République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes
De Master ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Intitulé de la spécialité**
Système Informatique

Présenté par :
Melle NEDJAM Thileli.
Melle BERKANI Nassima.

Thème

Filtrage du Trafic Réseau de L'université Mouloud Mammeri

Mémoire soutenu publiquement le 27/09/ 2018 devant le jury composé de :

Président : Mr. SADOUN.

Encadreur : Mme. BOURKACHE.

Examinatrice : Mme. BELATTEF.

Co-Encadreur : Centre des systèmes et réseaux d'information, de communication, de Télé-enseignement et Enseignement à Distance

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

Je tiens aussi à remercier très chaleureusement Mr Nadour pour son accueil, son aide, son attention et sa gentillesse tout au long du stage, qui ont fait de ces trois mois un moment très plaisant et intéressant.

Je remercie vivement l'ensemble des personnes du centre des systems et réseaux de l'Université Mouloud Mammeri pour l'excellent accueil, les conseils avisés et la bonne humeur partagée.

Je tiens également à remercier notre promotrice Mme Bourkache pour son aide, sa gentillesse et sa bonne humeur.

Nous adressons nos remerciements les plus sincères aux membres du jury qui nous font l'honneur de juger notre travail.

Ces remerciements ne seraient pas complets si nous n'avons pas pensé à les destiner, avec nos profondes reconnaissances, à nos parents qui nous ont offerts un environnement favorable pour mener à terme notre travail.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire.

Dédicace

Je dédie ce travail à mes chers parents autant d'expressions aussi vive soit elles ne sauraient exprimer ma gratitude et ma reconnaissance pour les sacrifices que vous avez consenti pour ma réussite, vous avez su m'inculquer le sens de la responsabilité et de la confiance en soi, je vous dois ce que je suis aujourd'hui. Je ferai toujours de mon mieux pour rester votre fierté, que dieu le tout puissant vous préservent bonne santé et longue vie.

A mon frère Younes et mes sœurs : Thanina, Yamina.

Vous avez animé mes jours vous me comblez de tendresse et d'affection je vous souhaite un avenir radieux plein de bonheur et de réussite.

A ma grand-mère Yamin et mes tantes .

A mes cousins et cousines : Lyes, Jugurtha, Momouh, Dany, Anis, Thinhinane, lydia , une pensée spéciale pour Amine .

A ma binôme: Nassima.

A tous mes amis : Zahia, Chabane, Ghilas, Nassima, Kahina, Sarah, Thinhinane, lynda, Katia, Nora, Kahina, Amine, Sofiane, Idir.

A mes meilleurs : Farid et Koceila.

A toute la section informatique M2 promotion 2017/2018

Nedjam Thileli

Dédicace

Je dédie ce travail à ceux qui ont éclairé ma vie ma très chère mère et mon très cher père pour l'éducation qu'ils m'ont prodigué, avec tous les moyens et au prix de tous les sacrifices qu'ils ont consenti à mon égard, pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance. Que le bon Dieu les garde et les protège.

A mes chères adorables sœurs Ryma, Wahiba, Karima, Farida, Nacira.

A mes frères Anis,Fouad,Soufian,Lyse .

A mes oncles et mes tantes.

A mes cousins et cousines.

A tous mes chers ami(e)Lynda, Tina, Dalila, Lydia, Nadjat(etc.).

A ma camarade Thileli.

A tous ceux qui me connaissent de près ou de loin.

A toute la promotion Master 2 Informatique 2018/2019.

Berkani Nassima

Sommaire

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Table des figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Liste des tableaux

Liste des abrivations

ACL: Access Control Lists

ADSL: Asymmetric Digital Subscriber Line

ARPA: Advanced Research Projects Agency

ASCII: American Standard Code for Information Interchange

CRC : cyclic redundancy check est un logiciel

Cisco ASA: Cisco Adaptive Security Appliance

DNS: Domain Name System

DOD: Department of Defense

DoS : Denial-of-Service

DMZ : Zone démilitarisée

FTP: File Transfer Protocol

GNU: GNU's Not UNIX

HTML: fichier HyperText MarkupLanguage

HTTP: Hypertext Transfer Protocol

IDS: Intrusion Detection System

IIS: Internet Information Services

IPS : Intrusion prévention system

LAN : Local Area Network ou Réseau Local d'Entreprise

LÈSE : Laboratoire de Mécanique, Structure et Énergétique

LOGEA : Laboratoire de Néo matériaux Environnement & Aménagement

MAN: Métropolitan Area Network

MAC : Media Access Control

MAU: Medium attachment unit

MIME: Multipurpose Internet Mail Extension

NAT: Network Address Translation

OSI: Open System Interconnexion

Owasp: Open web application security project

RPC: remote procedure call

S.I: Système d'Information

SSL: Secure Sockets Layer

SNMP: Simple Network Management Protocol

SQL: Structured Query Language

SWOT: Strngths/Weaknesses and Opportunities/Threats ou Forces/Faiblesses et Opportunités/Menaces

TCP: Transmission Control Protocol

TLS: Transport Layer Security

UDP: User Datagram Protocol

USB: Universal Serial Bus

VPN : Virtual Private Network ou Réseau privé virtuel

WAN: Wide Area Network ou Réseau Etendu

WAF: **Web** Application Firewall

WAFEC: the Web Application Firewall Evaluation Criteria

WASC: Web Application Security Consortium

XML : Extensible Markup Language

XSS: Cross-Site Scripting

Introduction Générale

De nos jours, la plus part des entreprises possèdent de nombreux postes informatiques qui sont en général reliés entre eux par un réseau local. Ce réseau permet d'échanger des données entre les divers collaborateurs internes à l'entreprise et ainsi de travailler en équipe sur des projets communs. La possibilité de travail collaboratif apportée par un réseau local constitue un premier pas. L'étape suivante concerne le besoin d'ouverture du réseau local vers le monde extérieur, c'est à dire internet.

La mise en ligne des services via les applications web est devenue une action indispensable voir naturelle pour n'importe quelle entreprise, banque, magasin, ou fournisseur désirant la disponibilité permanente vis-à-vis de ses clients. Dans le but de satisfaire la disponibilité permanente, leurs serveurs sont régulièrement victimes d'attaques visant soit à les rendre défaillants soit à accéder aux données sensibles qu'ils contiennent. Il est devenu primordial de les protéger contre des actes malveillants et aux attaques applicatives.

Les attaques de types applicatifs à l'occurrence de XSS et SQLI sont reconnues comme étant des vulnérabilités qui ne dépondent pas du réseau mais plutôt de l'application. Ce qui implique que des mécanismes de sécurité comme les pare-feu ou les IDS/IPS sont efficaces pour éliminer un volume important de menaces ciblant les couches inférieures ils s'avèrent nettement moins
adaptés à la protection contre les menaces de plus en plus ciblées et spécialisées qui affectent désormais les applications des entreprises.

En effet, les pare-feu comme Iptable, Nftable offrent une protection par rapport à l'en-tête, ils ont en charge d'autoriser la transmission des paquets en fonction de leur source et leur destination, mais n'arrivent pas à analyser leur contenu. Ce qui fait, ils sont incapables d'obtenir des informations sur les attaques et les attaquants.

Aussi, les IDS/IPS n'ont pas l'habilité de comprendre la logique du protocole HTTP et donc n'arrivent pas à détecter si une requête est normale ou malicieuse dans la couche application. Et par la suite n'arrivent pas à détecter ni à prévenir les nouvelles attaques sans signatures. Devant une telle situation, l'utilisation d'un pare-feu applicatif (WAF : Web Application

Firewall en anglais) s'avère très nécessaire pour assurer la disponibilité des organismes quelles que soit leurs activités vis-à-vis leurs clients.

L'utilisation des WAFs permet le «Virtual Patching », donc, chaque requête est contrôlée avant d'être envoyée au serveur. Si la requête est considérée comme valide, elle sera relayée au serveur. En revanche, si le WAF découvre un contenu dangereux dans cette requête, il y répondra sans solliciter le serveur concerné.

Le centre de calcul et réseau est l'un des organismes qui a compris cette problématique ,et qui travaille sans relâche pour mettre en œuvre cette nouvelle technologie de traitement, pour assurer une gestion plus sécurisé, plus fiable, plus rigoureuse, moins fastidieuse, et pour minimiser le risque d'erreurs et assurer la disponibilité de l'information à toute éventuelle demande.

Notre travail, s'inscrit dans cette démarche et consiste en la mise en place d'un pare feu applicatif pour le filtrage de trafic de l'UMMTO.

Pour ce faire et pour mener à bien notre travail nous avons opté pour une démarche méthodologique articulée autour des étapes suivantes :

- Chapitre 1 : Généralités sur les réseaux et la sécurité.
- Chapitre 2 : les firewalls (les types, l'architecture, et leurs déploiements).
- Chapitre 3 : Analyse et conception (concerne la présentation de l'organisme d'accueil et l'étude du système existant).
- Chapitre 4 : Réalisation, c'est l'installation et la configuration des outils.

I. Introduction

Chaque ordinateur connecté à internet et d'une manière plus générale à n'importe quel réseau informatique, est susceptible d'être victime d'une attaque.

Ainsi il est nécessaire de se protéger de ces attaques réseaux en installons des dispositifs de sécurité. Nous consacrons la première partie de ce chapitre à expliquer quelques notions générales sur les interconnexions réseaux, le modèle OSI ainsi que le protocole TCP/IP, ensuite dans la deuxième partie nous allons montrer les moyens et les dispositifs de sécurité utilisés pour l'assurer.

I.1. Réseaux Informatiques

I.1.1. Présentation

Un réseau informatique est une collection d'objets de télécommunication d'informatiques, qui sont connectés entre eux par des supports de transmission (filaires ou non filaires), pour le but d'établir des communications entre ses différentes machines afin d'assurer des échanges d'informations tel que le transfert de fichiers, le partage de ressources (imprimantes et de données), la messagerie ou l'exécution des programmes à distance.

Le défi des réseaux informatiques est d'interconnecter les différents matériels issus des développements technologiques et d'être le plus transparent possible vis-à-vis de l'utilisateur.

Le protocole TCP/IP s'impose de plus en plus comme langage de communication qui permet de fédérer un environnement hétérogène. [12]

I.1.2. Les Différents Types De Réseaux Informatiques

On distingue différents types de réseaux informatiques, selon leur taille (nombre des machines), leur vitesse de transfert des données ainsi leur localisation, se résume en trois grandes catégories :

a) Réseau LAN

C'est un ensemble d'ordinateurs et d'équipements informatique reliés les uns aux autres dans un même bâtiment, site ou dans des sites différents ayant une aire géographiquement proche, ne dépassant pas 10 Km. [4]

b) Réseau MAN

C'est l'Interconnexion de plusieurs LAN géographiquement proches. Ainsi il permet à deux ordinateurs distants de communiquer comme s'ils faisaient partie d'un même réseau local. [4]

c) Réseau WAN

Interconnecte plusieurs LAN à travers de grandes distances géographiques. Le WAN le plus utilisé est internet qui permet de connecter des ordinateurs à l'échelle planétaire. [4]

I.1.3. Les Topologies d'interconnexion des réseaux

La configuration spatiale des équipements du réseau est appelée topologie physique. On distingue généralement les topologies suivantes :

I.1.3.1. Le bus

La topologie en bus repose sur un câblage, sur lequel viennent se connecter des nœuds (postes de travail, équipements d'interconnexion, périphériques). Il s'agit d'un support multipoints. Le câble est l'unique élément matériel constituant le réseau et seuls les nœuds génèrent les signaux. L'inconvénient majeur repose sur le fait qu'une seule coupure du câble empêche toute station d'échanger des informations sur le réseau. [4]

I.1.3.2. L'anneau

Chaque équipement est relié à l'équipement voisin de telle sorte que l'ensemble forme une boucle fermée. Les Nœuds sont actifs, ils reçoivent et régénèrent le message. Mais en cas de coupure de l'anneau le réseau est interrompu, ce qui est le cas lors de l'installation d'une nouvelle station de travail.

On peut résoudre cette sensibilité aux coupures, en doublant l'anneau unidirectionnel ou bidirectionnel. [4]

I.1.3.3. L'étoile

Dans une topologie en étoile tous les MAU (Medium attachement unit)du réseau sont connectés à un nœud central c'est le concentrateur. L'ensemble des messages transite par lui.

Le câblage du réseau est plus coûteux que celui de la topologie en bus. Il est effectué à l'aide de câble en paires torsadées. [4]

I .1.4. Architecture des réseaux

Un réseau, permet de connecter des ordinateurs entre eux. Mais les besoins sont très divers, depuis le réseau domestique ou d'une toute petite entreprise jusqu'aux réseaux des grandes

sociétés. Voyons deux approches fondamentalement différentes, encore que l'une puisse facilement évoluer vers l'autre.

I.1.4.1. Poste à Poste (Peer to Peer)

Les postes de travail sont simplement reliés entre eux par le réseau. Aucune machine ne joue un rôle particulier. Chaque poste peut partager ses ressources avec les autres postes. C'est à l'utilisateur de chaque poste de définir l'accès à ses ressources. Il n'y a pas obligatoirement d'administrateur attitré et chaque poste peut partager toute ou une partie de sa mémoire de masse, comme il peut partager son imprimante. [5]

I.1.4.2. Clien_serveur

L'architecture client_serveur désigne le mécanisme de communication entre la machine cliente et le serveur, elle est basée sur la fourniture des services.

Il existe plusieurs variantes des architectures client_serveur, selon le nombre des niveaux (tiers) mis en œuvre. On distingue alors :

a. L'architecture 2-tiers

L'architecture à deux niveaux (2-tiers) est l'architecture la plus classique, elle décrit les systèmes Client_serveur dans lesquels, la logique applicative est enfouie soit dans l'interface utilisateur chez le client, soit dans la base de données chez le serveur (ou dans les deux à la fois).

Dans cette architecture, le serveur exécute la requête du client et fournit directement le service, sans faire appel à d'autres intermédiaires. [5]

b. L'architecture 3-tiers

Dans cette architecture, la logique applicative réside dans un niveau intermédiaire, séparément des données et de l'interface utilisateur. Un niveau supplémentaire est ajouté :

- ✓ Un client (l'ordinateur demandeur de ressources) équipé d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.
- ✓ Un serveur d'application (appelé middleware) qui fournit la ressource, mais en faisant appel à un autre serveur.
- ✓ Un serveur de données qui fournit au serveur d'application les données requises pour répondre au client.

c. L'architecture N-tiers (multi-tiers)

L'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et recevoir des applications puissantes et simples à maintenir.

Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre les niveaux.

Cette distribution est facilitée par l'utilisation des composants « métiers » spécialisés et indépendants introduits par les concepts orientés objets (langages de programmation et middleware), ces composants rendent un service. Et ils sont capables de communiquer entre eux et peuvent coopérer en étant implantés sur des machines distinctes. [5]

I.1.5. Les Modèles de transmission de données

Les protocoles de transmission de données sont ceux qui permettent à deux entités de communiquer à travers un réseau de télécommunication.

Un protocole est un ensemble de règles à respecter pour que ces deux entités puissent s'échanger de l'information. Il faut que les deux extrémités utilisent les mêmes règles pour que la communication puisse se faire.

Nous allons nous intéresser principalement aux protocoles de l'architecture de référence de l'ISO et nous les comparerons à ceux de l'architecture TCP/IP.

I.1.5.1. Modèle OSI

Le modèle OSI définit de quelle manière les protocoles sont associés entre eux pour permettre une communication entre les ordinateurs et les périphériques en réseau. Il spécifie le comportement d'un système ouvert (qui échange en permanence des informations). Ce modèle est normalisé par l'ISO.

Se décompose en 7 couches, Chaque couche est responsable d'un aspect de la communication. Une couche de niveau N communique avec les couches de niveau N-1 (encapsulation) et N+1(désencapsulation). [4]

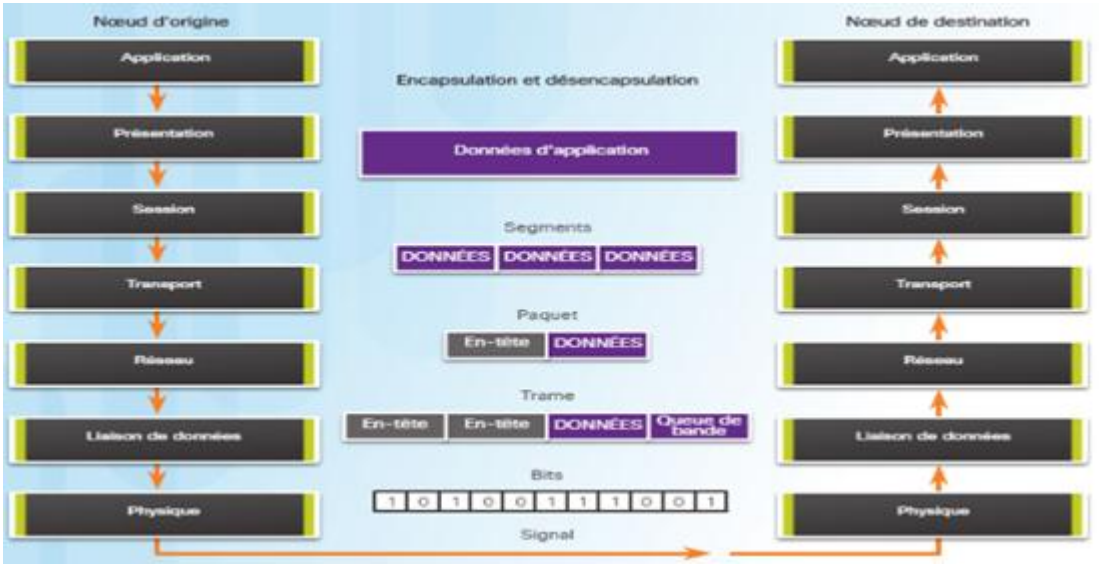


Fig. I.1. Modèle OSI. [13]

I.1.5.1.1. Description des couches

Chaque couche réseau définie par le modèle à un rôle bien précis, qui va du transport du signal codant les données à la présentation des informations pour l'application destinataire.

a) La couche physique

Elle a pour rôle la transmission bit à bit sur le support, entre l'émetteur et le récepteur, des signaux électroniques, électromagnétiques ou lumineux qui codent des données numériques (0 ou 1). [4]

b) La couche Liaison de données

C'est au niveau de cette couche que les données numériques sont traduites en signal. Les bits de données sont organisés en trames. Un entête est créé dans lequel l'émetteur et le destinataire sont identifiés par leur adresse physique. [4]

c) La couche Réseau

C'est la couche 3 du modèle OSI, qui correspond à la couche internet du modèle TCP-IP, est responsable du routage, c'est la fonction principale de cette couche. Une fois que la couche transport a assuré son rôle, les données sont envoyées à la couche réseau. Cette dernière se chargera d'ajouter toutes informations en rapport avec le routage, soit l'adresse IP du destinataire. C'est la seule couche du modèle OSI qui utilise la connexion logique. En fait, bien que cette couche ait pour rôle de déterminer

Le chemin physique à emprunter en se basant sur l'adresse IP du destinataire, les conditions du réseau et plusieurs autres facteurs, elle ne peut pas établir une connexion physique. D'où son rôle se limite à la connexion logique. Une fois qu'elle a ajouté à l'entête du paquet des informations qui lui sont spécifiques. [4]

d) La couche Transport

Il s'agit du cœur du modèle OSI. Au niveau de cette couche, différents mécanismes sont mis en œuvre pour établir un mode connecté, cette couche assure l'arrivée des paquets dans l'ordre et sans erreur, en échangeant les accusés de réception de données et en retransmettant les paquets perdus. Cette communication est dite de type de bout en bout. Les protocoles de la couche transport à ce niveau sont TCP et UDP.

TCP assurent des services de bout en bout fiables. UDP assure des services de datagramme peu fiables. [4]

e) La couche Session

La notion de session est assez proche de celle de connexion. Il existe cependant quelques détails qui peuvent justifier la présence de ces deux concepts. Une seule session peut ouvrir et fermer plusieurs connexions, de même que plusieurs sessions peuvent se succéder sur la même connexion.

Le protocole le plus connu à ce niveau est l'appel de procédure distante RPC. [4]

f) La couche Présentation

Elle assure la mise en forme des données : paramètres internationaux, pages de codes, formats divers**,** cette couche peut également exploiter des fonctions de chiffrement et de compression. Des codages comme MIME, ASCII, peuvent être utilisés ici. [4]

g) La couche Application

Représente des données pour l'utilisateur ainsi que du codage et un contrôle du dialogue : des mécanismes de communication offerts aux applications de l'utilisateur.

Elle contient une variété de protocoles qui sont utiles aux utilisateurs (HTTP, FTP, DNS, etc.). [4]

I.1.5.2. Modèle TCP/IP

Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules ou couches) mais il contient uniquement quatre couches.

Le nom du modèle TCP/IP est étroitement lié à deux protocoles **:** le protocole **TCP** et le protocole **IP**, Ceci est dû au fait que sont les deux protocoles les plus utilisés pour Internet. Voici une figure qui illustre le modèle TCP/IP : [4]

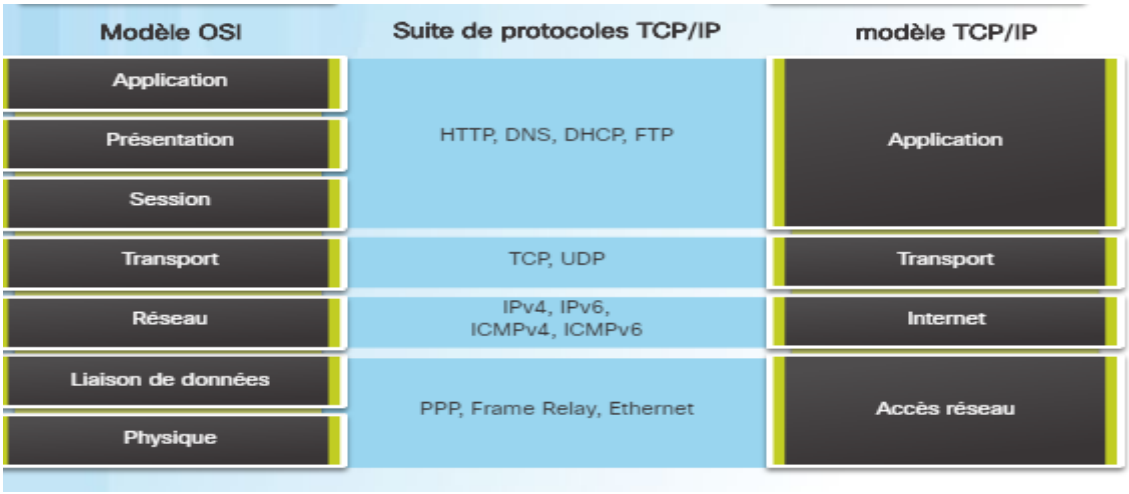


Fig. I.2. Modèle TCP/IP.

NB : Le modèle OSI a été mis à cotés pour faciliter la comparaison entre les deux modèles.

