS : C'est sur cette machine que vous pourrez faire tourner votre application. Il y aura une configuration à faire afin de garantir la bonne redirection des paquets (voir le fichier de config dans les ressources du TP). Ceci est en fait un container docker contenant un ubuntu minimale en ligne de commande.

— Toolbox-2 : Cette machine contient un serveur web, ftp, DHCP, syslog, snmp... C'est une boite à outil afin de vous permettre de choisir les protocoles qui seront utilisés dans votre démonstration (joignable sur 192.168.0.100 via un navigateur ou curl).

— Deux machines dans la zone client : Une machine en ligne de commande et une autre via un simple navigateur, l'idée c'est de vous servir de l'une ou l'autre pour lancer votre attaque (si c'est une attaque interne bien sûr...)
— Les switchs peuvent être configuré en mode VLAN mais rien de plus

4 Votre marge de manœuvre

Quelques conseils et recommandations :

— **L'architecture n'est pas figée** : c'est juste une proposition mais vous pouvez la modifier autant que vous voulez du moment que c'est la machine ubuntu-guest-VotreIPS qui utilise votre application. Vous pouvez par exemple remplacer le serveur web par une autre machine sur laquelle vous feriez tourner une api flask... ou un site web que vous aviez programmé dans vos TPs précédents.

— **Vous n'êtes pas seul** : L'utilisation de Scapy n'est pas forcément évidente, vous pouvez donc vous entraider entre groupes ou personnes. Ce que je ne veux pas voir c'est une copie conforme du travail d'un autre étudiant ou d'un autre groupe car cela ne me permet d'évaluer vos compétences. Mis à part cela, vous pouvez par exemple travaillez à plusieurs sur le même protocole puis ensuite chacun peut orienter son programme sur une attaque différente sur le même protocole.

— **Soyez inventifs** : Le choix de l'attaque doit être votre choix et non celui du camarade d'à côté, il y a un nombre considérable de cas possibles. Pour l'exemple que j'ai donné, mon attaque portait sur un champ très large et il vous est donc possible de le moduler pour rendre l'attaque ou la réponse différente. Certes vous pouvez vous aider des exemples du cours ou chercher sur internet mais vous pouvez aussi réfléchir par vous-même au protocole que vous souhaitez protéger et comment vous pourriez le faire. Votre programme pourrait être par exemple un squelette brute avec des fichiers de configuration pour les différents pattern.

— **Variez votre mode de communication** : Votre rapport peut-être fait par exemple sous forme de vidéo ou de diaporama, cela peut prendre bien moins de temps à réaliser et être tout aussi clair (je demande juste une démonstration avec explication et métriques). Le dépôt maximal sur moodle est de 100Mo, mais au pire si ce n'est pas assez envoyez-moi un mail et on trouvera une solution acceptable.

— **Vous souhaitez utiliser un autre outil ?** : Tous les outils que je propose me paraissent intéressants car simples et pertinents en terme de pédagogie. Cependant, je suis ouvert à toute proposition argumentée.

— **GNS3 ne marche pas** : En cas de problème technique avec GNS3, n'hésitez pas à me poser des questions, c'est très important que vous perdiez le moins de temps possible sur la partie intégration. En plus, j'y ai passé pas mal de temps donc il y a de grandes chances que j'aie eu le même problème. N'hésitez pas à exporter votre architecture et me l'envoyer dans le cas où elle serait différente de la mienne.

— **Utilisez les séances programmées sur l'EDT** : cela vous donne un cadre pour dire "je commence" puis "j'arrête". Je laisse souvent des grands délais mais ça c'est pour vous permettre de peaufiner ce que vous avez fait lors des séances programmées, pas pour que vous passiez tout un weekend dessus.

5 Ressources

Vous avez en ressources :

— le travail réalisé en TD

— un fichier d'architecture à importer (attention, si vous êtes sous windows, il faudra sans doute utiliser la machine virtuelle gns3 via virtualbox à télécharger sur le site officiel de GNS3). L'importation ne fonctionnera pas si votre machine virtuelle n'est pas fonctionnelle.

— Un fichier "Config-UbuntuGuest-VotreIPS" contenant des commandes à utiliser sur votre machine pour la configurer car malheureusement l'architecture exportée ne les a pas mémorisé mais c'est assez simple à suivre.