REPUBLIQUE DU BENIN

*****

UNIVERSITE D'ABOMEY-CALAVI (UAC)

*****

ECOLE POLYTECHNIQUE D'ABOMEY CALAVI (EPAC)

*****

GÉNIE INFORMATIQUE ET TÉLÉCOMMUNICATIONS

(4e Année- RII 1)

*****

Matière : Système d'Exploitation Avancé (SEA)

*****

RAPPORT DE RECHERCHES SUR NETKAT

Réalisé par le groupe 3 :

1- HOUENOU Emmanuel

2- TOBOU Charmel

3- YAYA NADJO Sènanmie

Sous la supervision du :

Dr Emery ASSOGBA

Année Académique 2021-2022

PLAN

INTRODUCTION

Iptable est un pare-feu logiciel pour les distributions Linux. Cela inclut des exemples iptables d'autorisation et de blocage de divers services par port, interface réseau et adresse IP source. Dans la pratique, certains environnements de déploiement, ont beaucoup d'iptables ce qui devient assez contraignant. De ce fait, des études ont abouti à des méthodes de filtrage sur ces environnements. C'est au nombre de ceux-ci que figure ***netkat***, fonctionnant avec le principe de base du **netcat** d'origine. Par la suite, nous explorons l'univers de netkat, cela dit, d'abord pourquoi avoir pensé à netkat ensuite comment il fonctionne ainsi que ses limites.

I. VUE D'ENSEMBLE

Depuis le temps vous aurez sans doute remarqué que GNU/Linux fournit à peu près tous les outils nécessaires à la communication entre machines. D'ailleurs pour ce billet, on vous propose de découvrir l'un de ces utilitaires que l'on a tendance à oublier et qui, pourtant, rend fiers les services de transmission. Il s'agit de netcat, abrégé en *nc*.

Son installation est des plus simples. On doit installer le package *nmap-ncat* qui se trouve dans le dépôt de base de la distribution Linux. Le binaire se trouve placé dans /usr/bin.

Cet outil est disponible de façon générale sur les systèmes Unix, Windows, et MacOS X. Sa fonctionnalité principale consiste à ouvrir les connexions réseau localement sur la machine tout en envoyant les paquets de façon synchrone (paquets TCP) ou asynchrone (paquets UDP).

En fait, il gère les sockets (ou connexions réseau), en établissant n'importe quelle communication vers un serveur distant, en choisissant l'adresse IP et le port et en ouvrant ce dernier en écoute de paquets entrants et/ou sortants.

Netkat, sur lequel se porte la présente exposition, se base sur les principes de bases de netcat, pour essayer de résoudre certains problèmes au niveau du noyau, pour les distributions linux, en se basant sur la technologie **eBPF**. Notons aussi qu'un langage nommé netkat a vu le jour, pour permettre une meilleure expérience avec la technique netkat.

II. OBJECTIFS

- Faire une différence entre netkat et netcat
- Comprendre aisément comment utiliser netkat

III. QU'EST-CE QUE NETKAT ?

Netkat, comme nous le disons plutôt, est une technologie basée sur netcat, mais utilisant des sockets bruts pour éviter les iptables et/ou d'autres mécanismes de filtrage du système d'exploitation. Les environnements Kubernetes ont beaucoup d'iptables, d'où la nécessité de contourner parfois iptables, pour les tests, le débogage, le dépannage, etc.

Netkat utilise la pile TCP/IP en créant un **socket RAW** attaché à l'interface avec la route vers l'adresse IP de destination, attachant un **filtre eBPF** pour le trafic en calculant les paramètres de connexion spécifiés comme arguments.
Il crée également une file d'attente d'entrée et attache un filtre eBPF, pour supprimer le trafic de la connexion dirigé vers le socket RAW, empêchant ainsi la pile hôte de supprimer la connexion.

IV. POURQUOI LE NETKAT ?

Tout d'abord, il permet d'étendre les fonctions du netcat en les orientant vers le kernel. Ensuite, il constitue une méthode eBPF, ce qui permet d'agrandir les techniques de filtrage eBPF.

V. PRÉSENTATION DE NETKAT

Netkat fonctionne comme le netcat d'origine, mais en utilisant des sockets bruts, il n'est donc pas affecté par les crochets **netfilter**. Un socket RAW reçoit une copie de tous les paquets, cependant, nous n'avons pas besoin de tous, seulement de ceux utilisés par la connexion netkat. Sur le socket, nous filtrons les paquets que nous ne voulons pas.

Netkat peut obtenir le tuple de la connexion à l'avance, il peut donc filtrer à l'aide de eBFP et de l'option socket *SO_ATTACH_FILTER*, ainsi que de la capacité de filtrage eBPF de contrôle du trafic.

Les paquets reçus sur le socket sont des paquets bruts (avec des en-têtes Ethernet), mais comme ils contournent la pile TCP/IP du noyau, quelque chose doit

rassembler les paquets et obtenir un flux de données. Ceci est réalisé en utilisant une pile TCP/IP en espace utilisateur. À haut niveau, le code fait :

- Obtenir les détails de la connexion : IP port source, destination, et interface hôte
- Paramétrer le code eBPF avec les détails de connexion, générer le code eBPF
- Injecter le code eBPF dans le socket brut et dans l'interface d'entrée

En conséquence, nous avons une nouvelle version de netcat qui contourne les iptables.