La différence flagrante : 4 couches pour le modèle TCP/IP et fusion des couches Présentation et Session en une grosse couche Application.

I.1.5.2.1. Processus de communication pour les 4 couches du modèle TCP/IP

Le processus de communication complet qui comprend ces différentes étapes :

1. La première étape est la création des données au niveau de la couche application du périphérique final à l'origine de la demande,
2. Maintenant, les données vont être segmentées et encapsulées lorsqu'elles vont descendre la pile de protocoles dans le périphérique final source,
3. La troisième étape consiste à générer des données sur les différents supports qui se trouvent au niveau de la couche d'accès au réseau dans la pile,
4. Cette quatrième étape permet le transport des données à travers l'inter-réseau, qui est composé de supports et de n'importe quels périphériques intermédiaires,
5. Désormais, la couche d'accès au réseau du périphérique final de destination va réceptionner les données,
6. L'étape suivante est la désencapsulation et l'assemblage de nos données lorsqu'elles remontent la pile au niveau du périphérique de destination,

7. Nous arrivons à la dernière étape, c'est la transmission de ces données à l'application de destination. Celles-ci se transmettent au niveau de la couche application du périphérique final de destination. [9]

I.1.5.3. Comparaison entre les deux modèles OSI et TCP/IP

Les deux modèles OSI et TCP/IP comportent :

Critères	Modèle OSI et TCP/IP
Ressemblances	✓ Des couches, ✓ Une couche application, bien que chacune fournisse des services différents, ✓ Des couches réseaux et transports comparables, ✓ Tous deux s'appuient sur un réseau à commutation de paquets, et non sur un réseau à commutation de circuits.
Différences	✓ Le modèle TCP/IP est relativement flou vis à vis la couche application en effet c'est aux constructeurs de spécifier cette couche, ✓ Le modèle OSI est trop détaillé. ✓ La dernière grande différence est liée au mode de connexion. Certes, les modes orientés connexion et sans connexion sont disponibles dans les deux modèles mais pas à la même couche.

Tab. I.1.Comparaison entre OSI et TCP/IP.

I.1.6. La pile de protocoles intervenants au niveau de chaque couche du modèle OSI

Un protocole réseau est un ensemble de règles et de procédures de communication utilisées de part et d'autre par toutes les stations qui échangent des données sur le réseau.

Il existe de nombreux protocoles réseaux, mais ils n'ont pas tous, ni le même rôle, ni la même façon de procéder. Certains protocoles réseaux fonctionnent au niveau de plusieurs couches du modèle OSI, d'autres peuvent être spécialisés dans la réalisation d'une tâche correspondant à une seule couche du modèle OSI. Un paquet transmis sur le réseau est constitué de plusieurs

Couches d'informations correspondant aux différents traitements de chacun des protocoles de la pile. Voici une figure qui illustre la pile de protocole : [6]

Fig. I.3. Les couches du modèle OSI et les protocoles correspondants. [13]

I .1.7. Equipements d'interconnexions

Des matériels sont donc utilisés pour interconnecter les réseaux entre eux. Ils permettent également de segmenter les réseaux de taille importante.

Pour les premiers niveaux du modèle OSI correspondent des équipements réseaux spécifiques qui sont présentés dans le tableau suivant :

Niveau OSI	Equipement	Unité échangée
7	- Serveur - Passerelle (Gateway)	
3	- Routeur - Brouteur=routeurs + pont - Passerelle	Datagramme
2	- Pont(Bridge) - Commutateur	Trame
1	- Répéteur - Concentrateur(Hub)	Bit

Tab.I.2. Equipements d'interconnexions.

I.2. La sécurité informatique

I.2.1. Présentation

Le système d'information représente l'ensemble des données de l'entreprise ainsi que ses infrastructures matérielles et logicielles. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

La sécurité informatique, d'une manière générale, consiste à s'assurer que les ressources matérielles et logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

I.2.2. Critères de la sécurité

La sécurité informatique vise généralement cinq principaux objectifs :

a. **L'intégrité** qui garantit que les données sont bien celles que l'on croit être, qu'elles n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle),
b. **La confidentialité** qui consiste à rendre l'information inintelligible à d'autres personnes, autres que les seuls acteurs de la transaction,
c. **La disponibilité** qui permet de garantir l'accès à un service ou à des ressources,
d. **La non-répudiation** de l'information qui est la garantie qu'aucun des correspondants ne pourra nier la transaction,

e. **L'authentification** qui consiste à assurer l'identité d'un utilisateur, c'est-à-dire à garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. [3]

I.2.3. Nécessité d'une approche globale

La sécurité d'un système informatique fait souvent l'objet de métaphore. En effet, on la compare régulièrement à une chaîne en expliquant que le niveau de sécurité d'un système est caractérisé par le niveau de sécurité du maillon le plus faible. Ainsi, une porte blindée est inutile dans un bâtiment si les fenêtres sont ouvertes sur la rue. Cela signifie que la sécurité doit être abordée dans un contexte global et notamment prendre en compte les aspects suivants :

1. **La sensibilisation** des utilisateurs aux problèmes de sécurité,
2. **La sécurité logique :** c'est-à-dire la sécurité au niveau des données, notamment les données de l'entreprise, les applications ou encore les systèmes d'exploitation,
3. **La sécurité des télécommunications** : technologies réseau, serveurs de l'entreprise, réseaux d'accès,
4. **La sécurité physique :** soit la sécurité au niveau des infrastructures matérielles. [1]

I.2.4. Mise en place d'une politique de sécurité

La sécurité des systèmes informatiques se cantonne généralement à garantir les droit d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des ressources possèdent uniquement les droits qui leur ont été octroyés.

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils

Puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, dont la mise en œuvre se fait selon les quatre étapes suivantes :

1. Identifier les besoins en termes de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences,

2. Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés,

3. Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés,

4. Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace.

La politique de sécurité est donc l'ensemble des orientations suivies par une organisation en matière de sécurité. [1]

I.2.5. Les attaques

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque. Une **attaque** est l'exploitation d'une faille (vulnérabilité) d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables. [3]

Dans ce qui suit, nous décrivons brièvement la classification des attaques ainsi qu'une description de quelques attaques basées sur ces faiblesses. [3]

I.2.5.1. Les scénarios d'attaques

Les scénarios d'attaques peuvent être classés en deux grandes catégories :

a) Attaque passive : Dans ce genre d'attaques, les informations ne sont pas modifiées. L'attaquant collecte seulement les informations qui circulent sur le réseau.

b) Attaque active : Il y a trois cas possible pour mener une attaque active : [15]

1. **L'interruption :** L'intrus intercepte le message envoyé par l'utilisateur A pour B et l'interrompt, ceci illustré par la figure présentée ci-dessous :

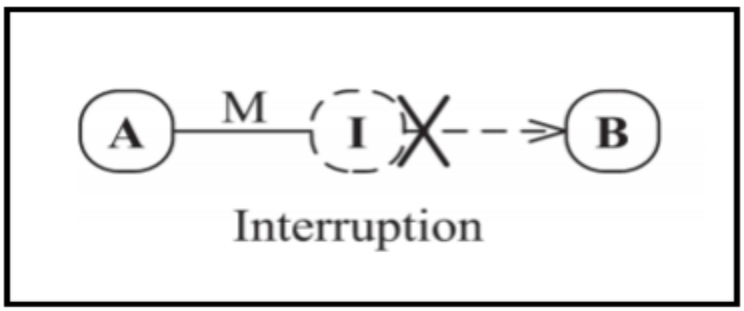


Fig. I.4. L'interruption.

2. **La modification :** L'intrus intercepte le message envoyé par l'utilisateur A et le modifie avant de le faire suivre à l'utilisateur B, comme il est montré dans la figure qui suit :

Fig. I.5. La modification.

3. **La fabrication :** L'intrus fabrique un message et l'envoie à l'utilisateur B en se passant pour l'utilisateur A, ceci est illustré dans la figure ci-dessous :

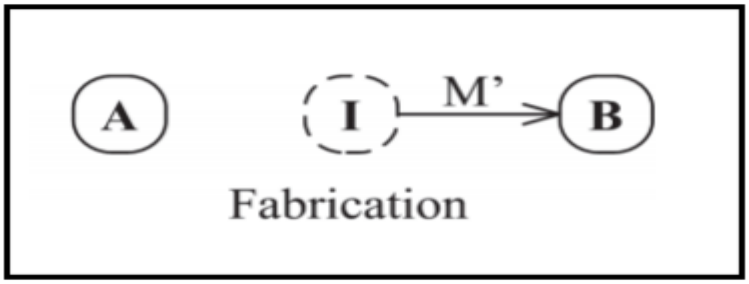


Fig. I.6. La fabrication.

I.2.5.2. Types d'attaques

Nombreuses ont été et sont encore les attaques informatiques. Nous en donnons un bref aperçu, triés par cible d'attaque :

a) Hardware : le hardware est un point d'attaque facile car il est visible. La liste des attaques humaines est sans fin, que ces dernières soient involontaires ou volontaires.

b) Software : le software peut être détruit, modifié, effacé, déplacé. Le résultat est identique dans chaque cas, on perd l'accès au programme voulu. La modification est sans doute la pire des attaques car elle peut causer de dangereux troubles ultérieurs. Les bombes logiques, les

chevaux des Troie, les virus sont différentes techniques de modification ayant chacune leurs propres spécificités.

c)Données : la confidentialité des données peut être mise en défaut par "mise sur écoute", par simple requête, en déroutant les appareils de sortie de données, La modification des données est en général plus compliquée à mettre en œuvre car elle nécessite une plus grande connaissance technologique.

d)Réseau : les réseaux ajoutent à l'ensemble de la sécurité le problème de la communication. L'utilisation de moyens de transports partagés et les accès longue distance sont deux points cruciaux dont il faut tenir compte.

e) Accès : l'utilisation abusive d'un accès peut découler sur des pertes de performances, des pertes commerciales, mais aussi des pertes de données.

I.2.5.3. Description de quelques attaques

Les attaques réseaux sont aujourd'hui nombreuses, elles touchent généralement les trois composants suivants d'un système : la couche réseau, le système d'exploitation et la couche application. Dans ce qui suit on va citer des exemples d'attaques de chaque composant :

I.2.5.3.1. Les attaques de la couche réseau

a) **L'attaque IP spoofing :** Est une technique consistant à remplacer l'adresse IP de l'expéditeur par l'adresse IP d'une autre machine. Le pirate commence par choisir le système qu'il veut attaquer, ensuite, après avoir obtenu le maximum de détails sur le système cible, il détermine les adresses IP autorisés à se connecter au système cible.

b) **Ecoute du réseau (sniffer) :** Il existe des logiciels qui permettent d'intercepter certaines informations qui transitent sur un réseau local, en transcrivant les trames dans un format plus lisible (Network packetsniffing). C'est l'une des raisons qui font que la topologie en étoile autour d'un hub n'est pas la plus sécurisée, puisque les trames qui sont émises en « broadcast » sur le réseau local peuvent être interceptées. De plus, l'utilisateur n'a aucun moyen de savoir qu'un pirate a mis son réseau en écoute. La meilleure solution est l'utilisation de mot de passe non redoutable, de carte à puce ou de calculette à mot de passe.

I.2.5.3.2. Les attaques de la couche application

a) **SQL Injection :** Comme son nom l'indique, l'attaque par 'SQL injection' consiste à envoyer du code SQL au serveur de manière à passer une requête valide sur le moteur de la base de données.

Les menaces de cette attaque sont différentes selon les objectifs de l'attaquant :

- Usurper une identité afin de se connecter sur une application web.
- Rendre l'application inutilisable.
- Supprimer toutes les données de la table visée, voir celle de la base de données complète.

Recommandation : comme cité précédemment, ce genre d'attaques repose principalement sur l'utilisation de caractères spécifiques qui permettent de mettre en commentaire des portions de code et d'insérer du code malicieux.

Certes, il est rare qu'une application ait besoin d'accepter les caractères suivants :

& ~ <>, ; : ! £ $ [ ] ( ) - | ' « _ .

Cependant, les applications web de gestion du contenu telles que les forums doivent les accepter.

Donc, pour protéger une application contre les attaques par injection SQL :

Il faut vérifier que les caractères utilisés sont adéquats, ce contrôle doit être effectué au niveau du client utilisant 'java Script' et au niveau du serveur lorsque les paramètres sont récupérés pour remplir la faille de sécurité.

Exemple : le langage PHP offre une fonction « htmlspecialchars » qui permet de transformer les caractères et une fonction mysql_real_escape_string permettant de filtrer les entrées.

- Il faut s'assurer que les valeurs sont bien du format requis (longueur, intervalle de valeurs, etc.).

a) **Attaque XSS (Cross Site Scripting)** : Les failles XSS se produisent lorsque l'application donne lieu à des pages contenant des données soumises au préalable par un client sans les avoir validées ou assainies. Ces pages, renvoyées aux clients, peuvent donc inclure du code exécutable frauduleux qui va s'exécuter dans le navigateur de ces clients. Donc, la cible de cette attaque est l'utilisateur d'un site web, au travers de l'exploitation d'une vulnérabilité de ce site (d'où le terme cross-site).

On distingue entre deux types d'attaques XSS :

- **Persistant (stored) :** a lieu lorsque l'attaquant réussisse à stocker dans la base de données du code malveillant qui sera exécuté par la victime lorsqu'elle tentera d'afficher les données de l'application Web et peut atteindre plusieurs victimes.
- **Reflected :** le principe de l'attaque reste le même que dans le cas persistant, à la différence que le code malveillant n'est pas stocké de façon permanente sur le serveur vulnérable. Il peut par exemple, être inclus dans un paramètre de requête que l'on soumet au site vulnérable. L'attaquant, dans ce cas, doit trouver un moyen de forcer sa victime à invoquer cette URL avec ce paramètre particulier (par exemple, en lui proposant de cliquer sur un lien dans un email).

  Le code malveillant téléchargé et exécuté dans le navigateur de la victime peut avoir différents objectifs. L'attaquant peut, par exemple, faire exécuter à sa victime un script dans son navigateur afin de rediriger automatiquement ce client vers une autre URL (qui peut être une copie conforme du site légitime) pour voler ses identifiants de session.

  Recommandations

- Le serveur web doit être régulièrement mis à jour.
- Les entrées utilisateurs doivent être soigneusement contrôlées et validées avant l'exécution.
- Les caractères !@ £ $ * + - '( ) ! ? ; :>< / | [ ] ainsi que leurs versions encodées doivent être bannis lors du remplissage des formulaires pour éviter l'exécution du code JavaScript.
- Les transformations suivantes doivent être effectuées :

  & → &amp ;

  < → &lt ;

  > → &gt ;

  ' → &#x27 ;

  / → &#x2F ;

I.2.5.3.3. Les attaques du système d'exploitation

Cheval de Troie : Dans ce type d'attaque, le pirate, après avoir accédé à votre système ou en utilisant votre crédulité, installe un logiciel qui va à votre insu, lui transmettre par Internet les informations de vos disques durs. Un tel logiciel, aussi appelé troyen, peut aussi être utilisé pour générer de nouvelles attaques sur d'autres serveurs en passant par

le vôtre. Certains d'entre eux sont des «key logger» c'est à dire qu'ils enregistrent les frappes faites au clavier.

I.2.6. Dispositifs de la sécurité informatique

Nous constatons que les attaquants disposent de plusieurs moyens pour réussir chaque phase d'attaque, la disponibilité des outils d'attaques et la richesse des sources d'informations accentuent le risque des intrusions. Par conséquent les administrateurs sécurisent de plus en plus leurs systèmes informatiques. Ils s'appuient sur diverses solutions comme les pare feux les systèmes de détection d'intrusions, les antivirus, etc. Qu'on va détailler dans ce qui suit :

I.2.6.1 Pour le système d'exploitation

a) Les antivirus

La majorité des solutions antivirus détectent les formes de malware les plus répandues. Cependant, les cybercriminels développent et déploient chaque jour de nouvelles menaces. Pour disposer d'une solution antivirus efficace, il faut donc veiller à ce que les signatures soient toujours à jour. Une signature est semblable à une empreinte digitale. Elle identifie les caractéristiques d'un code malveillant. [2]

I.2.6.2. Pour la couche réseau

a) Les réseaux privés virtuels

Un Réseau Privé Virtuel est un moyen de communication assurant la sécurité des transferts de données sur des réseaux publics ou partagés (exemple :l'ADSL). Un VPN est, en fait, un réseau de communication avec les mêmes paramètres de sécurité qu'un réseau privé. Ses principales caractéristiques sont :

- Confidentialité des données : le chiffrement assure que le contenu des données transmises n'est connu que des parties qui échangent l'information. De ce fait, un tiers interceptant le trafic du VPN n'aura pas la possibilité d'en déterminer la teneur.
- Intégrité des données : les méthodes cryptographiques employées assurent que les données reçues à travers le VPN par le destinataire sont identiques à celles envoyées par l'expéditeur.

- Authentification des utilisateurs du VPN : il est important de savoir quels sont ceux qui participent au processus afin d'éviter les problèmes de sécurité liés à l'usurpation d'identité et par la même à l'accès illicite aux réseaux privés.

Le VPN est une technologie permettant l'extension logique du réseau, ou d'un sous-réseau, de l'organisation par l'ajout de postes ou sous-réseaux se trouvant à l'extérieur des limites physiques de celle-ci. Concrètement les employés travaillant de chez eux seront virtuellement dans le réseau interne de l'organisme ou deux sites distants, voire aux antipodes l'un de l'autre, pourront partager le même réseau. [8]

b) Mécanisme de translation d'adresse

Le mécanisme de translation d'adresses « NAT » a été mis au point afin de répondre à la pénurie d'adresses IP, Le principe du NAT consiste donc à utiliser une adresse IP routable (ou un nombre limité d'adresses IP) pour connecter l'ensemble des machines du réseau en réalisant, au niveau de la passerelle de connexion à internet, une translation (littéralement une « traduction ») entre l'adresse interne (non routable) de la machine souhaitant se connecter et l'adresse IP de la passerelle. Cette passerelle peut être un routeur tel que montré dans la figure suivante :

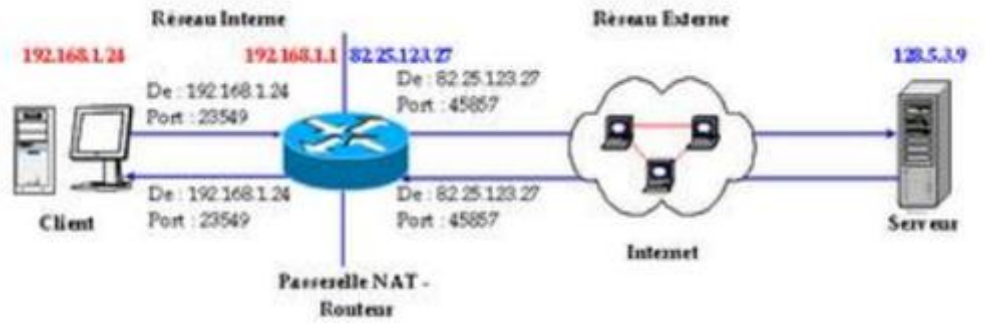


Fig. I.7. Mécanisme de translation d'adresse.

D'autre part, le mécanisme de translation d'adresses permet de sécuriser le réseau interne étant donné qu'il camoufle complètement l'adressage interne. En effet, pour un observateur externe au réseau, toutes les requêtes semblent provenir de la même adresse IP.

c)Pare feux

Un pare feu (firewall) est un système physique ou logique qui inspecte les flux entrant et sortant du réseau. Il se base sur un ensemble de règles appelées ACL afin d'autoriser ou interdire le passage des paquets. A ce niveau-là il existe principalement deux types de pare feux :

- ✓ Pare-feu avec filtrage des paquets : ce pare feu filtre les paquets en utilisant des règles statiques qui testent les champs des protocoles jusqu'au niveau transport,
- ✓ Pare-feu à filtrage des paquets avec mémoire d'états : ce modèle conserve les informations des services utilisés et des connexions ouvertes dans une table d'état. Il détecte alors les situations anormales suite à des violations des standards protocolaires. [11]

I.2.6.3. Pour la couche application

a) Zone démilitarisée

Lorsque certaines machines du réseau interne ont besoin d'être accessibles de l'extérieur (serveur web ou un serveur de messagerie), il est souvent nécessaire de créer une nouvelle interface vers un réseau à part, accessible aussi bien du réseau interne que de l'extérieur, sans pour autant risquer de compromettre la sécurité de l'entreprise. On parle ainsi de « zonedémilitarisée ».

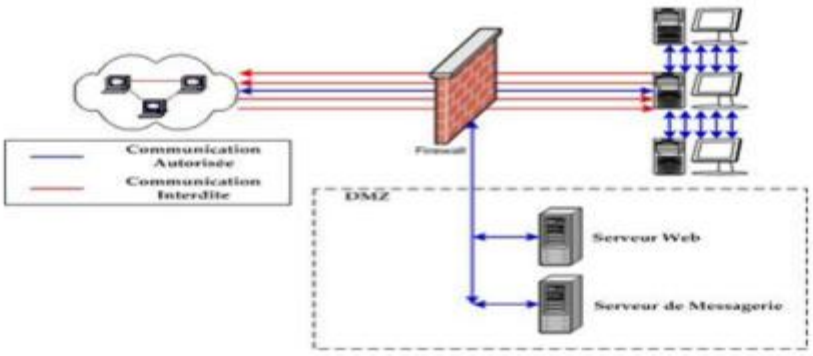


Fig.I.8. Architecture DMZ.

Une zone démilitarisées (ou DMZ en anglais demilitarized zone) est un sous-réseau séparé du réseau local et isolé de celui-ci et d'Internet par un pare-feu. Ce sous-réseau contient les machines étant susceptibles d'être accédées depuis Internet. Le pare-feu bloquera donc les accès au réseau local pour garantir sa sécurité et les services susceptibles d'être accédés depuis Internet seront situés en DMZ. La politique de sécurité mise en œuvre sur la DMZ est généralement la suivante :

– Trafic du réseau externe vers la DMZ autorisé,
– Trafic du réseau externe vers le réseau interne interdit,
– Trafic du réseau interne vers la DMZ autorisé ;
– Trafic du réseau interne vers le réseau externe autorisé,
– Trafic de la DMZ vers le réseau interne interdit,
– Trafic de la DMZ vers le réseau externe refusé,

La DMZ possède donc un niveau de sécurité intermédiaire, mais son niveau de sécurisation n'est pas suffisant pour stocker des données critiques pour l'entreprise. Il est à noter qu'il est possible de mettre en place des zones démilitarisées en interne afin de cloisonner le réseau interne selon différents niveaux de protection et ainsi les intrusions venant de l'intérieur.

b) Proxy et Reverse Proxy

1. **Proxy :** un proxy, ou encore appelé serveur mandataire, relaie une requête venant d'un client à destination d'un serveur auquel l'accès est indirecte en général. Lorsqu'on tape par exemple une adresse telle que *http://www.yahoo.com/index.html*, l'ordinateur se connecte à la hôte *http://www.yahoo.com/*et demande la page *index.html*, comme le montre la figure ci-dessous :

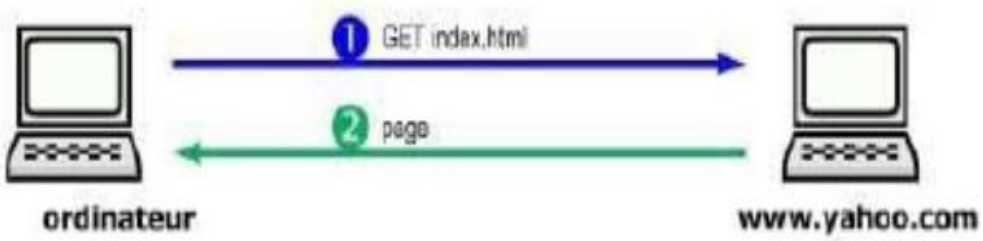


Fig.I.9. Requête HTTP sans proxy.

Cependant, avec proxy, quand on tape*http://www.yahoo.com/index.html*, l'ordinateur va se connecter au proxy et lui demander de chercher dans la page *http://www.yahoo.com/*. De cette façon, il protège l'ordinateur du client des attaques auxquelles il est exposé. En effet, il peut autoriser la connexion du client à l'extérieur et interdire les ordinateurs d'Internet de venir se connecter sur la sienne.

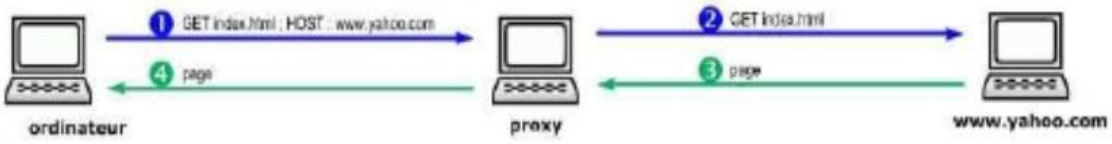


Fig.I.10. Requête HTTP avec proxy.

2. **Reverse Proxy** : un reverse proxy est un serveur proxy permettant aux utilisateurs d'internet d'accéder indirectement à certains serveurs internes. Cela revient à avoir des clients HTTP sur le Web (souvent inconnus) dont les requêtes passent par le proxy pour accéder aux serveurs du site qui héberge aussi celui-ci.

c) WAF

Un **Web Application Firewall** (WAF) est un type de pare-feu qui protège le serveur d'applications Web dans le backend contre diverses attaques. Le WAF garantit que la sécurité du serveur Web n'est pas compromise en examinant les paquets de requête HTTP / HTTPS et les modèles de trafic Web.

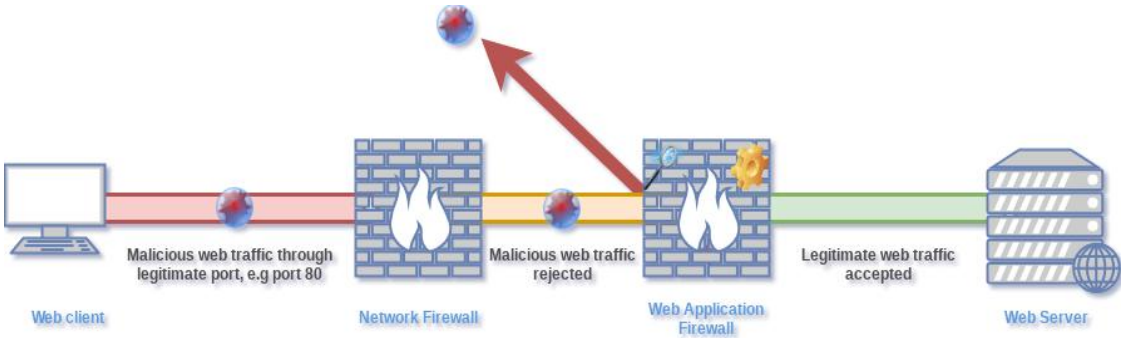


Fig. I.11. Web Application Firewall

I.3. Conclusion

La sécurité des réseaux informatiques est un sujet d'actualité. Ces systèmes sont trop ouverts, avec le grand nombre de réseaux que constitue Internet, ce qui fait que la sécurité de ces réseaux n'est pas totalement garantie. Les Pare-feu, les Proxys et les réseaux privés virtuels sont des outils développés et utilisés pour renforcer davantage cette idée de sécurité. Pour définir des mécanismes de sécurisation, il est nécessaire de définir avant tout, les objectifs de sécurité, pour obtenir tant que possible une sécurisation assez fiable de réseaux.

II. Introduction

L'objectif du réseau est d'offrir un maximum de connectivité et d'accès aux ressources. L'objectif de la sécurité est de limiter ces accès. Ces deux objectifs concurrents et contradictoires se trouvent être ceux d'un pare-feu. Depuis leur création, les pare-feu ont grandement évolué. Ils sont effectivement la première solution technologique utilisée pour la sécurisation des réseaux. Nous souhaitons structurer ce chapitre comme suit dans la premier partie nous allons présenter les pare-feu, leur fonctionnement ainsi les différents types de filtrages.

Dans la deuxième partie nous allons basé sur les pare-feu applicatif

II.1. Présentation des pare-feu

Un Pare-feu [appelé aussi Coupe-feu, Garde-barrière ou Firewall], est un système permettant de protéger un ordinateur, ou un réseau d'ordinateurs, des intrusions provenant d'un réseau tiers [notamment Internet]. Le Pare-feu est un système permettant de filtrer les paquets de données échangés avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum les interfaces réseaux (cartes réseaux) suivantes :

- Une interface pour le réseau à protéger (réseau interne),

- Une interface pour le réseau externe. [15]

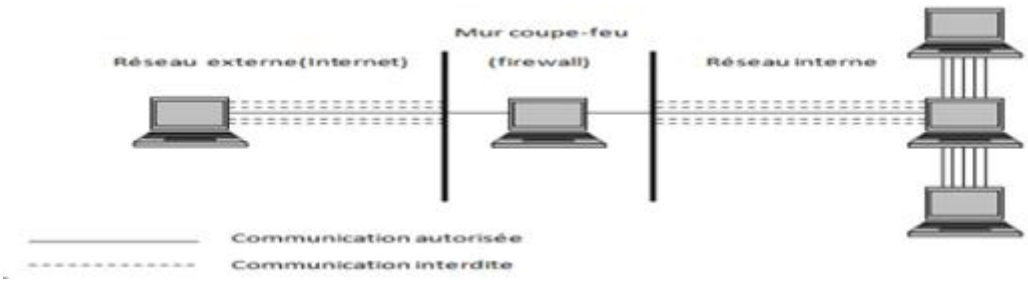


Fig. II.1. Pare-feu.

La configuration du Firewall est telle que les données arrivant sur l'une des cartes ne soient pas transmises directement sur l'autre mais de manière sélective, selon des critères de filtrage déterminés lors de sa configuration.

Le filtrage réalisé par le Pare-feu constitue le premier rempart de la protection du système d'information. Le système Pare-feu est un système logiciel, reposant parfois sur un matériel

Réseau dédié, constituant un intermédiaire entre le réseau local et un ou plusieurs réseaux externes. Il est possible de mettre un système Pare-feu sur n'importe quelle machine et avec n'importe quel système pourvu que :

- La machine soit suffisamment puissante pour traiter le trafic,
- Le système soit sécurisé,
- Aucun autre service que le service de filtrage de paquets ne fonctionne sur le serveur.

Un firewall est toujours caractérisé par la politique de sécurité qu'il utilise, dont le but est d'implémenter les prévisions d'utilisation des ressources informatiques ainsi que les procédures qui permettent les préventions des incidents en matière de sécurité.

Deux niveaux de sécurité sont mis en œuvre : la politique de conception du firewall et celle de l'accès aux services de la machine.

La politique de conception du firewall étant orientée vers les couches de niveau bas, il s'agira de mettre en œuvre les restrictions d'accès et de filtrage des services telles qu'elles sont définies par la politique de sécurité. Ceci ne peut être fait que si on a connaissance des limitations technologiques et des capacités de la pile des protocoles TCP/IP. Par contre, la politique d'accès aux services est orientée vers le niveau haut ou supérieur. Il s'agira donc de définir les services qui seront permis à accès, et ceux dont l'accès seront interdits. [16]

II.2. Les différents types de pare-feu

Il existe trois types de pare-feu qui sont les suivants :

II.2.1. Les pare-feu bridges

Les pare-feu bridges agissent comme des câbles réseau avec la fonction de filtrage en plus, leurs interfaces ne possèdent pas d'adresse IP et ne font que transférer les paquets d'une interface à une autre. Cette absence d'adresse IP est particulièrement utile, car cela signifie que le pare-feu est indétectable pour un haker lambda.

En effet, quand une requête est émise sur la câble réseau, le pare-feu bridge ne répondra jamais, car ses adresses MAC ne circuleront jamais sur le réseau, et comme il ne fait que transmettre les paquets, il sera totalement invisible sur le réseau. Cela rend impossible toute attaque dirigé contre le pare-feu. Parmi ses avantages on trouve qu'il est impossible de l'éviter puisque les paquets passeront par ses interfaces et il est peu coûteux, par contre sa configuration est souvent contraignante. [14]

II.2.2. Les pare-feu matériels

Les pare-feu matériels se trouvent souvent sur des routeurs achetés dans le commerce par de grands constructeurs comme Cisco. Intégrés directement dans la machine. Leur configuration est souvent relativement ardue, mais leur avantage est que leur interaction avec les autres fonctionnalités du routeur est simplifiée de par leur présence sur le même équipement réseau, ils sont intégrés au matériel réseau avec une administration relativement simple et ils ont un bon niveau de sécurité mais ils dépendent du constructeur pour les mises à jour. [14]

II.2.3. Les pare-feu logiciels

Les pare-feu logiciels sont présents à la fois dans les serveurs et les routeurs, nous pouvons les classer en deux catégories ; les pare-feu personnels et les pare-feu plus sûr. Les pare-feu personnels sont assez souvent commerciaux et ont pour but de sécuriser un ordinateur particulier, et non pas un groupe d'ordinateurs. Souvent payants, ils peuvent être contraignants et quelque fois très peu sécurisés. Les pare-feu nommé plus-sûr tournent généralement sous linux, car ils offrent une sécurité réseau plus élevée et un contrôle plus adéquat, ils ont généralement pour but d'avoir le même comportement que les pare-feu matériels des routeurs. Le plus courant est iptables (anciennement ipchains), qui utilise directement le noyau linux. Toute fonctionnalité des firewalls de routeurs est potentiellement réalisable sur une telle plateforme. [14]

II.3. Principe de fonctionnement

Un système Pare-feu contient un ensemble de règles prédéfinies permettant :

- D'autoriser la connexion [allow],
- De bloquer la connexion [deny],
- De rejeter la demande de connexion sans avertir l'émetteur [drop].

L'ensemble de ces règles permet de mettre en œuvre une méthode de filtrage dépendant de la politique de sécurité adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- Soit d'autoriser uniquement les communications ayant été explicitement autorisées : « tout ce qui n'est pas explicitement autorisé est interdit »,
- Soit d'empêcher les échanges qui ont été explicitement interdits.

La première méthode est sans nul doute la plus sûre, mais elle impose toutefois une définition précise et contraignante des besoins en communication.

II.4. Le filtrage de paquets

Dans une stratégie de sécurité réseau, il faut toujours séparer le réseau interne de l'extérieur (internet), tout en gardant les connexions nécessaires entre eux. Cette séparation est faite à l'aide d'un mécanisme de filtrage de paquets.

Le filtrage de paquets est l'un des meilleurs moyens pour éviter les tentatives de piratage. Il consiste à filtrer dès l'entrée du réseau tout ce qui n'est pas sensé y être utile.

II.4.1. Définition du filtrage de paquets

Tous les transferts de données à travers les réseaux se font sous forme de paquets. Le filtrage de paquets est un mécanisme de sécurité réseau qui fonctionne en contrôlant les données qui transitent en entrée et en sotie du réseau. Un filtrage de paquets est installé sur une machine du réseau, généralement, en un point d'accès entre l'extérieur (internet) et le réseau interne. La machine dotée de filtre de paquets analyse l'en-tête du paquet et décide de l'autorisation ou du rejet de paquets selon les conditions de la liste d'accès.

II.4.2. Fonctionnement du filtrage de paquets

Généralement, tous les filtres de paquets fonctionnent de la même manière. L'algorithme suivant matérialise la progression des paquets :

1. Les critères de filtrage doivent être définis, ils sont appelés règles de filtrage de paquets,
2. Analyse des champs d'en-tête des paquets IP, UDP ou TCP,
3. Application de la règle suivante,
4. Si la règle autorise la transmission du paquet alors le paquet est autorisé à entrer et aller à (7),
5. Si la règle n'autorise pas la transmission du paquet alors le paquet est rejeté,
6. Si c'est la dernière règle de filtrage alors le paquet est rejeté,
7. Reprendre les tests pour les prochains paquets qui arrivent. [15]

II.5. Les différents types de filtrage

Selon la nature de l'analyse et des traitements effectués par un Firewall, différents types de Firewalls existent. Ils se distinguent le plus souvent en fonction du niveau de filtrage des données auquel ils opèrent : niveau 3 (IP), niveau 4 (TCP, UDP) ou niveau 7 (FTP, HTTP, etc.) du modèle OSI.

Un système Pare-feu fonctionne sur le principe du filtrage simple de paquets. Il analyse les en-têtes de chaque paquet de données (datagramme) échangé entre une machine du réseau interne

et une machine extérieure transitent par le Pare-feu et possèdent les en-têtes suivants, systématiquement analysés par le Firewall :

- Adresse IP de la machine émettrice,
- Adresse IP de la machine réceptrice,
- Type de paquet (TCP, UDP, etc.),
- Numéro de port (rappel : un port est un numéro associé à un service ou une application réseau).

Les adresses IP contenues dans les paquets permettent d'identifier la machine émettrice et la machine cible, tandis que le type de paquet et le numéro de port donnent une indication sur le type de service utilisé. [15]

II.5.2. Filtrage dynamique

Le filtrage simple de paquet ne s'attache qu'à examiner les paquets IP indépendamment les uns des autres, ce que correspond au niveau 3 du modèle OSI. Or, la plupart des connexions reposent sur le protocole TCP, qui gère la notion de session, afin d'assurer le bon déroulement des échanges. D'autre part, de nombreux services initient une connexion sur un port statique, mais ouvrent dynamiquement (c'est-à-dire de manière aléatoire) un port afin d'établir une session entre la machine faisant office de serveur et la machine client.

Ainsi il est impossible avec un filtrage simple de paquet de prévoir des ports à laisser passer ou à interdire. Pour y remédier, le système de filtrage dynamique de paquet est basé sur l'inspection des couches 3 et 4 du modèle OSI, permettant d'effectuer un suivi des transactions entre le client et le serveur.

Un dispositif pare-feu de type stateful inspection est ainsi capable d'assurer un suivi des échanges, c'est-à-dire de tenir compte de l'état des anciens paquets pour appliquer les règles de filtrage. De cette manière, à partir du moment ou une machine autorisée initie une connexion à une machine située de l'autre côté du pare-feu. L'ensemble des paquets transitant dans le cadre de cette connexion sont implicitement acceptée par le pare-feu.

De ce fait, ils seront capables de traiter les paquets non plus uniquement suivant les règles définies par l'administrateur, mais également par rapport à l'état de la session :

– NEW : Un client envoie sa première requête.

– ESTABLISHED : Connexion déjà initiée. Elle suit une connexion NEW.

– RELATED : Peut-être une nouvelle connexion, mais elle présente un rapport direct avec une connexion déjà connue.

– INVALID : Correspond à un paquet qui n'est pas valide.

Les attributs gardés en mémoires sont les adresses IP, numéros de port et numéros de séquence des paquets qui ont traversé le firewall. Les firewalls à états sont alors capables de déceler une anomalieprotocolaire de TCP. De plus, les connexions actives sont sauvegardées dans une table des états de connexions.

II.5.3. Filtrage applicatif

Le filtrage applicatif permet comme son nom l'indique de filtrer les communications application par application. Ce filtrage opère donc au niveau 7 (couche application) du modèle OSI. Le filtrage applicatif suppose donc, une connaissance des applications présentes sur le réseau, et notamment de la manière dont les données sont échangées (ports, etc.).

Un Firewall effectuant un filtrage applicatif est appelé généralement passerelle applicative (ou Proxy), car il sert de relais entre deux réseaux en s'interposant et en effectuant une validation fine du contenu des paquets échangés.

Le Proxy représente donc un intermédiaire entre les machines du réseau interne et le réseau externe, subissant les attaques à leur place. De plus, le filtrage applicatif permet la destruction des en-têtes, précédant le message applicatif, ce qui permet de fournir un niveau de sécurité supplémentaire.

Il s'agit d'un dispositif performant, assurant une bonne protection du réseau, pour peu qu'il soit correctement administré. En contrepartie, une analyse fine des données applicatives requiert une grande puissance de calcul et se traduit donc souvent par un ralentissement des communications, chaque paquet devant être finement analysé.

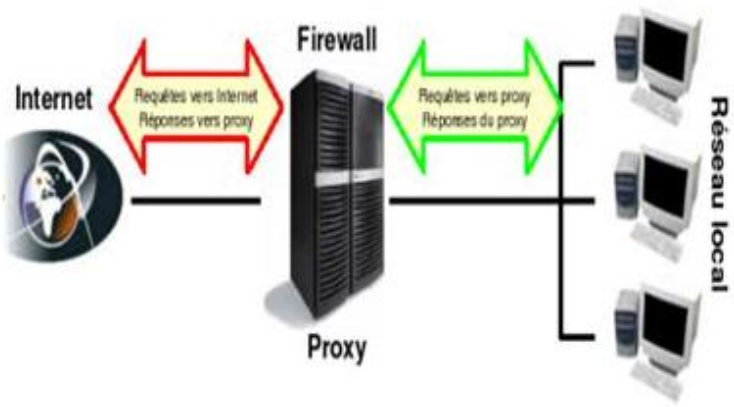


Fig. II.2: Firewall applicatif.

II.6. Types d'architectures

Le Firewall n'est pas seulement une solution logicielle de sécurité implantée sur une machine, c'est aussi une architecture réseau de machines filtrantes. L'approche simpliste d'un Firewall localisé sur une machine jouant le rôle de grand chef d'orchestre n'a plus cours à présent dans les grandes entreprises, car elle est trop peu sécurisée en cas de panne ou faille dans cette unique défense. La mise en place de plusieurs filtres de différents niveaux assurent une meilleure sécurité du réseau, mais ils s'accompagnent d'un coût plus élevé. [17]

II.6.1. Firewall avec routeur de filtrage

La solution Firewall la plus simple, mais aussi la moins sûre, se borne au réseau. On l'obtient en configurant le routeur qui assure la connexion avec l'Internet.

Le routeur doit être configuré avec une liste d'accès. L'image suivante illustre cette solution appelée Firewall avec routeur de filtrage :

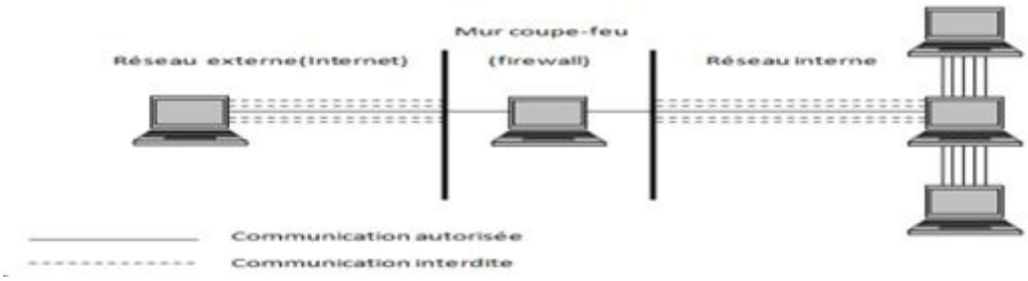


Fig. II.3 : Firewall avec routeur de filtrage.

Une liste d'accès définit les conditions pour qu'un paquet puisse franchir un routeur. Les informations contenues dans ces listes portent :

- les adresses IP,
- les numéros de ports,
- D'autres informations dans le paquet comme les drapeaux TCP,

Le type de la règle, c'est-à-dire soit une autorisation soit un refus de faire traverser le paquet.

Quand un paquet arrive sur le routeur, la liste est parcourue et le traitement du paquet est lié à la première condition rencontrée qui correspond au paquet. [17]

II.6.2. Passerelle double- le réseau bastion

Il existe une autre possibilité permettant de réaliser un Firewall d'application à peu de frais : la passerelle double. Comme son nom l'indique, il s'agit d'un ordinateur inclus à la fois dans les

deux réseaux Internet et Intranet. Cette machine doit être équipée de deux cartes réseau. Comme elle est la seule soupape de sécurité entre les deux réseaux, elle doit être configurée avec le plus grand soin.

La passerelle double n'autorise aucun trafic IP entre les réseaux. On l'appelle également réseau bastion, car il contrôle tous les services accessibles de l'extérieur comme de l'intérieur du réseau interne tels que les serveurs Web, FTP et Mail. Un " Serveur Proxy " supplémentaire est également configuré pour permettre aux utilisateurs du réseau interne d'accéder à Internet. Le nom "réseau bastion" découle des mesures particulières de protection qui sont prises en prévision de possibles intrusions. [17]

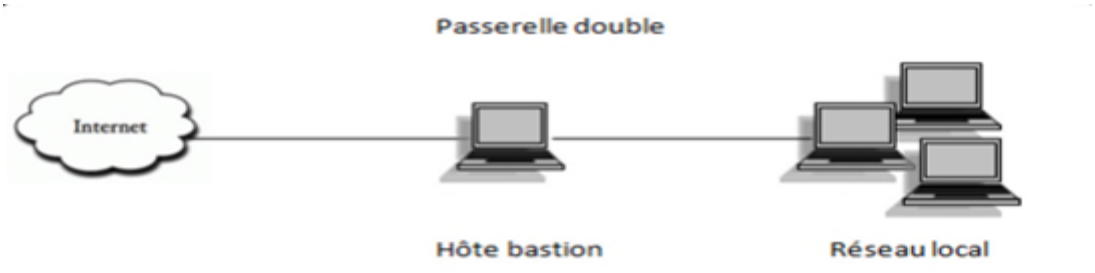


Fig. II.4 : Passerelle double.

La passerelle double est la possibilité la plus simple pour réaliser un Firewall d'application n'autorisant aucun trafic IP entre les réseaux.