La figure ci-dessous présente le fonctionnement du netkat lors d'une connexion réseau.

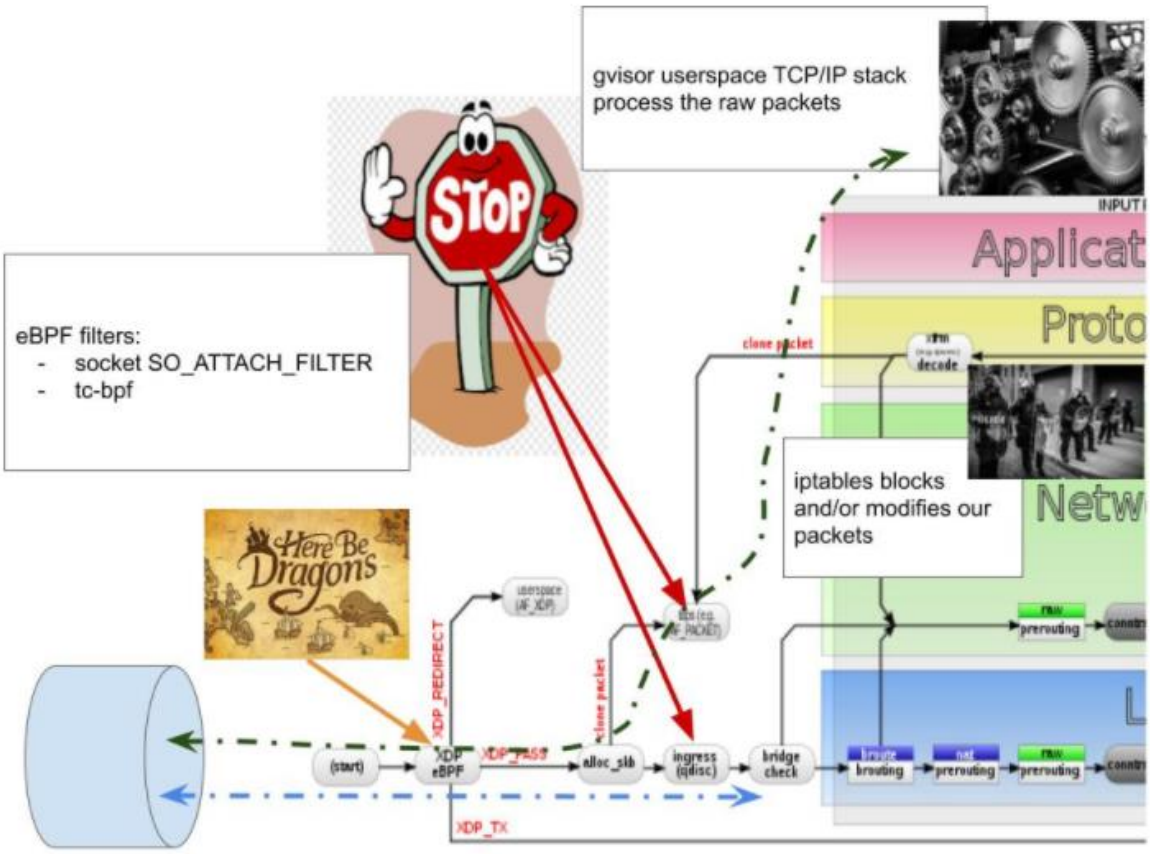

Figure : Présentation du fonctionnement de netkat.

VI. LIMITES DE NETKAT

Comme toutes techniques de filtrage eBPF, elle est en voie de développement et nécessite donc de plus de tests pour la découverte de nouvelles issues, failles.

La confusion entre son langage sémantique dédié, est due au manque de documents lié à cette branche. En effet, pour faire du filtrage avec netkat, nous devons installer des dépendances

Étant basé sur le netcat d'origine, il s'avère limité dans son extension vers d'autres n'étant pas liées à la sécurité réseau.

De moins en moins de résultats concluants liés à l'exploration de ce domaine existent en raison de sa complexité d'usage.

CONCLUSION

En conclusion, on peut retenir que netkat est un système formel qui peut être utilisé pour programmer, modéliser et raisonner sur les réseaux. Il est livré avec une boîte à outils formelle riche comprenant des sons et une axiomatisation algébrique complète, une théorie des automates, une sémantique dénotationnelle et une représentation symbolique efficace. Ces fondements théoriques permettent des outils puissants tels qu'une procédure de décision qui permet une vérification entièrement automatique des propriétés importantes du réseau et un compilateur rapide qui

implémente de puissantes abstractions de programmation. Cependant netkat possède des limites.

RÉFÉRENCES

https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands

https://www.it-connect.fr/a-la-decouverte-de-lutilitaire-netcat-sur-linux/

https://github.com/aojea/netkat/blob/main/development.md

https://github.com/aojea/netkat

http://www.haifux.org/lectures/217/netLec5.pdf

https://linux-kernel-labs.github.io/refs/heads/master/labs/networking.html

https://www.speedcheck.org/fr/wiki/socket/