II.6.3. Firewalls avec réseau de filtrage

La combinaison des deux méthodes est ici plus sûre et efficace. Au niveau du réseau, un routeur sous écran est configuré de façon à n'autoriser les accès de l'extérieur et de l'intérieur que par l'intermédiaire du réseau bastion sur lequel fonctionnent tous les serveurs assurant les serveurs Internet. Cette possibilité est appelée Firewall avec réseau de filtrage. L'image suivante illustre cette solution :

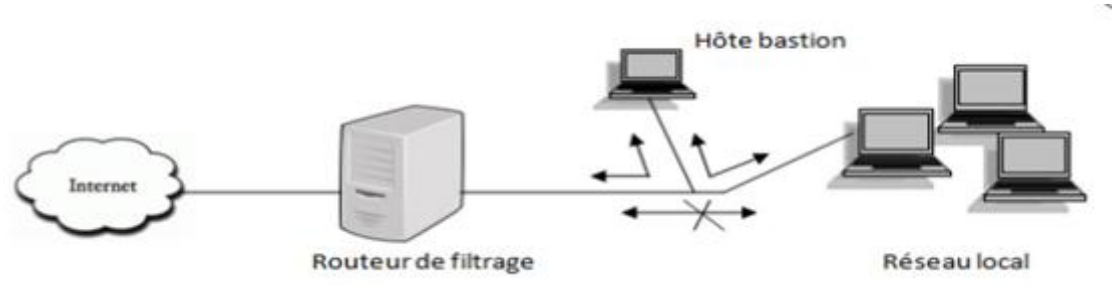


Fig. II.5 : Firewalls avec réseau de filtrage.

Firewall avec réseau de filtrage dans lequel seuls les accès au réseau bastion sont autorisés.

Pour la grande majorité des entreprises, cette solution est sûre et abordable, car les prestataires Internet assurent la seconde partie de la protection à l'autre bout de la ligne. En effet, votre entreprise y est également connectée à un routeur, et le trafic de données est réglé par un serveur Proxy au niveau de la couche application. Les pirates doivent par conséquent franchir deux obstacles. [17]

II.6.4. Firewall avec sous-réseau de filtrage

Cette solution est de loin la plus sûre, mais également la plus onéreuse. Un Firewall avec sous-réseau de filtrage se compose de deux routeurs sous écran. L'un est connecté à Internet, et l'autre à l'intranet/LAN. Plusieurs réseaux bastions peuvent s'intercaler pour former entre ces deux routeurs, en quelque sorte, leur propre réseau constituant une zone tampon entre un Intranet et l'Internet appelée " zone démilitarisée ". De l'extérieur, seul l'accès aux réseaux bastions est autorisé. Le trafic IP n'est pas directement transmis au réseau interne. De même, seuls les réseaux bastions, sur lesquels des serveurs Proxy doivent être en service pour permettre l'accès à différents services Internet, sont accessibles à partir du réseau interne. L'image suivante illustre cette variante : [17]

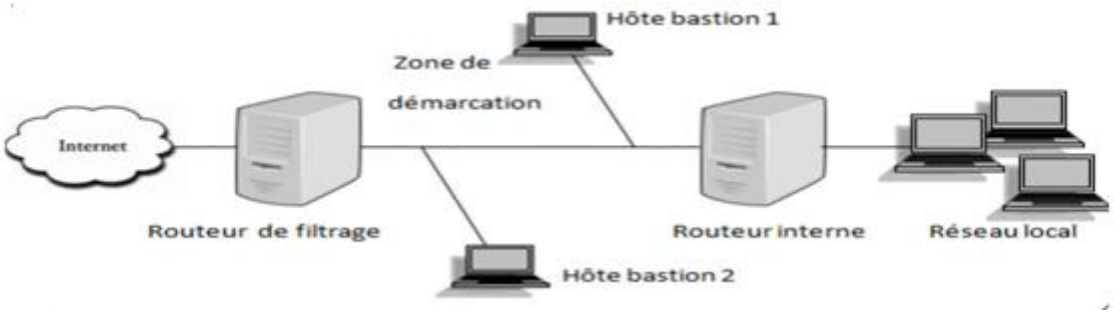


Fig. II.6 : Firewall avec sous-réseau de filtrage.

II.7. Les pare-feu applicatifs

- **Préliminaire**

II.7.1 Application Web

Une application web est une application manipulable grâce à un navigateur web. De la même manière que les sites web, une application web est généralement placée sur un serveur et se

manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, etc.). [18]

Fig. II.7 : Application Web à architecture 3 tiers.

II.7.2. Serveur Web

Un serveur web est un logiciel informatique qui permet d'héberger un ou plusieurs sites internet. Il assure donc la communication avec le navigateur internet utilisé par un internaute (grâce au Protocol réseau HTTP). Un serveur Web est généralement capable de gérer à la fois du contenu statique (un logo, une page HTML simple) ou dynamique (contenu extrait de base de donnée), les serveurs Web les plus connus sont Apache, IIS,Lighthttp . [19]

II.7.3. Serveur d'application

Un serveur d'application est un environnement informatique qui fournit les briques nécessaires à l'exécution d'applications transactionnelles sur le web.

Il doit répondre à cinq critères techniques :

- S'interfacer avec un serveur http (HTML, XML).
- Fournir un moteur d'exécution des traitements (ex : Java Virtual Machine).
- S'ouvrir sur le système d'information de l'entreprise (XML, Web service, connecteur SGBDR, etc.).
- Permettre l'ajout de briques techniques et métier
- Répondre aux contraintes induites par les architectures centralisées :
    - Gestion de contextes (différenciation des clients/ temps de session par le biais de Cookies, d'URL long ou encore de variable cachée).
    - La répartition de charges (exécution de plusieurs instances réparties sur différentes machines) et le pooling de connexions (évitant de création de goulet d'étranglement).

- Les reprises sur incident (l'application est répliquée sur plusieurs serveurs physiques. En cas de « plantage » au niveau applicatif ou serveur, la requête utilisateur est redirigée vers un serveur disponible de manière transparente). [20]

II.7.4. Protocole HTTP

Le protocole HTTP est un protocole de couche application, il permet la communication entre le navigateur web de l'internaute et un serveur dans un format spécifique orienté requête/réponse.[21]

a. **Une requête HTTP** est une demande de ressource (une page HTML par exemple) ; émise par le client via son navigateur, en cliquant sur un lien, ou bien en saisissant l'adresse d'un site Web. [21]
b. **La réponse HTTP** à cette demande contient la ressource demandée, ou bien une page d'erreur dans le cas où cette ressource n'existe pas, ou si son accès est protégé par exemple.

Ces ressources sont accompagnées d'un code d'état HTTP, permettant de savoir si la demande a abouti correctement, ou bien dans le cas contraire, les raisons de l'erreur.

Lorsqu'un serveur répond à une requête HTTP, il renvoie, en plus des ressources de type page HTML et image, un code d'état et des informations de contrôle.

Ces codes d'état permettent de connaitre l'issue de la conversation entre le client navigateur web et le serveur.

Les codes d'erreurs sont organisés par catégorie de réponse :

- Succès (200 OK, etc.).
- Redirection (301 ressource déplacée de manière permanente, 302 ressource déplacée de manière temporaire, etc.).
- Erreur du client (403 traitement refusé, 404 document non trouvé, etc.).
- Erreur du serveur (500 erreur interne, etc.).
- Http utilise la notion d'URL pour permettre la localisation d'une ressource sur un serveur Web.

Les URL en HTTP ont la syntaxe suivante :

http://<adressedu serveur> :<port du serveur>/<chemin>/<ressource>

Dans cette URL, le port du serveur est facultatif s'il vaut 80, de plus le chemin et le nom de la ressource doivent être saisis en respectant la distinction majuscule/minuscule.

Le protocole HTT¨P offre plusieurs possibilités pour gérer le transfert des informations entre le client et le serveur, appelées méthodes HTTP. Les méthodes HTTP les plus utilisées sont la méthode GET pour émettre une demande de ressource telle une page HTML, ou encore la

méthode POST pour transmettre des données à destination du serveur en utilisant un formulaire HTML. [21]

II.7.5. Analyse d'un exemple de requête et réponse http

a. Requête d'un client

Toute requête HTTP est basée sur le format d'échange de données MIME (Multipurpose Internet Mail Extension), qui comporte un entête et un corps.

Dans le cas d'une requête HTTP, seul l'entête est utilisé, le corps reste vide (sauf pour les requêtes de type POST)

Les réponses HTTP quant à elles utilisent les deux. Dans tous les cas, entête et corps sont séparés par une ligne vide.

Voici un exemple de requête HTTP :

```
<font color="#000080">GET /index.html HTTP/1.1
Host: www.mtds.com
User-Agent: foobar
Referer: www.kernel.com
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr-F,fr;q=0.5</font>
```

Fig. II.8 : Exemple de requête http.

Depuis la version 1.1 de HTTP, les deux premières lignes de l'entête HTTP sont devenues impératives. Les autres lignes sont indicatives ou servent à la négociation de contenu (préférences en format multimédias, en termes de langage et autres).

b. Réponse envoyée par le serveur

Voici une réponse possible du serveur suite à la requête décrite précédemment :

```
<font color="#000080">HTTP/1.1 200 OK
Date: Mer, 11 juin 2014 10:13:21 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) PHP/4.3.10-12
Content-Length: 305
Content-Type: text/html; charset=iso-8859-1
<html>Bonjour</html></font>
```

Fig. II.9 : Exemple de réponse du serveur.

Le serveur confirme qu'il parle HTTP en version 1.1, donne un code de succès (200 OK) et spécifie diverses informations, dont la plus importante est le champ Content-Type. Le navigateur utilise en effet ce champ pour interpréter la réponse. Ici, la valeur texte/html ce champ indique au navigateur d'interpréter cette page comme du code HTML. D'autres transformations sont également effectuées.

II.7.6. Le protocole HTTPS

Quand vous naviguez sur internet, vous surfez sur des pages dont le titre commence toujours par « http ». Or ce protocole de communication n'est pas sécurisé et votre connexion comporte de nombreuses données personnelles. C'est pourquoi, vous observerez que certaines
Pages sont hébergées sous le protocole « https ». Nous découvrirons en quelques lignes les critères de sécurité des sites sur lesquels vous laissez vos données.
Le protocole « http » n'est soumis à aucun chiffrement. C'est pourquoi le protocole
« https »Pour «http» 'sécure'a été inventé. Quand vous surfez sur un site ou certaines pages de sites, notamment les pages de transaction des sites marchands ou sur les sites bancaires, vous observerez que la racine de la page commence par « http ».

II.7.6. Les modes de chiffrement HTTPS

Le protocole « https » signifie que la communication entre vous et le serveur web du site est Chiffrée. Le flux de cette communication peut être plus ou moins fortement chiffré ou encrypté.
Ilexiste plusieurs modes de chiffrement : SSLv2, SSLv3 et TLS.
Le protocole SSL peut quant à lui être symétrique ou asymétrique
Le protocole TLS est un mode de chiffrement asymétrique
Pour plus de sécurité, les protocoles asymétriques sont préférés car ils rendent moins facile la Possibilité d'accéder aux données véhiculées par les flux d'un utilisateur, c'est-à-dire vos mots de passe ou autres informations personnelles voire confidentielles.

II.8. Définition d'un pare-feu applicatif

Un WAF (Web Application Firewall) est un logiciel ou un équipement matériel qui permet principalement de protéger les applications Web contre des attaques applicatives (SQL injection, Cross Site Scripting, injection de code),
Ce composant permet :

- De bloquer des attaques en implémentant des filtres au niveau des données transportées par HTTP/HTTPS (HTML et XML).
- De limiter l'exposition à des vulnérabilités détectées dans la logique applicative en attendant une correction en profondeur.
- D'authentifier et/ou d'autoriser l'accès à toute ou partie de l'application.
- De tracer pour apporter la visibilité supplémentaire sur l'environnement applicatif (facilite l'investigation, apporte des preuves de tentatives d'intrusion, donne des informations sur le temps de réponses des serveurs web).

II.9. Mode de déploiement d'un firewall applicatif

Pour les firewalls qui sont des Appliance, il y a généralement deux façons de les placer :

1. Placer le WAF directement derrière un pare-feu classique et avant le serveur web hébergeant l'application web. Etant dans cette position, tout le trafic web passe à travers lui.
2. Positionner le pare-feu sur un port de surveillance du réseau (Network Monitoring port) ce qui le permet de recevoir une copie du trafic. [22]

II.10. Modèles de sécurité

Un pare-feu applicatif suit généralement soit un modèle de sécurité négatif ou positif.

- Le modèle de sécurité négatif : Un modèle de sécurité négatif surveille les demandes d'anomalies, de comportements inhabituels et d'attaques Web courantes. Il conserve les scores d'anomalie pour chaque requête, adresse IP, session d'application et compte d'utilisateur. Les requêtes avec des scores d'anomalie élevés sont soit journalisées, soit totalement rejetées.
- Le modèle de sécurité positif Lorsqu'un modèle de sécurité positif est déployé, seules les requêtes connues pour être valides sont acceptées, avec tout le reste rejeté. Ce modèle nécessite la connaissance des applications Web que vous protégez. Par conséquent, un modèle de sécurité positif fonctionne mieux avec les applications fortement utilisées, mais rarement mises à jour, afin de minimiser la maintenance du modèle.

II.11. Modes de fonctionnement

a. Mode passif : le firewall écoute le trafic sur un port de control. Ce mode est parfait pour tester le fonctionnement du WAF sans nuire le trafic mais n'influence pas de dernier. Le trafic est répliqué au niveau de la couche 1 du modèle ISO/OSI.

b. Mode bridge : le WAF fonctionne comme un routeur au niveau 2 (couche liaison de données) du modèle ISO/OSI.

c. Mode routeur : le WAF agit comme un routeur au niveau 3(couche réseau) du modèle ISO/OSI donc le WAF est en ligne directe avec le trafic Internet.

d. Mode reverse-proxy : le mode reverse proxy est le déploiement le plus courant et riche en fonctionnalités. En ce mode, le WAF se trouve en ligne directe avec le trafic venant de l'internet. Le WAF a une adresse IP publique et toutes les requetés concernant les applications web lui y sont directement adressées. Le WAF ensuite fait transmission au serveur web au nom du navigateur d'origine.

e. **Mode embarqué :** dans ce modèle le WAF est embarqué dans le serveur web comme un module logiciel. Cette approche a certains avantages pour le déploiement des applications web de complexité réduite parce qu'aucun matériel supplémentaire n'est nécessaire. [22]

II.12. Etat de l'art et choix du firewall applicatif

Le WASC « Web Application Security Consortium » est une organisation à but non lucratif qui a comme but de faire connaitre les meilleures pratiques concernant la sécurité du web. Le WASC a établi en collaboration avec OWASP le projet WAFEC (the Web Application Firewall Evaluation Criteria) dont le but est de développer une analyse détaillée des critères d'évaluation des WAF et la mise en place d'une méthodologie de test qui peut être utilisée par n'importe quel technicien raisonnablement compétent pour évaluer indépendamment la qualité d'une solution WAF. [23]

Le WAFEC sert deux objectifs : d'une part il aide les utilisateurs à comprendre ce qu'est un WAF et quel est son rôle dans la protection des sites web et d'autre part, WAFEC fournit un outil pour les administrateurs de sécurité, qui les aide à prendre une décision éclairée lorsqu' ils choisissent un WAF. [23]

Le WAFEC permet d'évaluer techniquement le meilleur WAF pour son environnement en fonction de 9 critères :

1. type d'architecture à déployer (pont, reverse-proxy, SSL, etc.),
2. support d'HTTP et d'HTML (version, encodage, etc.),
3. Techniques de détection (signatures, techniques de normalisation du trafic, etc.),
4. Technique de protection (brute force, cookies, sessions, etc.),
5. Journalisation (type de logs, gestion des données sensibles, etc.),
6. Rapport (type de rapports, distribution, format, etc.),
7. Administration (politiques. logs) ,
8. Performance,
9. Support XML.

Dans ce qui suit nous allons décrire brièvement deux solutions commerciales et deux open sources les plus réputées. [23]

II.13. Conclusion

Il y'a plusieurs types d'architectures de Pare-feu, chacune d'elle présente ses inconvénients et ses avantages. Donc pour la mise en place d'une architecture Firewall on a toujours recours à

revoir ces différentes architectures et choisir une selon les besoins, les moyens, et la politique de sécurité que l'entreprise souhaite voir respectée.

III. Introduction

L'étude de l'existant au niveau de l'organisme d'accueil, nous permet prendre contact avec ses éléments et comprendre l'architecture réseau de ses équipements est la garantie vers une solution fiable et cohérente.

Nous structurons ce chapitre comme suit :

Dans la première partie nous présentons la société d'accueil « service des réseaux et système informatique » au sein duquel nous avons effectué notre projet, dans la seconde section nous identifions les anomalies du système existant, auxquelles nous apportons des solutions adéquates et dans la troisième partie nous analysons les choix pour les solutions WAF.

III.1. Historique

Le centre Universitaire de Tizi-Ouzou est créé en 1977 (décret exécutif No 17-77 du 20 juin 1977). La première rentrée universitaire avait accueilli 490 étudiants dont une cinquantaine de nationalités étrangères, encadrés par 27 jeunes qui y firent leur entrée en 1977. Le C.U.T.O avait alors démarré avec (05) départements :

- Département des Sciences Exactes
- Département de Biologie
- Département des Sciences Juridiques et Administratives
- Département de Langues et Littérature Arabes
- Département des Sciences Économiques

En 1984, le centre universitaire (C.U.T.O), éclate en neuf (09) Instituts Nationaux d'Enseignement Supérieur (I.N.E.S).

- I.N.E.S de Génie Civil
- I.N.E.S des Sciences Médicales
- I.N.E.S des Sciences Juridiques et Administrative
- I.N.E.S des Sciences Économiques - I.N.E.S de Biologie - I.N.E.S de Langue et Littérature Arabes
- I.N.E.S d'Électrotechnique
- I.N.E.S d'Informatique
- I.N.E.S d'Agronomie

En 1989, cet important pôle a été élevé au rang d'université (U.T.O) par décret exécutif N° 89-139 du 01/08/1989

En 1991, L'Université de Tizi-Ouzou enrichit son offre en formation par la création du département de langue et cultures amaigries par l'arrêté Ministériel N° 11 du 1/1990.

III.1.1. Structure de l'Université

Les structures de l'UMMTO (Université Mouloud Mammeri de TiziOuzou) sont implantées sur six (06) sites, il s'agit des campus suivants :

- Hasnaoua I, abritant le rectorat et les services centraux, la faculté des sciences économiques et de gestion, la faculté des lettres et des langues ainsi que cinq laboratoires de recherche
- Hasnaoua II, il abrite essentiellement les filières technologiques à savoir la faculté du Génie Électrique et Informatique et la faculté du Génie de la Construction, la faculté des Sciences Biologiques et Agronomiques et la faculté des sciences fondamentales.
- Le campus Biomédical, abrite la faculté des Sciences Médicales et quatre laboratoires de recherche
- Le site de Tamda, le plus récent, Il abrite la faculté des sciences humaines et sociale
- Le site Habitat, est exploité par les étudiants en TACT
- Le site de Boukalfa, il, regroupe les départements de la faculté de droit et des sciences politiques.

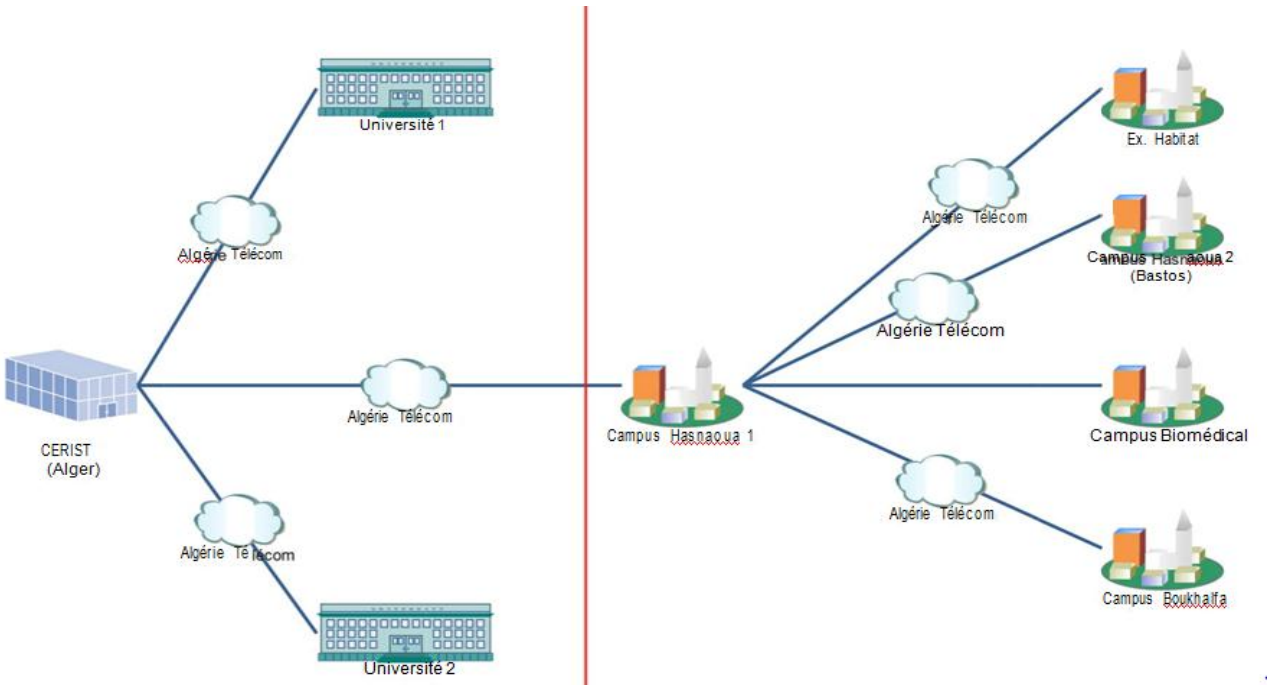


Fig. III.1 : Structure de l'UMMTO.

L'Université comprend : Le rectorat, les services communs et de nombreuses Facultés réparties sur plusieurs pôles : Hasnaoua I, Hasnaoua II, Boukhalfa et Tamda.

III.1.2. Organisation de l'UMMTO

L'UMMTO est organisée suivant l'organisme ci-dessous :

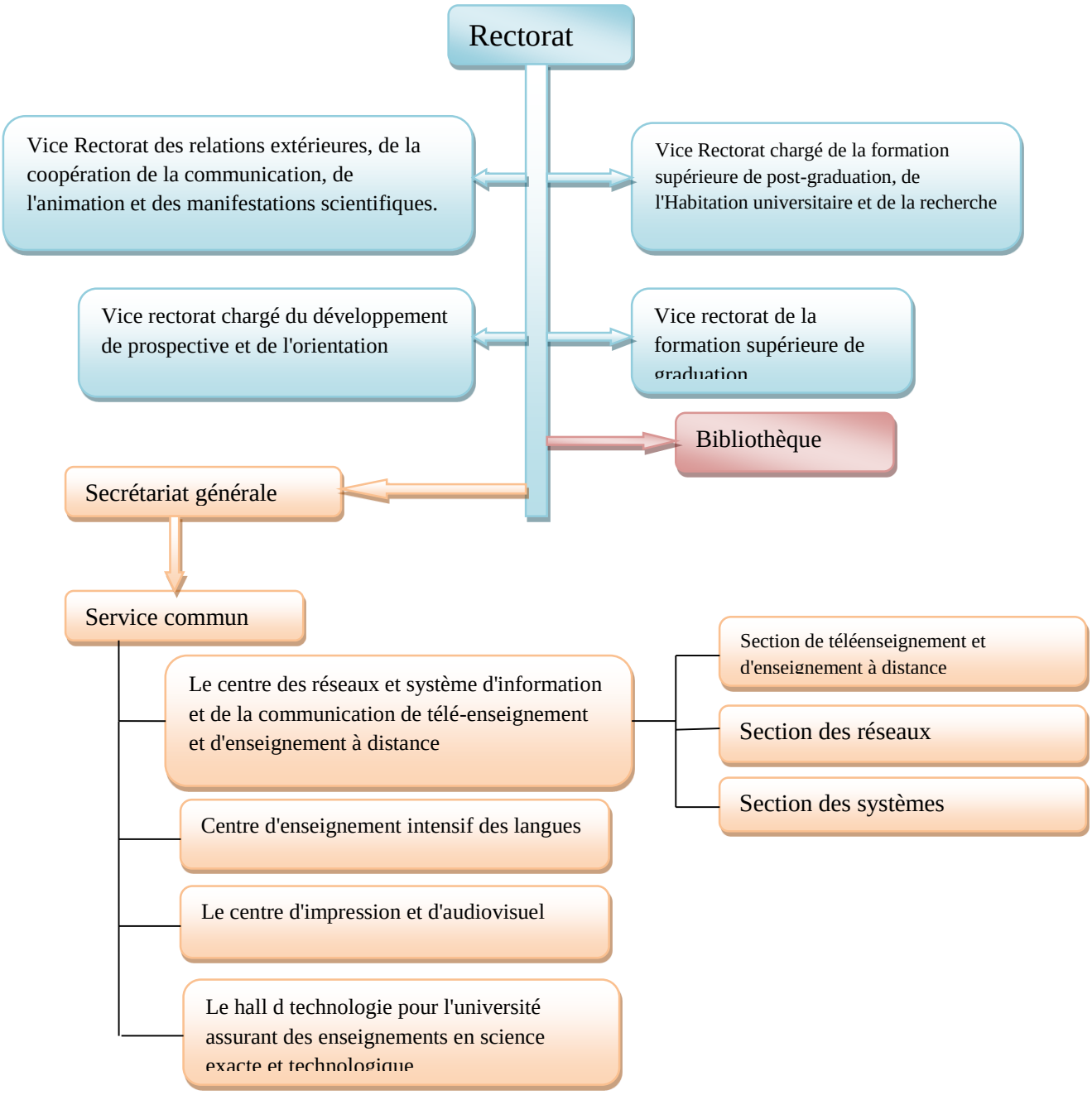


Fig. III.2 : Organigramme de l'UMMTO.

III.1.3. Structures de recherche

En plus de la formation notre université dispose de 29 laboratoires de recherche nous citerons parmi eux :

- Laboratoire de Ressources Naturelles
- Laboratoire de Technologies Avancées en Génie Électrique
- Laboratoire de Réformes Économiques & Dynamiques Locales
- Laboratoire de Production, Amélioration & Production des Végétaux & des Denrées
- Laboratoire de Mécanique, Structure & Énergétique (LÈSE)
- Laboratoire de Néo matériaux Environnement & Aménagement (LOGEA)

III.1.4. Présentation de l'organisme d'accueil

Le centre des Systèmes et Réseaux d'Information, de Communication, de Télé-enseignement et Enseignement à Distance est créé suite à l'arrêté interministériel du 24 Aout 2004, fixant la nouvelle organisation administrative de l'université et ses services communs, comme illustré dans la figure ci-dessous:

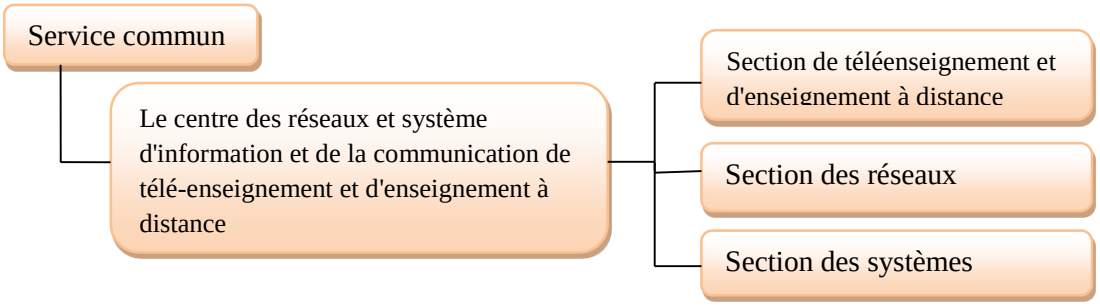


Fig.III.3 : organigramme de l'organisme d'accueil.

III.1.5. Présentation du centre

Le centre des Systèmes et Réseaux d'Information, de Communication, de Télé-enseignement et Enseignement à Distance est chargé de :

- L'exploitation, l'administration et la gestion des infrastructures de réseaux
- L'exploitation et le développement des applications informatiques de gestion pédagogique
- Le suivi et l'exécution des projets de Télé-enseignement et enseignement à distance
- Assurer l'appui technique à la conception et à la production de cours en ligne

- La formation et l'encadrement des intervenants dans l'enseignement à distance

Chaque département, chaque laboratoire de recherche et chaque service administratif du rectorat et des facultés se sont vus dotés de l'outil internet. Les trois composants de l'université (étudiant, enseignant et administration) bénéficient ainsi de ce service, devenu une nécessité pour le bon déroulement des différentes activités de notre établissement.

III.1.6. Organisation du Centre de calcul et réseau

Le centre de calcul et réseau est reparti en trois sections :

III.1.6.1. Section des systèmes

La Section Système d'Information (S.I), a pour mission de mettre en œuvre la politique des systèmes d'information et des technologies de l'information et de la communication, la gestion d'une manière plus générale à tout ce qui touche au traitement automatique de l'information.

III.1.6.2. Section des réseaux

La section réseau a pour missions de maintenir le fonctionnement normal du réseau intranet de l'université :

- D'assurer la sécurité des équipements réseaux et des services offerts par le réseau au système d'information et aux applications,
- Fournir des services de connexion internet, de messagerie électronique, de support utilisateur,
- D'étude et de suivi des projets réseaux de l'université Mouloud Mammeri.

III.1.6.3. Section de Télé-enseignement et Enseignement à Distance

Le domaine technique englobe la mise en place d'une solution e-learning répondant à la fois aux besoins et aux ambitions de cette université. Il s'agit notamment de l'installation, de l'administration et de la maintenance des plates formes de e-learning. En plus de cela, cette cellule gère une salle de visioconférence.

III.1.7. Personnels Administratifs et Techniques

Le personnel du service des réseaux, géré par le responsable du centre, compte actuellement dix-neuf (19) dont 06 ingénieurs d'état en informatique, 05 ingénieurs principaux 07 techniciens supérieurs en informatique et une Secrétaire de Direction. Ils interviennent dans la mise en place de l'intranet et participent à l'intégration des nouvelles technologies de la communication dans les différents services administratifs et les structures pédagogiques. En

plus de l'exploitation locale du service internet au niveau des différentes structures de notre université, le service de messagerie électronique permet l'ouverture d'une boite aux lettres électronique (e-mail) à chaque enseignant, disposant d'un compte sur le réseau de l'Université.

III.1.8. Etude de l'existant

L'étude de l'existant consiste à identifier les informations et les procédures utiles à la compréhension du système d'information.

Le but de cette étape est d'analyser la situation existante, de collecter les informations circulantes au sein de notre domaine d'étude et aussi mieux comprendre son organisation.

III.1.9. Description de l'existant

Le réseau de l'université met en œuvre des données sensibles, les stocke, les partage en interne et les communique parfois à d'autres universités ou personnes.Cette ouverture vers l'extérieur aux bénéficesdes informatisations, d'isoler le réseau de l'extérieur ou de risquer la confidentialité des données de l'université.

L'architecture du réseau de l'université se présente comme suit :

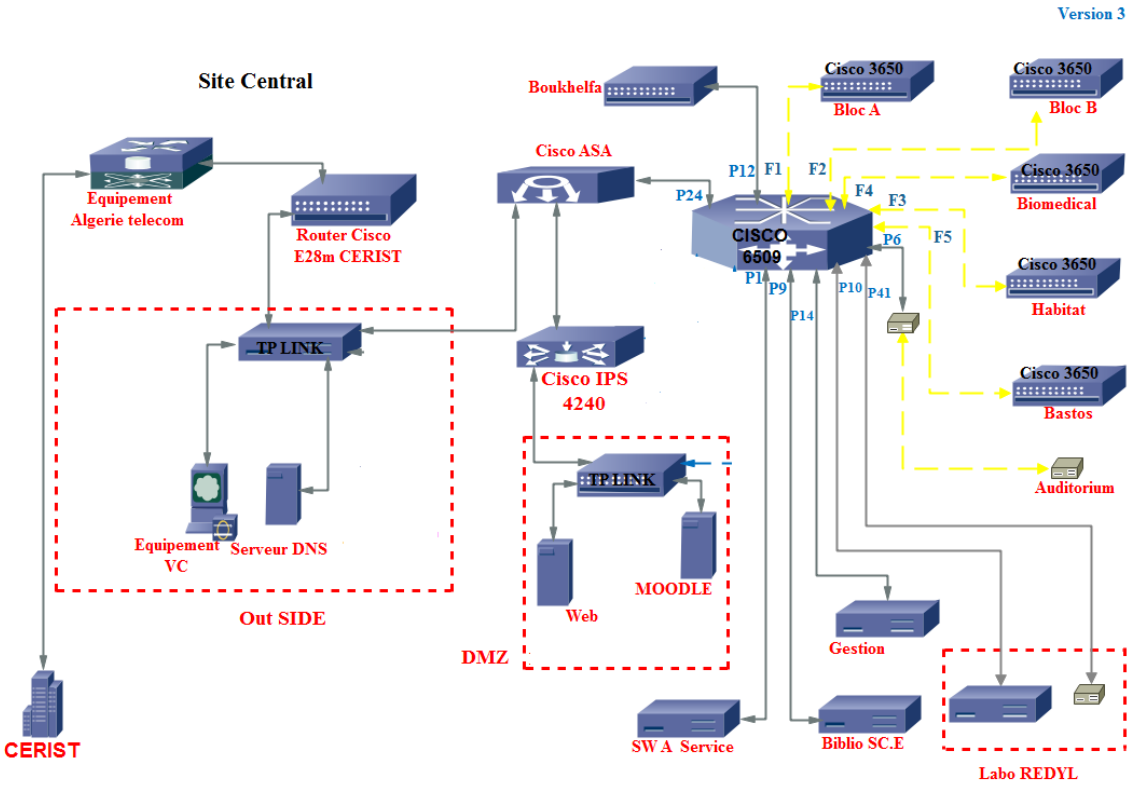


Fig.III.4 : Architecture réseau de l'UMMTO.

D'après cette architecture nous constatons que :

- L'UMMTO est constituée de 7 sites distants tel que l'habitat, HasnaouaI, HasnaouaII, etc. Dont chaque site représente un LAN qui est relié via la fibre-optique au routeur central Cisco 6509.
- La zone DMZ : (zone démilitarisé) est composée d'un ensemble de serveurs séparés du LAN et d'Internet par un pare-feu Cisco ASA 501 et un IPS Cisco 4240.
- La zone out Side : les serveurs sont lisibles et ils sont accessible par internet.

III.1.10. Critique de l'existant

L'étude de l'existant nous a permis de recenser des insuffisances du système existant qui se résument aux points suivants :

- L'architecture de l'UMMTO est dotée d'un IPS CISCO qui est mis à l'écoute (transfert des paquets), et d'un pare-feu ASA basique
- L'architecture de l'UMMTO possède un nombre important de postes et de serveurs qui rend la gestion de ce réseau une tâche délicate,
- L'ouverture de l'université vers l'extérieur est indispensable et dangereuse en même temps et peut laisser place aux étrangers pour pénétrer au réseau local de l'université, et notamment accomplir des actions douteuses de destruction ou de vol d'informations.
- L'université présente aussi un manque de contrôle et de journalisation du trafic entre le réseau interne et externe.

III.1.11. Solution proposée

La gestion des serveurs distants et le monitoring de ses équipements étant le plus grand souci de l'administrateur. Nous avons jugé nécessaire de mettre en évidence un outil pour contrôler le fonctionnement du réseau, étudier les données collectées et de définir des seuils d'alertes qui peuvent servir pour le déclenchement des alertes lors de détection des problèmes.
Il s'agit donc et sans doute d'une mise en place d'un composant firewall. Notre choix porte sur le firewall applicatif open source qui pourra, grâce à ses différentes fonctionnalités,

1. D'apporter la sécurité nécessaire au réseau locale de l'université et notamment aux serveurs web et de détecter les tentatives d'intrusion.
2. Le firewall propose donc un véritable contrôle sur le trafic réseau de l'université. Il permet d'analyser, de sécuriser et de gérer le trafic réseau, et d'utiliser ainsi convenablement le réseau de l'université.

La nouvelle architecture proposée pour le réseau de l'université est la suivante :

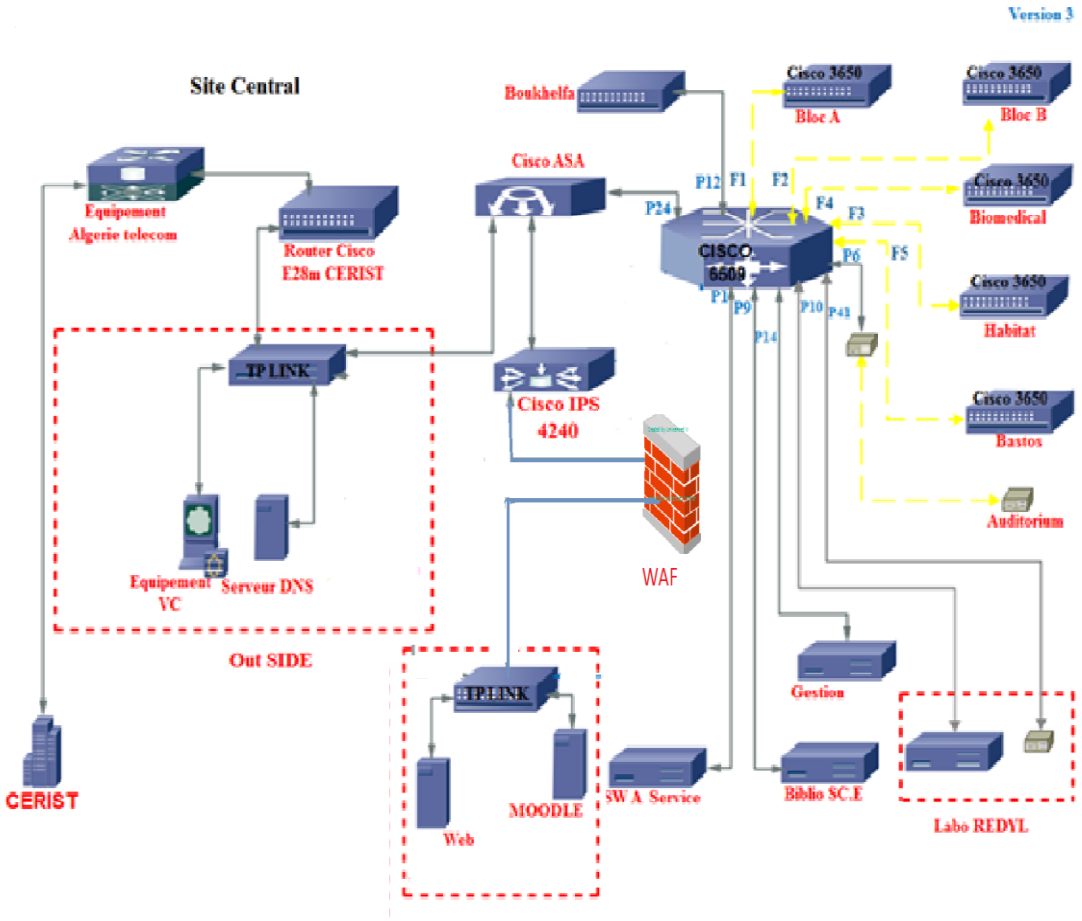


Fig.III.5 : architecture de l'UMMTO avec le pare-feu WAF.

III.2. Les solutions WAF

Il existe deux types de solutions WAF les solutions commerciales et les solutions open source.

III.2.1 Solutions commerciales

a. Imperva

Le pare-feu applicatif SecureSphere d'Imperva est l'un des appareils les plus réputés du marché.Selon Gartner, Imperva « apparait le plus souvent sur la liste des produits préférés par les clients de gartner. »

Imperva, utilise ce que la société appelle 'l'inspection transparente', une technologie pour combiner la sécurité avec la haute performance.

Les politiques de sécurité sont basées sur un modèle de sécurité positive et Imperva à également des options pour le suivi de base de données de vulnérabilités. [24]

b. F5

Le pare-feu applicatif f5 vendu par la société F5 comme module complémentaire pour sa ligne des produits BigIP Application Delivery Controllers. Le module WAF utilise un modèle de sécurité positive pour la définition des politiques de sécurité et est livré avec toutes les fonctionnalités que sont attendus d'un WAF prêts pour les entreprises. Une des caractéristiques les plus intéressantes est l'intégration avec Whitehead Sentinel Vulnérabilités trouvées à partir d'un scan. [25]

III.2.2 Solutions Open Source

a. ModSecurity

Modsecurity est un module d'apache qui est disponible sous licence GPLv2, cette solution fonctionne au niveau de la couche application sur le protocole HTTP, elle contient un moteur de détection et de prévention d'intrusion pour les applications.

Ce moteur se base sur des règles de filtrage et de signatures. Modsecurity a connu un succès grandissant, relayé par une communauté d'utilisateurs compétente.il a fini par supplanter certaines solutions commerciales. [26]

b. AQTRONIX WabKnight

AQTRONIX WebKnight est un pare-feu d'application conçue pour IIS et plusieurs autres serveurs web et est publié sous la licence GNU General Public License. Il s'agit d'un filtre ISAPI qui sécurise votre serveur web en bloquant certaines demandes. Si une alerte est déclenchée WebKnight prendra le relais et de protéger le serveur web. Cette protection se fait en balayant toutes les demandes et traitant ces dernières à l'aide des règles de filtrage, définies par l'administrateur. Ces règles ne sont pas basées sur une base de données de signatures d'attaque qui nécessitent des mises à jour régulières. WebKnight utilise des filtres de sécurité en cas de débordement de la mémoire tampon, d'injection SQL, de traversée de répertoire, d'encodage ou toute autre attaque. De cette façon webKnight peut protéger les servers contre les attaques connues et inconnues. [27]

III.2.3Etude de choix

Dans cette étape, nous allons réaliser l'étude de choix des solutions WAF à l'aide de la matrice SWOT.

III.2.3.1 L'analyse SWOT

L'analyse SWOT (Strngths/Weaknesses and Opportunities/Threats, en Français Forces/Faiblesses et Opportunités/Menaces) est telle que définie par les services de la « Commission européenne »

« Un outil d'analyse stratégique. Il combine l'étude des forces et des faiblesses d'une organisation, d'un territoire, d'un secteur, etc. avec celle des opportunités et des menaces de son environnement, afin d'aider à la définition d'une stratégie de développement »

Il s'agit d'un outil permettant l'identification des points forts, des faiblesses, des opportunités et des menaces d'une entreprise particulière. Les points forts et les faiblesses constituent les facteurs internes qui créent la valeur ou la détruisent. Les opportunités et les menaces constituent les facteurs externes qu'une entreprise ne peut pas contrôler.

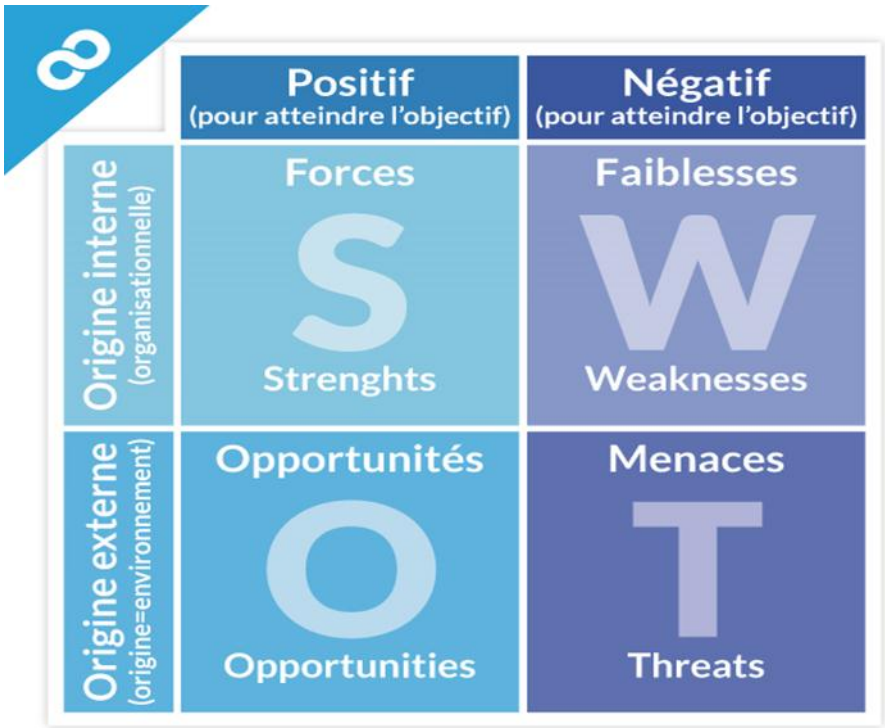


Fig.III.6 : La matrice SWOT.

Force : une ressource ou caractéristique du produit ou de l'organisation étudiés qui sert à être mise en valeur.

Faiblesse : une limite, un défaut ou une « non compétence » du produit ou de l'organisation qui va l'empêcher de parvenir au succès.

Opportunité : toute situation favorable à une entreprise pour parvenir à se donner un avantage concurrentiel sur le projet/produit.

Menace : toute situation non favorable dans l'environnement extérieur qu'est une menace pour l'évolution du projet.

1. Application de l'Analyse SWOT sur les Solutions Commerciales

De nos jours, les solutions WAF reconnaissent une très grande popularité et utilité, c'est pour cela que les sociétés n'hésitent pas à investir dans des WAF. L'inconvénient de ces solutions propriétaires de sécurisation des applications web est le coût élevé

Force	Faiblesse
-dispose d'une très forte notoriété et d'une bonne image de marque. - un offre de fonctionnalités diversifiés et adaptés à la demande clientèle.	-Redevance annuelle. - cout très élevé
Opportunité	Menace
-la forte demande par les grandes entreprises. -le domaine est concurrentiel et en cours de développement.	-L'apparition des outils libre qui englobent toutes les caractéristiques des outils propriétaires.

Tab.III.1. Matrice SWOT pour la solution commerciale.

1. Application de l'analyse SWOT sur les solutions Open Source

Il existe divers pare-feu applicatifs libres qui garantissent la gratuité et la disponibilité du code source. Les plus connus, sont ModSecurity, Webknight,

Force	Faiblesses
- Code source accessible à une large gamme de développeurs de ce type de solutions réparation très rapide et efficace des bugs et des erreurs de programmation. - Stabilité : les WAFs open source sont réputés plus stables que les WAFs commerciaux. - Pas de redevance annuelle.	- Mise à jour non fournie automatiquement.
Opportunités	Menace
- Augmentation des pare-feu d'application.	- L'apparition des nouveaux outils plus performants

Tab.III.2 : Matrice SWOT pour les solutions open source

2. Synthèse

Les solutions commerciales fournissent une protection contre les attaques applicatives avecun coût très élevé, alors que l'utilisation des solutions open source permet d'avoir une protection gratuite.

3. Application de l'Analyse SWOT pour ModSecurity

Force	Faiblesse
-Pas de redevance annuelle -Flexibilité. -Intégré dans le serveur web. -Filtrage des requêtes et des réponses. -Journalisation très performante - Analyse du contenu lors d'utilisation de la méthodePOST	-Fonctionne seulement pour Apache et Nginix.
Opportunité	Menace
- Intégration facile dans d'autres outils. - Multiplication du nombre d'attaques d'applications	- Les faux positifs

Tab.III.3 : Matrice SWOT pour ModSecurity.

4. Application de l'Analyse SWOT pour Webknigth

Forces	**Faiblesses**
-Gratuit. -journalisation. -personnalisable.	-Comme pour tout autre filtre ISAPI , le code chargé dans le serveur compromet son intégrité. Des extensions mal conçues peuvent avoir un impact sur les performances voire dans le pire de faire planter le serveur. -Fonction des nouveaux outils plus performants
Opportunités	**Menaces**
-Multiplication du nombre d'attaque d'application	- l'application des nouveaux outils plus performants

Tab.III.4 : Matrice SWOT pour Webknigth.

5. Solution Choisie

Après avoir effectué l'analyse SWOT pour chaque solution, on voit que l'outil open source ModSecurity répond mieux au cahier de charge de notre projet.

III.3. Conclusion

Au cours de ce chapitre nous avons présenté l'organisme d'accueil, nous avons identifié le cadre général de notre projet, l'étude préalable nous a permis de comprendre la Problématique et le travail à réaliser Finalement, nous avons analysél es différentes solutions du firewall pour en extraire les outils à utiliser.

IV.Introduction

La performance et la qualité d'une application repose sur le bon choix des outils adéquats répondants aux besoins de cette dernière.

Dans la première partie nous allons donc présenter l'environnement de configuration et les outils utilisés, En deuxième lieux nous allons détailler la configuration de notre pare-feu afin de mettre en évidence les composants et les techniques de configuration utilisées, et dans la troisième partie on testera quelques attaques web et on analysera. Les attaques de chaque teste.

IV.1. Les Outils Utilisés

La réalisation de notre pare-feu a nécessité l'utilisation de plusieurs outils de configuration, que nous citerons dans ce qui suit :

IV.1.1. virtualBox

VirtualBox est un logiciel libre de virtualisation de systèmes d'exploitation publié par Oracle. Il est non seulement un produit extrêment riche en fonctionnalités et très performant pour les entreprises, mais c'est également la seule solution professionnelle disponible gratuitement en tant que logiciel open source sous les termes de la version GNU General Public License 2.

VirtualBox est activement développé avec des versions fréquentes et dispose d'une liste sans cesse croissante de fonctionnalités, de systèmes d'exploitation invités pris en charge et de plates-formes sur lesquelles il s'exécute. [28]

Fig.IV.1. virtualBox.

IV.1.2. Debian9 Strech

Debian est une suite de programmes de base et d'utilitaire libres permettant à un ordinateur de fonctionner.

Debian 9.0 C'est l'actuelle version stable, elle a été initialement publiée le 17 juin 2017. Cette version comprenait de nombreux changements majeurs tel que :

- Plus de 90% des paquets source inclus dans Debian 9 compileront des paquets binaires identiques bit par bit. Il s'agit d'une fonctionnalité de vérification importante qui protège les utilisateurs contre les tentatives malveillantes de falsification de compilateurs et de création de réseaux. [29]

- La version Stretch est la première version de Debian à inclure la branche moderne de GnuPG dans le paquet gnupg. Cela implique une meilleure cryptographie, de meilleurs paramètres par défaut, une architecture plus modulaire et une prise en charge améliorée des cartes à puce. [29]

Cette version comprend de nombreux progiciels mis à jour, tels que :

- Apache 2.4.25
- Firefox 45.9
- Tomcat 8.5
- PHP 7,0
- PostgreSQL 9.6
- Python 2.7.13 et 3.5.3, etc.
- Plus de 51 000 autres logiciels prêts à l'emploi, constitués d'un peu plus de 25 000 paquets sources.

Fig.IV.2. Debian9 Stretch.

IV.1.3. Apache

Apache est un serveur web open source basé sur le protocole http et produit par l'ASF (apache software fondation). La première version de ce serveur est sortie en avril 1995, et depuis 1996 est devenu le serveur web le plus utilisé dans le monde.

Apache est le fruit d'une multitude de correctifs et d'additions au serveur http du NCSA (National Center for Super computer Applications), l'un des tous premiers serveurs web à voir le jour. [30]

Fig.IV.3. Apache server.

Le serveur abrite plusieurs fonctionnalités comme :

- La possibilité d'héberger plusieurs sites dans un seul serveur web.
- L'utilisation des langages interprétés perl, PHP et Python.
- La sauvegarde des accès dans un fichier log.
- L'utilisation du fichier « .htaccess » pour la protection des répertoires et l'URL

Pourquoi apache ?

Apache offre plusieurs avantages à savoir :

- Riche en fonctionnalités
- Personnalisable
- Simplifications administrative
- Extensible
- Efficacité
- Stabilité

IV.1.4. NetFilter

Netfilter est donc un Framework permettant d'incorporer un pare-feu au sein du noyau Linux à partir de la version 2.4. Il fonctionne en mettant en place des accroches ou crochet (hooks) dans le noyau Linux permettant d'intercepter les paquets réseaux qui y transitent (en émission ou en

réception) pour ensuite indiquer s'ils peuvent passer ou non, ou si d'autres actions doivent y être appliquées.

NetFilter est le pare-feu linux, IPtables et NFtables ne sont que des intermédiaires permettant de configurer NetFilter. [31]

IV.1.5. Nftables

Nftables est un outil d'interface utilisateur pour le pare-feu Linux Netfilter permettant la gestion, la classification et le filtrage de paquets ainsi que le NAT.

La version de nftables dans Debian stable Stretch est la v0.7 et le noyau est la **v4.9**. Ceci est clairement une version très récente des deux composants. Dans le cas de nftables, est la dernière version publiée à ce moment-là de cette écriture.

Globalement, NFtables a été conçu dans l'optique de résoudre les problèmes et les faiblesses d'IPtables. [31]

IV.1.5.1. Mode de fonctionnement

Nftables repose sur des règles qui spécifient les actions. Ces règles sont attachées aux chaînes. Une chaîne peut contenir une collection de règles et est enregistrée dans les hooks netfilter. Les chaînes sont stockées dans des tables. Une table est spécifique à l'un des protocoles de couche 3. Tel que :

1- Table : est un conteneur de chaine, il existe 5 familles de table :
   - Les tables de la famille "IP"
   - Les tables de la famille "Inet" (combine ipv4 et ipv6)
   - Les tables de la famille "ARP"
   - Les tables de la famille "IP6"
   - Les tables de la famille "Bridge"

2- Chaine : Une chaine va contenir et stocker des règles, il existe 5 chaines :
   - Prerouting
   - Input
   - Output
   - Forward
   - Postouting

3- Règle : Une règle n'est qu'au final une action à appliquer.

Les hooks sont les éléments sur lesquels s'appliquent les règles et qui permettent donc de changer le comportement des paquets en entrée ou sortie de l'interface en fonction des règles configurées. [31]

Lorsqu'un paquet arrive ou sort du système, on peut penser qu'il n'effectue que le chemin suivant entre l'interface réseau (en haut) et les applications (en bas)

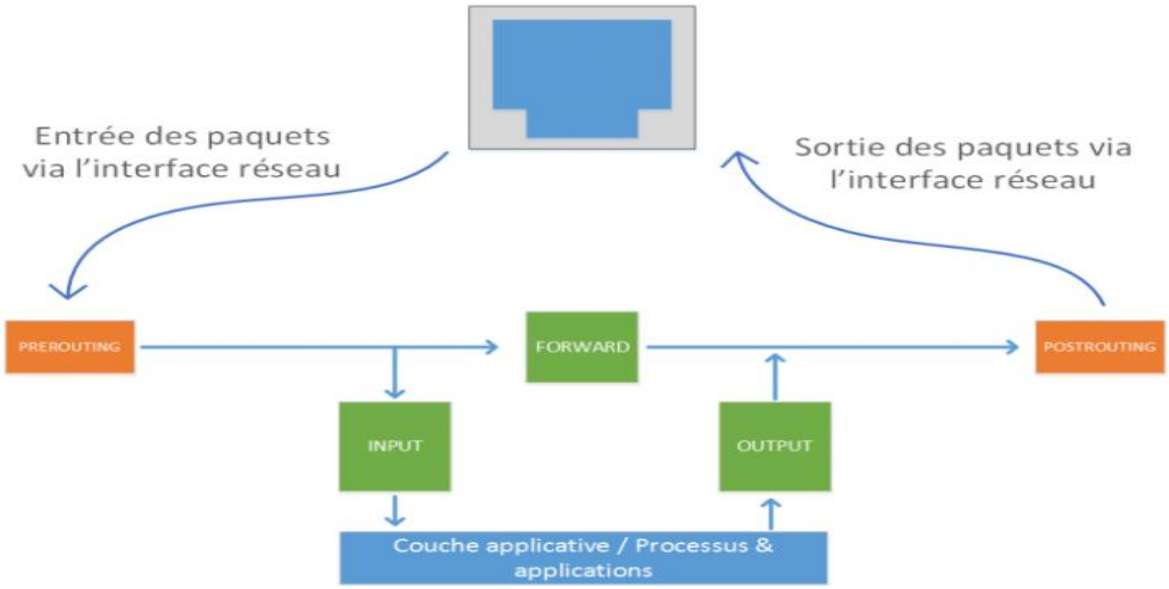


Fig.IV.4. Flux réseau avec les contrôles de NetFilter.

Comme nous pouvons le voir sur le schéma ci-dessus, on peut trouver différents *Hooks* (ou *crochets)* en fonction de l'avancement du paquet dans son traitement par le système.

Pour chaque crochet positionné par NetFilter, on va pouvoir :

- Modifier le paquet, puis le rendre ou non au système.
- Imposer au système de supprimer le paquet, cela correspond alors à un rejet du paquet.
- Indiquer au système que le paquet est accepté pour ce hook, jusqu'au prochain.

Chaque *Hook* est relié à une chaine que nous pouvons configurer dans Nftables :

- Hook NF_IP_PRE_ROUTING qui correspond à la chaine *"PREROUTING"* dans NFtables

Dans ce crochet, les paquets sont analysés dans leur forme brute, sans traitement préalable du système. On peut alors déterminer si on autorise le paquet à entrer plus loin dans le système ou non.

- Hook NF_IP_LOCAL_IN qui correspond à la chaine *"INPUT"* dans NFtables

Ici, les paquets sont prêt à être envoyés à la couche applicative, c'est-à-dire aux applications qui les traiterons (exemple : un serveur web ou un service FTP).

- Hook NF_IP_FOWARD qui correspond à la chaine *"FORWARD"* dans NFtables

Lorsque ce Hook est utilisé, c'est que les paquets n'iront pas vers la couche applicative, mais seront redirigés vers une autre interface réseau. Par exemple dans le cas où la machine est un routeur.

- Hook NF_IP_LOCAL_OUT qui correspond à la chaine *"OUTPUT"* dans NFtables

Il s'agit ici du même fonctionnement que la chaine "OUTPUT", mais en sortie de la couche applicative. On va donc autoriser ou non un paquet à sortir vers l'interface réseau

- Hook NF_IP_POSTROUTING qui correspond à la chaine *"POSTROUTING"* dans NFtables

On retrouve ici le même principe que la chaine "PREROUTING", mais pour la sortie des paquets. Les paquets analysés sont ici de nouveau dans leur forme brute. [31]

IV.1.6. Modsecurity

ModSecurity est un projet issu du monde open source qui a débuté en 2002. Aujourd'hui disponible en version 2.6.x, il a acquis des performances honorables, tant en termes de Stabilité et de traitements. Ce projet est accessible gratuitement et il est soutenu par BreachSecurity qui développe des solutions dédiées aux entreprises clef en main avec ses machines dédiées à la tâche. La philosophie de ModSecurity est aussi très transparente. Ainsi rien n'est effectué implicitement puisque tout est accessible dans la configuration. Bref les utilisateurs ne contrôlent point par point le système et sans surprises. [32]

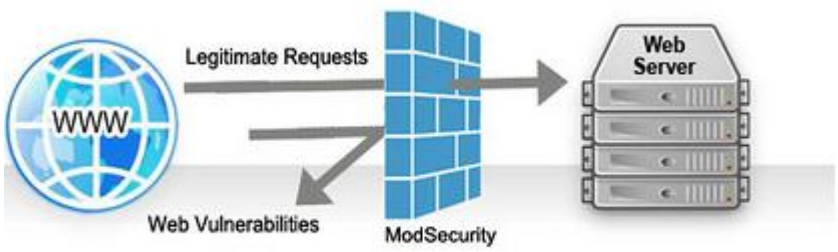


Fig.IV.5. Modsecurity.

IV.1.6.1. Fonctionnalités de ModSecurity

• **Meilleure journalisation :** ModSecurity permet de journaliser le contenu des requêtes POST HTTP voir des transactions HTTP complètes, ce qui a pour effet de rendre le travail d'un attaquent potentiel plus difficile.

•**Opération en temps réel :** le module a la possibilité de voir les requêtes directement et donc peut opérer tout de suite.

•**Anomalies :** En cas d'opération anormales, le WAF peut examiner les taux de requêtes, les adresses IP, les sessions HTTP et les comptes utilisateur et agir en fonction des résultats.

•**Liste noire et blanche :** L'utilisateur peut employer une approche basée sur une signature et préciser ce qu'il souhaite autoriser et ce qu'il souhaite interdire.

• **Protection d'autres serveurs web :** ModSecurity rend possible la protection des applications web hébergées sur des serveurs web autres qu'Apache en utilisant ce dernier comme serveur mandataire inverse. Le module mod_proxy permet cette utilisation à Apache.

IV.1.6.2. Fonctionnement Global de ModSecurity

ModSecurity dispose de cinq phases de traitement qui correspondent chacune à une étape clé, Pour chaque transaction, les phases sont exécutées séquentiellement de la phase 1 à la phase5, Une transaction correspond à une requête d'un client et la réponse renvoyée par le serveur.

Phase 1 : Traitement des en-têtes de la requête

Cette phase permet notamment d'inspecter les arguments passés dans l'URL, dans le cas d'une requête en GET ainsi que de vérifier les cookies ou le UserAgent.

Phase 2 : Traitement du corps de la requête

L'analyse du corps de la requête permet d'inspecter les arguments dans le cas d'une requête en POST.

Phase 3 : Traitement des en-têtes de la réponse

Les en-têtes de la réponse du serveur Web, sont observés par ModSecurity afin de décider s'il est nécessaire ou non d'inspecter le corps de la réponse.

Phase 4 : Traitement du corps de la réponse

Cette phase est identique à la phase 2 sauf qu'elle traite la réponse fournie par le serveur. Elle permet notamment de prévenir la fuite d'informations via des messages d'erreur.

Phase 5 : Journalisation

Cette phase permet la journalisation des requêtes, elle permet de définir comment la transaction doit être journalisée. Cette phase intervient en dernier, c.à.d. Lorsque le traitement de la transaction est déjà effectué. On ne peut donc pas bloquer une transaction lors de cette phase.

Les phases 1 et 2 permettent d'interrompre une requête avant qu'elle n'arrive au serveur. Les phases 3 et 4 permettent d'interrompre une réponse avant qu'elle n'arrive au client.

Le schéma ci-dessous illustre les cinq phases de traitement qu'on vient de décrire :

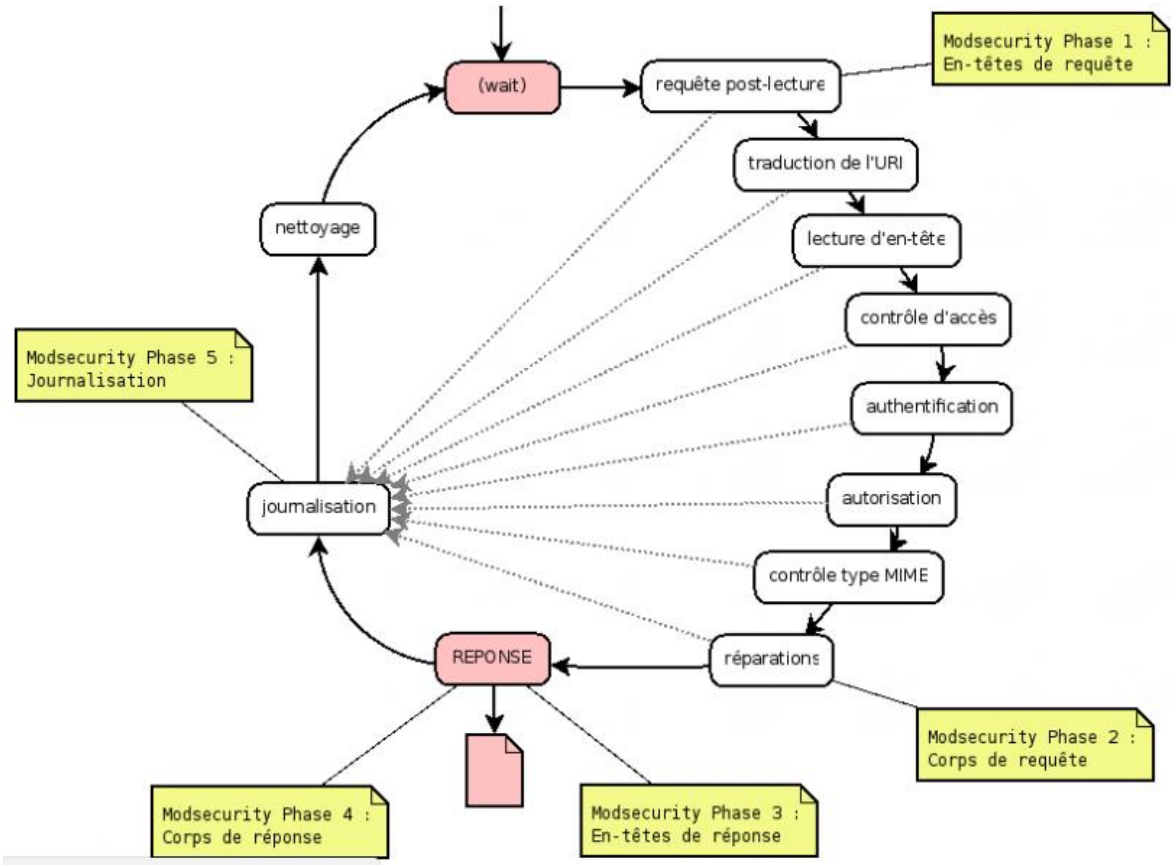


Fig.IV.6. Les cinq phases d'intervention de ModSecurity.

IV.1.6.3. Déploiement de ModSecurity

ModSecurity peut être déployé dans différents modes, comme décrit dans la figure ci-dessous :

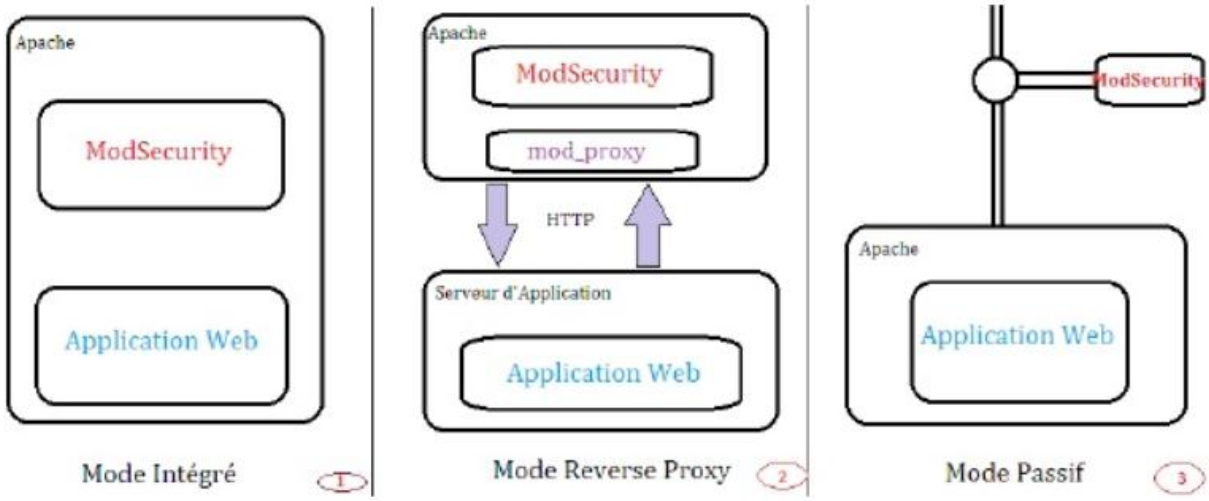


Fig.IV.7. Différents modes de déploiement de ModSecurity.

Cette figure montre que ModSecurity peut être utilisé comme :

1- Partie de serveur web, et dans ce cas il assure la sécurisation du serveur dans lequel il est intégré seulement,
2- Un reverse-proxy dans ce cas il assure la sécurisation d'un ensemble de serveurs
3- En mode passif, dans ce cas il écoute le trafic sur un port de control (n'est pas en ligne directe avec le trafic) et donc n'est pas capable de l'influencer.

IV.1.6.4. Éléments de Bases de ModSecurity

ModSecurity est constitué de deux éléments indispensables pour son fonctionnement :

- **La Configuration :** Indique à ModSeurity comment traiter les données qu'il reçoit.
- **Les Règles :** Décident que faire avec les données traitées par ModSecurity.

IV.1.7. Reverse-proxy

IV.1.7.1. Présentation

Un reverse proxy effectue le travail opposé à celui d'un proxy "standard". En effet, un proxy classique se place entre un client et les serveurs auxquels il peut accéder. Il transfère les requêtes HTTP du client aux serveurs externes et lui renvoie les réponses correspondantes.

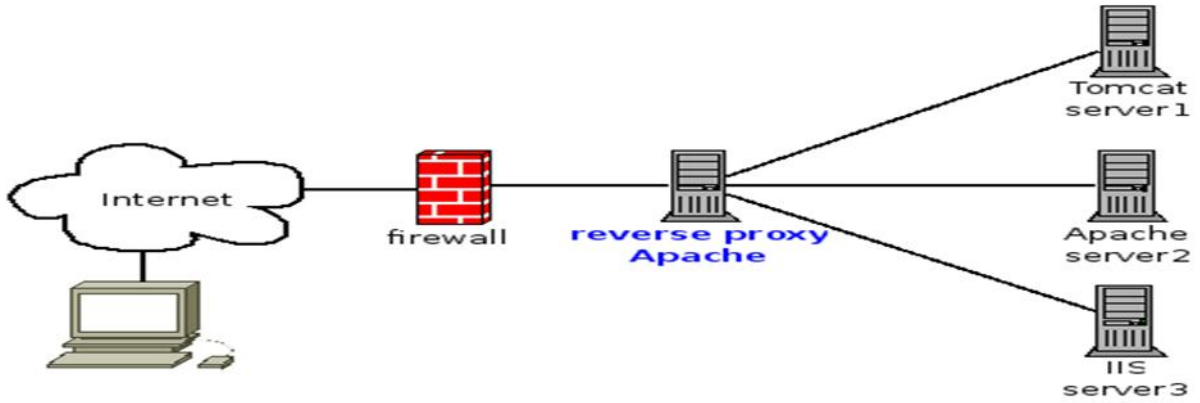


Fig.IV.8. Reverse proxy.

Un proxy inversé se place entre un serveur et tous ses clients. Plus généralement, on va utiliser ce type de proxy afin d'obtenir un seul point d'entrée vers un ou plusieurs serveurs de manière transparente pour l'utilisateur. Le reverse proxy récupère les requêtes HTTP des clients et se charge de les transmettre aux serveurs internes désignés par les requêtes correspondantes.

Il existe de nombreuses solutions logicielles qui implémentent ce mécanisme (Apache, Squid, Pound, Nginx, etc.)

IV.1.7.2. Principales fonctionnalités du RP

- Le Reverse Proxy peut porter le **chiffrement SSL** et être directement connecté à des serveurs d'authentification avec lesquels il va gérer les droits d'accès et la durée des sessions.
- Ensuite avec la **mémoire cache**, le proxy inverse peut décharger les serveurs Web de la charge de pages statiques. Et comme pour le proxy, le RP peut réaliser la **compression du contenu** des sites pour optimiser la bande passante et le chargement des contenus.
- Enfin le RP permet de faire de la **répartition de charge** en redirigeant les requêtes vers les différents serveurs.

IV.1.7.3. Fonctionnement du reverse proxy dans notre cas

La mise en place d'un reverse proxy avec Apache a besoin du module proxy qui est disponible nativement. Cependant, dans notre cadre qui nécessite une certaine sécurité nous devons utiliser des modules supplémentaires comme le module ssl (natif) ainsi que le module security. On peut éventuellement ajouter le module evasive afin de pouvoir limiter dans une certaine mesure les attaques de type DDOS (Distributed Denial of Service), comme il est illustré dans la figure suivante :

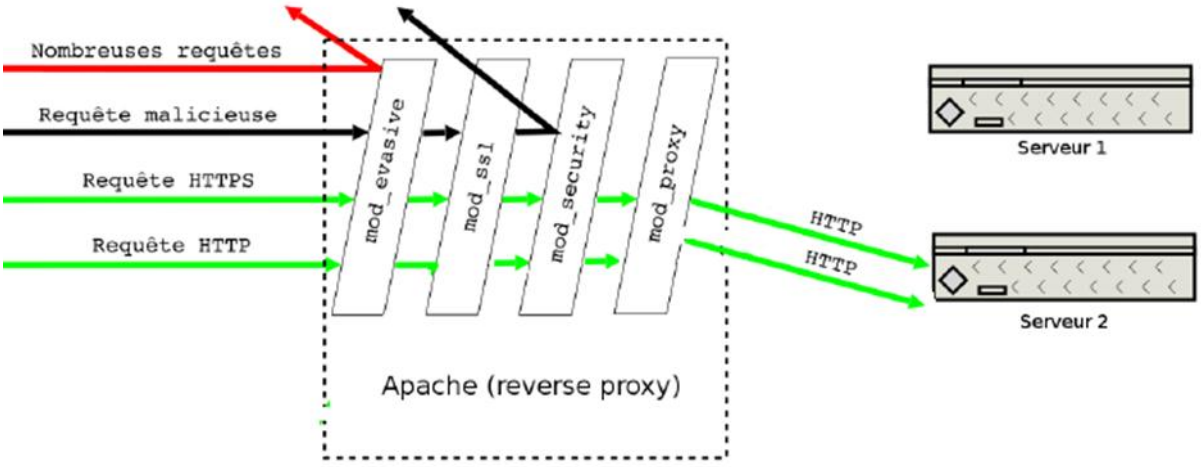


Fig.IV.9. Fonctionnement du reverse proxy

IV.1.7.3.1. mod_evasive

Ce module de sécurité pour Apache a pour objectif de détecter les accès massifs à certaines ressources qui sont révélateurs d'attaques de type DDOS et de les stopper (dans une certaine mesure) en bloquant temporairement les IP néfastes. Il peut être intéressant d'installer ce type d'outil sur un reverse proxy afin de limiter les risques de défaillances en cas de DDOS de faible ou moyenne envergure. [33]

IV.1.7.3.2. mod_ssl

Il s'agit du module dédié aux échanges chiffrés grâce aux protocoles SSL/TLS.

IV.7.3.3. mod_security

Ce module est un pare-feu applicatif permettant de filtrer les flux HTTPS de manière très précise parce qu'il a accès aux données de la transaction avant le chiffrement et après le déchiffrement. Il agit un peu à la manière d'un IDS mais en analysant le traffic réseau au niveau HTTP. mod_security fonctionne grâce à une ensemble de règles assez complexes permettant de bloquer des schémas d'attaques HTTP et de filtrer certaines requêtes suspicieuses. Ce module va plus loin en filtrant également les réponses HTTP ainsi que le contenu des requêtes de type POST. Lorsqu'il est correctement configuré, il permet d'empêcher de manière très efficace un nombre impressionnant d'attaques (injections sql, tentatives d'attaques type xss etc). [33]

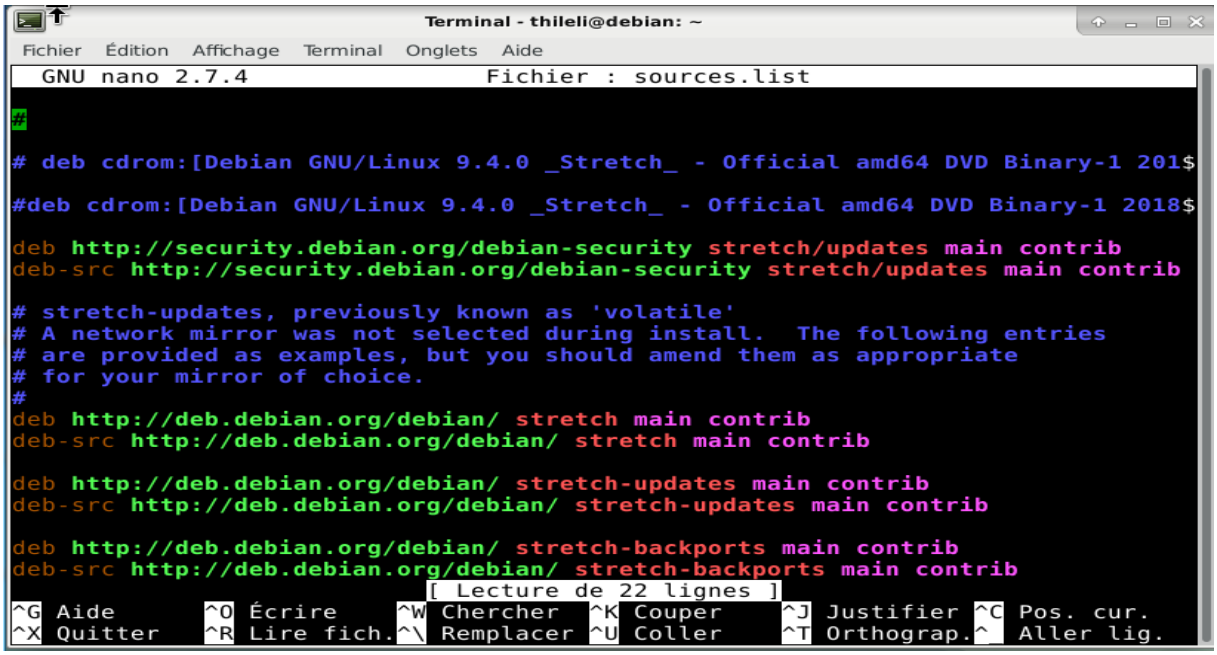
IV.1.7.3.4. mod_proxy

Son rôle est de récupérer les flux SSL/TLS qu'il reçoit et de les renvoyer sous forme de requêtes HTTP simples vers le/les serveurs protégés derrière le proxy. [33]

IV.2. Configuration

IV.2.1. Debian

On s'est rendu dans le fichier ***/etc/apt/sources.list*** et on a ajouté les lignes suivantes :



Fichier source.list.

Le fichier ***sources. list*** suivant propose les mêmes paquets que le ***sources. list*** de base, et inclut également les paquets ***non libres*** ou dépendants de ressources non libres proposés dans les sections ***contrib*** et ***non-free***.Le dépôt ***backports*** propose des paquets plus récents ou absents du dépôt principal.

L'installation de logiciels (plus généralement appelé ***paquets***) se fait directement par le gestionnaire de paquets APT. Contrairement à Windows. Le gestionnaire de paquets s'occupe de les télécharger et de les installer lui-même.

Une fois les modifications effectuées, il est nécessaire d'actualiser les listes de paquets :

apt-get update.

apt-get upgrade.

IV.2.2. Apache

L'installation d'apache se fait via la commande :

```
root@debian:~# service apache2 reload
```

Lancement du serveur apache

```
root@debian:~# systemctl start apache2
```

IV.2.3 Nftables

IV.2.3.1 Installation de nftables

L'installation de nftables se fait via la commande suivante :

```
root@debian:~# apt-get install nftables
```

- **Installation des librairies**

- **Libmnl** : cette librairie permet de fournir une interface à NFtables afin de communiquer avec le noyau du système

- **Libnftnl** : Il s'agit de la librairie propre à NFtables, elle fournit des API bas niveau pour transformer les messages netlink en objets

```
root@debian:~# apt-get install libnftnl-dev
root@debian:~# apt-get install libmnl-dev
```

- **Activation des services de nftables**

```
root@debian:~# systemctl status nftables
● nftables.service - nftables
   Loaded: loaded (/lib/systemd/system/nftables.service; disabled; vendor preset
   Active: inactive (dead)
```

```
root@debian:~# systemctl status nftables.service
● nftables.service - nftables
   Loaded: loaded (/lib/systemd/system/nftables.service; enabled; vendor preset:
```

IV.2.3.2 Configuration de Nftables

Nous disposons d'un serveur web (HTTP et HTTPS) sur lequel nous intervenons de temps à autre via SSH, Cela nous permet de nous fixer quelques règles :

- Connexion déjà établis : autorisé.
- Connexion invalide : refusé.
- Ports 80 et 443 ouverts en entrée (INPUT).
- Port 22 ouvert en entrée (INPUT).
- ICMP type réponse : Autorisé.
- Refus du reste du trafic.
- En sortie tout est autorisé.

Création de la table ***filter*** type***Inet*** avec la commande suivante :

```
root@debian:~# nft add table inet filter
```

Création des chaines ***Input, Output*** avec les commandes suivantes :

```
root@debian:~# nft add chain inet filter input { type filter hook input priority 0 \;}
root@debian:~# nft add chain inet filter output { type filter hook output priority 0 \;}
```

```
root@debian:~# nft list ruleset
table inet filter {
        chain input {
                type filter hook input priority 0; policy accept;
        }

        chain output {
                type filter hook output priority 0; policy accept;
        }
}
```

Maintenant on va passer à la configuration de nos règles voici quelques exemples :

```
root@debian:~# nft add rule filter input ct state established,related accept
root@debian:~# nft add rule filter input ct state invalid drop
root@debian:~# nft add rule filter input tcp dport 80 accept
```

Et on a obtenu la table suivante :

```
root@debian:~# nft list ruleset
table inet filter {
        chain input {
                type filter hook input priority 0; policy accept;
                iif "lo" accept
                ct state established,related accept
                ct state invalid drop
                tcp dport { ssh, https, http} ct state new accept
                counter packets 9 bytes 5184 drop
        }

        chain output {
                type filter hook output priority 0; policy drop;
        }
}
```

Pour sauvegarder les règles NFtables, Nous allons utiliser une syntaxe courante sous Linux qui consiste à rediriger la table ***filter*** qui s'affiche sur le terminal dans le fichier ***nftables.conf***.

```
root@debian:~# nft list table filter > /etc/nftables.conf
```

On tape ***: nano /etc/nftables.conf*** et on aura le résultat suivant :

```
 GNU nano 2.7.4                         Fichier : /etc/nftables.conf

#!/usr/sbin/nft -f

flush ruleset

table inet filter {
        chain input {
                type filter hook input priority 0;
                # accept any localhost traffic
                iif lo accept
                # accept traffic originated from us
                ct state established,related accept
                ct state invalid drop
                # activate the following line to accept common local services
                tcp dport { 22, 80, 443 } ct state new accept
                 ip protocol icmp icmp type  {echo-reply} accept
                # count and drop any other traffic
                counter drop}

chain output {
                type filter hook output priority 0;policy accept;
}

}
```

IV.2.4. ModSecurity

IV.2.4.1 Installation de ModSecurity

L'installation de modsecurity se fait via la commande

```
root@debian:~# apt-get install libapache2-modsecurity
```

Vérification le module modsecurity a été chargé.

```
root@debian:~# apachectl -M | grep --color security
 security2 module (shared)
```

L'installation de Modsecurity inclut un fichier de configuration recommandé qui doit être renommé en ***modsecurity.conf***:

```
root@debian:~# mv /etc/modsecurity.conf{-recommended,}
```

Recharger Apache

```
root@debian:~# service apache2 reload
```

On trouvera un nouveau fichier journal pour mod_security dans le répertoire des journaux Apache :

```
root@debian:~# ls -l /var/log/apache2/modsec_audit.log
-rw-r----- 1 root adm 41277 sept. 22 17:38 /var/log/apache2/modsec_audit.log
```

IV.2.4.2 Configuration de ModSecurity

ModSecurity admet une série d'options qui permettent de renforcer la sécurité des applications web. Ici, on décrit quelques directives de configuration de base.

SecRuleEngine On

SecRequestBodyAccess On

SecResponseBodyAccess On

Ces trois lignes activent le moteur de règles ModSecurity et l'accès au corps de la requête et de la réponse. La première directive a trois valeurs possibles, **On/Off** pour **activer/désactiver** l'exécution des règles, **DetectionOnly** pour traiter les règles mais rien ne sera bloqué. Les deux dernières directives décident si l'examen du corps des requêtes et des réponses sera effectué.

Cet examen nécessite que les données soient bufférisées et donc de la mémoire. Le contrôle de la quantité de cette dernière est valable via la directive :

SecRequestBodyInMemoryLimit 131072

La taille des données du corps de la requête peut également être limitée comme suit :

SecRequestBodyLimit 10485760

Si le contrôle des types de données est requis, il faut vérifier le type MIME (Multipurpose Internet Mail Extensions) des données comme il est le cas dans l'exemple ci-dessous :

SecResponseBodyMimeTypesClear

SecResponseBodyMimeType (null) text/plain text/html text/css text/xml

La première directive affiche la liste des types à vérifier. La seconde ligne définit les types qui nous intéressent. Dans cette exemple nous ne voudrons pas analyser des données de type JPEG ou PDF.

Contrôler l'uploade de fichiers peut être une chose très souhaitée. Chaque fichier uploadé peut être transféré dans un répertoire distinct. En outre, tous les fichiers uploadés du votre serveur peuvent être rassemblés à condition d'avoir de l'espace disque. Ceci peut être utile lors d'activités légales après que quelque chose soit mal passé.

SecUploadDir /var/spool/apache/private

SecUploadKeepFiles Off

Journalisation

Afin de pouvoir communiquer à son administrateur les derniers résultats des attaques sur un serveur Web bien donné, Modsecurity détaille dans des fichiers de log ses activités. Et laisse à son administrateur le choix de tout journaliser ou seulement les évènements qui ont déclenché une des vérifications internes.

- **Exemple** :

SecAuditEngine RelevantOnly

SecAuditLogRelevantStatus "^[45]"

Dans cet exemple ModSecurity va enregistrer seulement les réponses qui génèrent des codes de statut 4xx ou 5xx.

La journalisation peut être notée dans un simple fichier ou dans un répertoire avec un fichier pour chaque évènement comme pour l'exemple ci-dessous. Le dernier choix est habituellement réservé aux sites à gros volume.

SecAuditLogType Serial

SecAuditLog /var/log/www/modsec_audit.log.

- **Exemple de Configuration**

Voici un exemple du fichier de configuration de ModSecurity :

#Les options de configuration de base

SecRuleEngine On

SecRequestBodyAccess On

SecResponseBodyAccess Off

SecDataDir /var/log/modsec_data

#Manipulation de téléchargement de fichiers

SecUploadDir /var/tmp/modsec_upload

SecUploadFileMode 0600

SecUploadKeepFiles RelevantOnly

#Journal de débogage

SecDebugLog /var/log/apache2/modsec_debug.log

SecDebugLogLevel 0

Journal d'audit d'en série

SecAuditEngine RelevantOnly

SecAuditLogRelevantStatus ^5

SecAuditLogParts ABIFHKZ

SecAuditLogType Serial

SecAuditLog /var/log/apache2/modsec_audit.log

La taille du corps de la demande maximale, nous allons accepter pour tampon

SecRequestBodyLimit 131072

Stockez jusqu'à 128 Ko dans la mémoire

SecRequestBodyInMemoryLimit 131072

Organismes de jusqu'à 512 KB longueur de réponse de la mémoire tampon.

SecResponseBodyLimit 524288

Inclure les règles que nous utilisons pour les tests

Include /etc/apache2/modsecurity-test/test.conf

Détails de certains paramètres pour clarifier les options de configuration :

SecServerSignature

Permet de donner un nom personnalisé au serveur Web.

SecRuleEngine

Permet d'activer ou de désactiver ModSecurity

SecRequestBodyAccess

Permet de spécifier à ModSecurity d'inspecter ou non le corps des requêtes

SecResponseBodyAccess

Permet de spécifier à ModSecurity d'inspecter ou non le corps des réponses

SecDataDir

Spécifie le répertoire permettant à ModSecurity de stocker des informations.

#Ce répertoire est nécessaire notamment pour pouvoir utiliser les variables

#persistantes dans les règles. Il doit être inscriptible avec les droits

www-data pour pouvoir fonctionner.

SecDebugLogLevel

Cette option fixe la verbosité des logs de ModSecurity.

IV.2.4.3 Les règles de filtrage

Une règle ModSecurity s'écrit de la façon suivante :

SecRule VARIABLES OPERATEURS [FONCTIONS_DE_TRANSFORMATION,

ACTIONS]

- **Exemple Explicatif**

Supposons que nous somme dans le cas d'une page d'authentification, contenant deux champs login et mot de passe.

Dans cet exemple, nous allons chercher à bloquer l'attaque XSS et l'injection SQL.

```
# SQL Injection
SecRule REQUEST_FILENAME « ^/test_modsec/monfichier$ » « msg:'Virtual Patch SQL
violation',
tag:'WEB_ATTACK/ATTACK_SQL',chain,block,log,capture,multiMatch,t:none,t:cssDecode,
t:jsDecode,t:htmlEntityDecode,t:replaceComments,t:compressWhiteSpace »

SecRule ARGS_POST:login|ARGS_POST:password »(?i:'|\ »|\de\d\d|%[a-fA-F]\d%[a-fA-
F]\d|% \d% \d|\)|\bAND[\s]+|\bMOD[\s]+|\+[\s]*\d|%2b[\s]*\d|\-
[\s]*\d|\bOR[\s]+|\bIN[\s]+[\(]|\bTABLE[\s]+|\bCREATE[\s]+|\bDELETE[\s]+[\(]|\bUNION[\s]+[\(
]|\bLIKE[\s]+[\(]|\bJOIN[\s]+[\(]|\bSELECT[\s]+|\bLIMIT[\s]+|\bFROM[\s]+|%|_|\-\-|#|\/\*) »
```

Bloquer l'attaque par injection SQL.

Pour écrire cette règle, nous utilisons l'option chaine. La première règle va s'assurer que nous sommes bien sur notre page de login et formater les données pour les rendre uniformes. Cette étape nous met à l'abri des techniques de contournement via l'utilisation d'encodage, de commentaires ou autres. Une fois cette phase est effectuée, nous appelons la seconde règle qui va vérifier, via une expression régulière la présence de mots clés et de caractères étant des instructions SQL.

L'action à effectuer si nous trouvons une correspondance avec notre expression régulière est de bloquer la requête.

```
#XSS
SecRule REQUEST_FILENAME « ^/test_modsec/monfichier$ » « msg:'Virtual Patch Cross-
Site Scripting violation',
tag:'WEB_ATTACK/CROSS-
SITE_SCRIPTING',chain,block,log,capture,multiMatch,t:none,t:cssDecode,
t:jsDecode,t:htmlEntityDecode,t:replaceComments,t:compressWhiteSpace »
SecRule ARGS_POST:login
« (?i:'|\ »|<|>|\;|\bonchange[\s]*[=]|\bon(dblclick|click)[\s]*[=]|\bon(blur|focus)[\s]*[=]|\bon(submi
t|reset)[\s]*[=]|\bonmouse(over|out|down|up)[\s]*[=]|\bonkey(down|press|up|up)[\s]*[=]|\bon(lo
ad|unload|error|abort|move|resize|dragdrop)[\s]*[=]|\b(java|j|vb|live)script[:]|//|\/\*|\*\/) »
```

Bloquer l'attaque XSS.

Nous utilisons la même technique pour filtrer les XSS sauf qu'on ne l'applique qu'au champ login car le champ password n'est pas affiché. La dernière étape est d'indiquer à ModSecurity de bloquer la requête si une des règles a remonté une alerte, pour cela nous devons ajouter avant ces règles la directive suivante :

SecDefaultAction phase :2, log, deny, status :403

- **Projet OWASP: ModSecurity Core Rule Set (CRS)**

D'après la section précédente, la configuration de ModSecurity avec des règles de blocage est indispensable pour son bon fonctionnement. Ces règles peuvent être écrites par les administrateurs de ModSecurity, sinon, ces derniers peuvent utiliser les règles OWASP.OWASP est un ensemble de communautés de sécurité qui développement et maintient un ensemble de règles gratuites de protection des applications web. Cet ensemble de règles est appelé l'OWASP ModSecurity Core Rule Set (CRS). C'est un ensemble renforcé de règles de base que ModSecurity utilisera pour traiter les attaques sur le serveur.

Ce kit de base de ModSecurity, ajoute à ce dernier des règles par défaut qui se basent sur une approche de type liste noire. Ces règles offrent une protection considérable contre les attaques les plus connues, permettent de modifier les messages d'erreurs renvoyés par le serveur, et sont organisées dans plusieurs fichiers en fonctions du type d'attaque ou de protection.

Pour ajouter les règles OWASP à notre configuration on insert la commande suivante :

nano /etc/apache2/mods-enables/security2.conf

Et on ajoute ces lignes :

```
<IfModule security2_module>
        # Default Debian dir for modsecurity's persistent data
        SecDataDir /var/cache/modsecurity

        # Include all the *.conf files in /etc/modsecurity.
        # Keeping your local configuration in that directory
        # will allow for an easy upgrade of THIS file and
        # make your life easier
        IncludeOptional /etc/modsecurity/*.conf

        # Include OWASP ModSecurity CRS rules if installed
        IncludeOptional /usr/share/modsecurity-crs/owasp-crs.load
</IfModule>
```

Fichier security2. conf.

Et on tape la commande suivante pour vérifier si les règles ont été bien chargées :

nano /usr/share/modsecurity-crs/rules/ls

On obtiendra les fichiers de règles par défaut de la dernière version de ModSecurity :

```
REQUEST-901-INITIALIZATION.conf
REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf
REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.conf
REQUEST-905-COMMON-EXCEPTIONS.conf
REQUEST-910-IP-REPUTATION.conf
REQUEST-911-METHOD-ENFORCEMENT.conf
REQUEST-912-DOS-PROTECTION.conf
REQUEST-913-SCANNER-DETECTION.conf
REQUEST-920-PROTOCOL-ENFORCEMENT.conf
REQUEST-921-PROTOCOL-ATTACK.conf
REQUEST-930-APPLICATION-ATTACK-LFI.conf
REQUEST-931-APPLICATION-ATTACK-RFI.conf
REQUEST-932-APPLICATION-ATTACK-RCE.conf
REQUEST-933-APPLICATION-ATTACK-PHP.conf
REQUEST-941-APPLICATION-ATTACK-XSS.conf
REQUEST-942-APPLICATION-ATTACK-SQLI.conf
REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf
REQUEST-949-BLOCKING-EVALUATION.conf
RESPONSE-950-DATA-LEAKAGES.conf
RESPONSE-951-DATA-LEAKAGES-SQL.conf
RESPONSE-952-DATA-LEAKAGES-JAVA.conf
RESPONSE-953-DATA-LEAKAGES-PHP.conf
RESPONSE-954-DATA-LEAKAGES-IIS.conf
RESPONSE-959-BLOCKING-EVALUATION.conf
RESPONSE-980-CORRELATION.conf
restricted-files.data
scanners-headers.data
scanners-urls.data
```

Fichiers des règles OWASP.

IV.2.6. Mise en œuvre d'un Reverse Proxy Apache

La version installée contient déjà les différents modules Apache nécessaires à la mise en place d'un Reverse Proxy.

IV.2.6.1 Activation du module Reverse Proxy.

#a2enmod proxy

```
root@debian:~# a2enmod proxy
Module proxy already enabled
root@debian:~#
```

Nous devons ensuite activer les modules correspondant aux protocoles que nous souhaitons faire transiter par l'intermédiaire du Reverse Proxy. En l'occurrence, il s'agit de **proxy_http**.

#a2enmod proxy_http

```
root@debian:~# a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Module proxy_http already enabled
root@debian:~#
```

Après chaque modification de la configuration du serveur Apache, il faut relancer celui-ci avec la commande suivant :

systemctl reload apache2

IV.2.6.2 Configuration

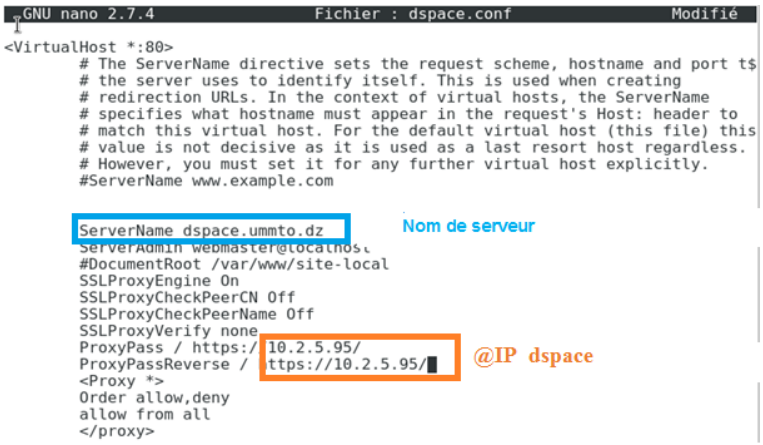
Pour que le reverse-proxy accepte les requêtes d'Internet et les transmettent aux serveurs appropriés sur le réseau local, on a besoin de désactiver le fichier de configuration **«** *000-default.conf* »dans le répertoire « */etc/apache2/sites-avaialable/*» et de le remplacer par un fichier hôte virtuel comme *dspace.conf* et qui contient la configuration du fichier par default avec les deux commandes suivants :

```
root@debian:/etc/apache2/sites-available# ls
000-default.conf  dspace.conf  fshs.conf
default-ssl.conf  fs.conf      site-local.conf
```

Il est conseillé de créer un fichier hôte virtuel séparé pour chaque serveur cible avec sa propre adresse IP.

Editer le fichier Virtual Host avec la commande suivante :

```
root@debian:/etc/apache2/sites-available# nano dspace.conf
```



Les instructions pour la fonction proxy sont définies dans la commande **<*VirtualHost*>.** La balise *Start* contient également l'adresse IP et le numéro de port ou Apache est configuré comme un serveur reverse-proxy doit recevoir les requêtes. Si toutes les adresses IP sont incluses, le métacaractère ***** est utilisé comme le montre l'exemple. Les informations de la balise ***VirtualHost*** sont donc affichées sous la forme de commande. Contrairement À la balise ***VirtualHost***, ces arguments définissent comment traiter les requêtes entrantes et les paquets de réponse. Les commandes ***ServerName, ProxyPass*** et ***ReversePass*** sont particulièrement importantes**.**

- **ServerName :** la commande ***ServerName*** définit le nom premier d'un serveur sur internet. Il doit être réglable soit via DNS ou via ***/etc/hosts***. Dans cet exemple, le serveur Apache est programmé pour accepter toutes les requêtes de ***dspace***.

- **ProxyPass :** est la principale directive de configuration du proxy. Dans ce cas, il spécifie que tout ce qui se trouve sous l'URL racine (/) doit être mappé sur le serveur principal à l'adresse indiquée. Par exemple, si Apache obtient une demande pour /despace, il se connectera à ***http://your_backend_server /despace*** et renverra la réponse au client d'origine.

- **ProxyPassReverse** : doit avoir la même configuration que ProxyPass. Il indique à Apache de modifier les en-têtes de réponse du serveur principal. Cela garantit que si le serveur d'arrière-plan renvoie un en-tête de redirection d'emplacement, le navigateur du client sera redirigé vers l'adresse proxy et non l'adresse du serveur principal, ce qui ne fonctionnera pas comme prévu.

a. Mod_evasive :

- Pour configurer mod_evasive, il faudra modifier le fichier /etc/apache2/apache2.conf pour ajouter ces quelques lignes :

  <IfModule mod_evasive20.c>

  DOSHashTableSize 3097

  DOSBlockingPeriod 20

  DOSSiteInterval 20

  DOSSiteCount 40

  DOSPageInterval 1

  DOSPageCount 3

  </IfModule>

  Il s'agit de bloquer un attaquant en lui renvoyant une erreur 403 pendant une période (DOSBlockingPeriod) s'il charge plus d'un certain nombre de page (DOSSiteCount) pendant un certain intervalle (DOSSiteInterval).

  Dans ce cas, on bloque un utilisateur s'il charge plus de 40 pages en 20 secondes

b.Mod_SSL

On utilise les directives suivantes pour configurer mod_ssl :

SSLProxyEngine : activation du mode Proxy en SSL

SSLProxyCheckPeerCN, SSLProxyCheckPeerName : permet de ne pas vérifier le CN et le nom du serveur distant.

SSLProxyVerify : pas de vérification du serveur distant

De plus, il existe aussi deux commandes supplémentaires : *ServerAlias* et *ProxyRequests*. Ces commandes ne représentent pas des fonctions de base pour le serveur proxy et sont donc plutôt facultatives.

Si les règles de la fonction proxy sont bien définies, on désactive la configuration par défaut et on active la nouvelle configuration via le terminal :

#a2dissite 000-default. conf

#a2ensite despace. conf

IV.2.6.3 Configuration du fichier hosts

Le fichier hosts est utilisé sous tous les systèmes d'exploitation lors de l'accès à Internet, ce fichier est consulté avant l'accès au serveur DNS. C'est un simple fichier qui contient sur la même ligne une adresse IP et parfois le nom de domaine.

Ce fichier permet de bloquer l'accès à un site Internet, de le rediriger, de bloquer les pubs et d'améliorer l'accès à des sites en fonction de la configuration de celui-ci. Il peut être utile pour protéger votre ordinateur, empêcher un enfant (par exemple) d'accéder à un site particulier. Ce fichier peut aussi servir à définir les noms des machines sur un réseau local (ou pas).

On édite le fichier hosts pour ajouter l'adresse IP du site-local avec la commande suivante

#nano /etc/hosts/

```
  GNU nano 2.7.4                Fichier : /etc/hosts

127.0.0.1       localhost
127.0.1.1       debian
10.0.2.15      site-local
#10.0.2.15      site-local.example.org
10.0.2.15       dspace.ummto.dz
10.0.2.15       fshs.ummto.dz
#10.2.5.95      dspace.ummto.dz
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

IV.3. Simulation d'attaques en présence et en absence du WAF

Dans cette section nous allons tester, analyser et comprendre comment ModSecurity bloque les Attaques Web Connues en fonction des modalités de déploiement :

IV.3.1. Attaques Injection SQL

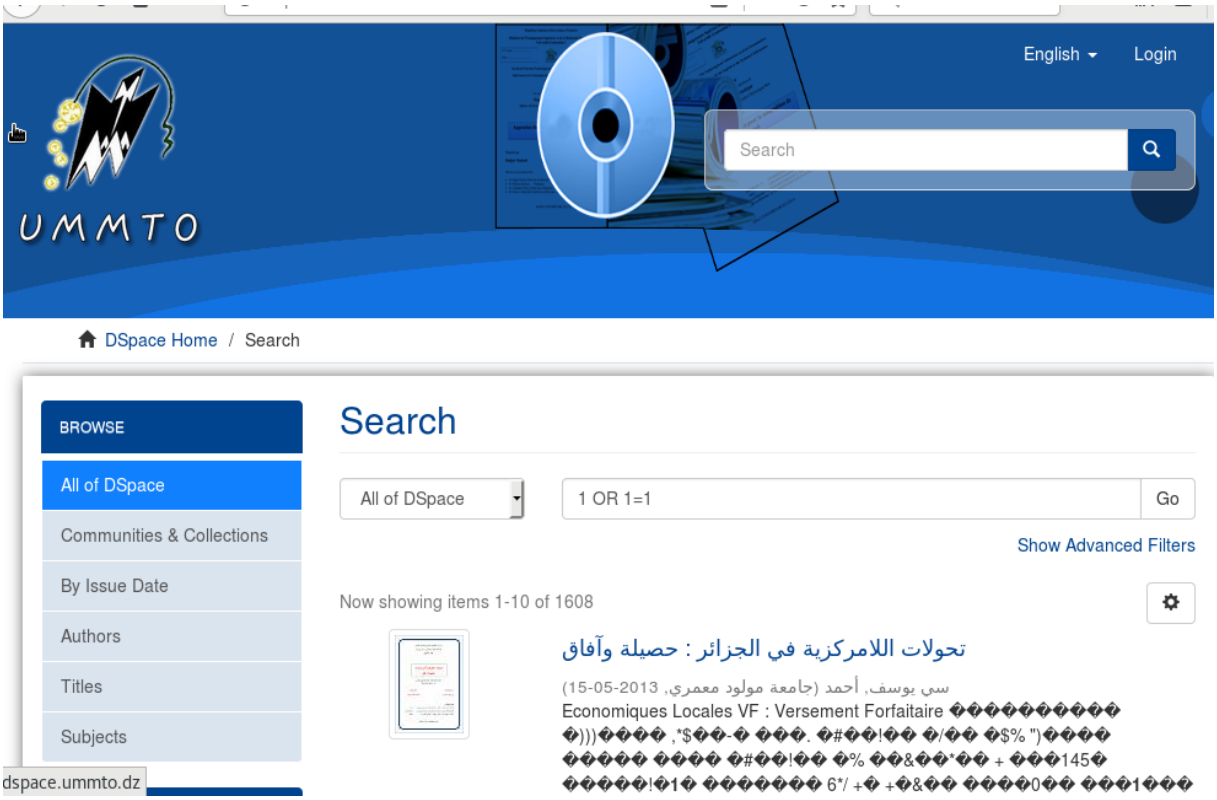
Supposons que notre visiteur n'est pas du genre visiteur gentil mais son ultime but est de Trouver une faille susceptible de lui donner un accès root à notre base de données. C'est au niveau du formulaire qu'il va jouer sa malice à savoir des requêtes malveillantes Intentionnelles qui seront transmises par la méthode POST qu'il va essayer de leurrer notre code PHP. Justement la requête sera tout à fait juste mais d'une manière illégitime

a. Sans Modsecurity

Réalisation, analyse des résultats des attaques avant la mise en place de ModSecurity

- **Analyse de l'Attaque Injection SQL**

  On introduit le code :1 OR 1=1

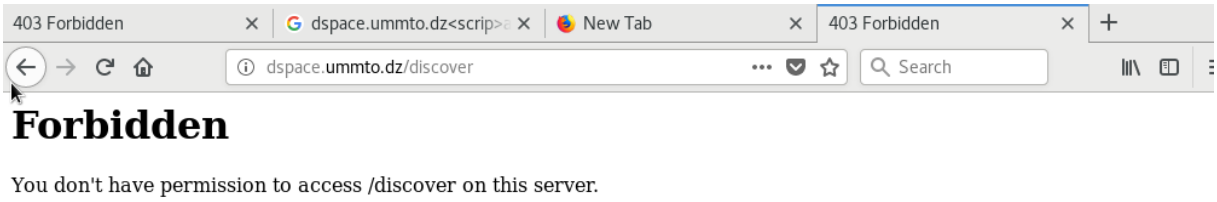


Après la soumission de la demande, le navigateur a répondu avec une liste de recherche.

b. Blocage de l'Attaque SQLI avec la présence de modsecurity

Maintenant nous utilisons le Firewall Applicatif Modsecurity pour détecter et bloquer les attaques courantes de l'application web. Nous allons activer l'option du paramètre SecRuleEngine en le mettant à On : **SecRuleEngin On**

On introduit de nouveau le code : **1 OR 1=1**



Attaque SQL Injection

La capture suivante montre le contenu du journal d'erreur d'apache, qui montre qu'il a rejeté la demande d'accès au serveur Web.

```
[Mon Sep 24 14:42:58.660797 2018] [:error] [pid 1467:tid 139754225964800] [client 10.0.2.15:33366]
 [client 10.0.2.15]
 ModSecurity: Access denied with code 403 (phase 2). detected SQLi using libinjection with
fingerprint '1&1' [file "/usr/share/modsecurity-crs/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"]
 [line "68"] [id "942100"] [rev "1"] [msg "SQL Injection Attack Detected via libinjection"]
 [data "Matched Data: 1&1 found within ARGS:query: 1 OR 1=1"] [severity "CRITICAL"]
 [ver "OWASP_CRS/3.0.0"] [maturity "1"] [accuracy "8"] [tag "application-multi"] [tag "language-multi"]
 [tag "platform-multi"] [tag "attack-sqli"] [tag "OWASP_CRS/WEB_ATTACK/SQL_INJECTION"]
 [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"]
 [hostname "dspace.ummto.dz"] [uri "/discover"] [unique_id "W6jb0n8AAQEAAAW7DKcAAAAX"],
 referer: http://dspace.ummto.dz/
```

- **Indique que la requête a été bloquée**
- **Chemin vers la règle déclenchée**
- **Evènement déclencheur de la règle de blocage**

Contenu error.log apache après l'activation de la règle de Blocage du SQL Injection

Dans cette capture ModSecurity nous signale une interdiction d'accès avec le code d'erreur 403. Il indique qu'après l'analyse de la phase 2 c'est à dire le contenu de la requête, il a détecté la présence de l'instruction 1 OR 1=1. Cette signature a été détecté par REQUESTS-942-APPLICATION-ATTACK-SQLI.conf particulièrement la ligne 68. Le contenu du journal d'audit de ModSecurity montre la raison du rejet de la demande d'apache car il a trouvé le modèle dans la règle.

```
--86fc092b-A--
[24/Sep/2018:14:42:58 +0200] W6jb0n8AAQEAAAW7DKcAAAAX 10.0.2.15 33366 10.0.2.15 80
--86fc092b-B--
POST /discover HTTP/1.1
Host: dspace.ummto.dz
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://dspace.ummto.dz/
Content-Type: application/x-www-form-urlencoded
Content-Length: 16
Connection: keep-alive
Upgrade-Insecure-Requests: 1

--86fc092b-C--
query=1+OR+1%3D1
--86fc092b-F--
HTTP/1.1 403 Forbidden
Content-Length: 217
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

--86fc092b-E--
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /discover
on this server.<br />
</p>
</body></html>

--86fc092b-H--
Message: Access denied with code 403 (phase 2). detected SQLi using libinjection with fingerprint '1&1' [fi
le "/usr/share/modsecurity-crs/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "68"] [id "942100"] [
rev "1"] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: 1&1 found within ARGS:
query: 1 OR 1=1"] [severity "CRITICAL"] [ver "OWASP_CRS/3.0.0"] [maturity "1"] [accuracy "8"] [tag "applica
tion-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-sqli"] [tag "OWASP_CRS/WEB_ATTACK/S
QL_INJECTION"] [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"
]
Apache-Error: [file "apache2_util.c"] [line 273] [level 3] [client %s] ModSecurity: %s%s [uri "%s"]%s
Action: Intercepted (phase 2)
Apache-Handler: proxy-server
Stopwatch: 1537792978659324 2606 (- - -)
Stopwatch2: 1537792978659324 2606; combined=1258, p1=275, p2=939, p3=0, p4=0, p5=43, sr=12, sw=1, l=0, gc=0
Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.9.1 (http://www.modsecurity.org/); OWASP_CRS/3.0.0.
Server: Apache
Engine-Mode: "ENABLED"
```

Date et l'heure

En-tête de la requête

- Corps de la requête

En-tête de la réponse

- Corps de la réponse

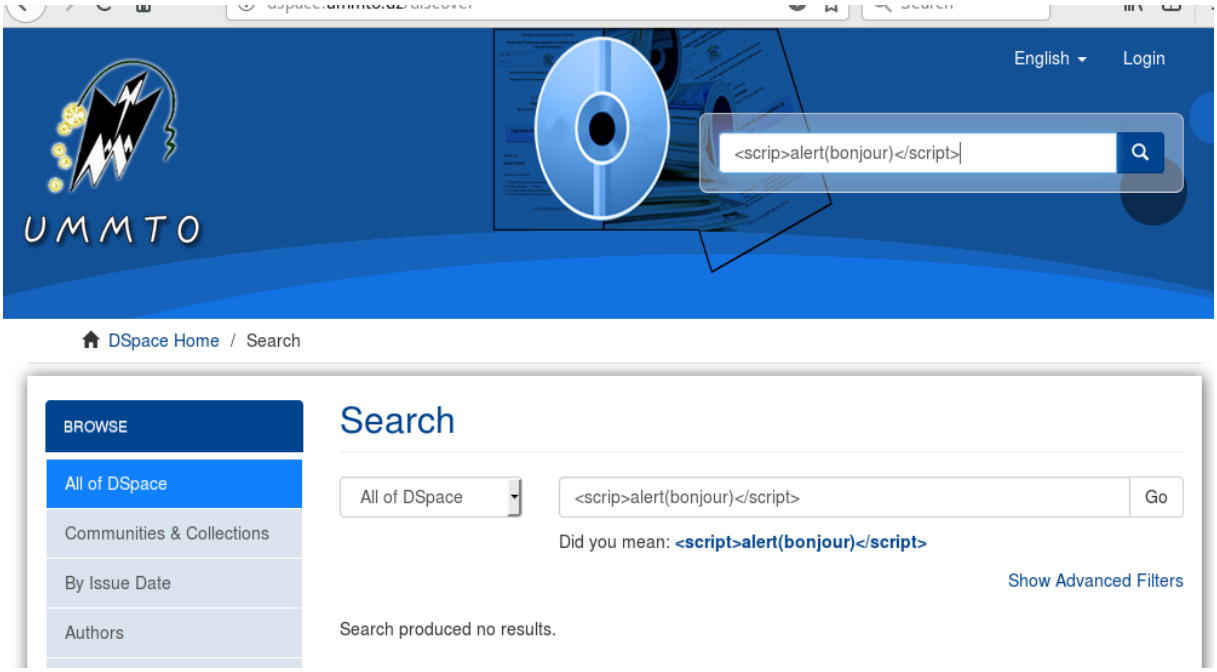
- Indique que la requête a été bloquée
- Chemin vers la règle déclenchée
- Evènement déclencheur de la règle de blocage

Fichier modsec_audit .log après l'activation de la règle de blocage SQLI

IV. 3.2. Analyse de l'attaque XSS

a. Sans Modsecurity

On introduit Code : **< script > alert (1) < /script >**

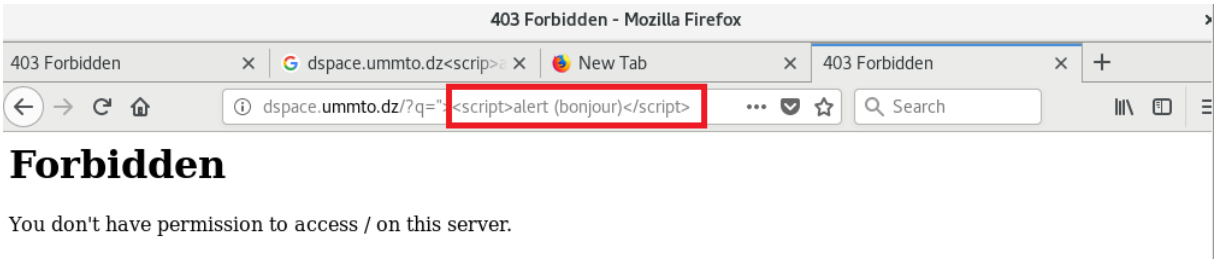


b. Blocage de l'Attaque XSS avec la présence de modsecurity

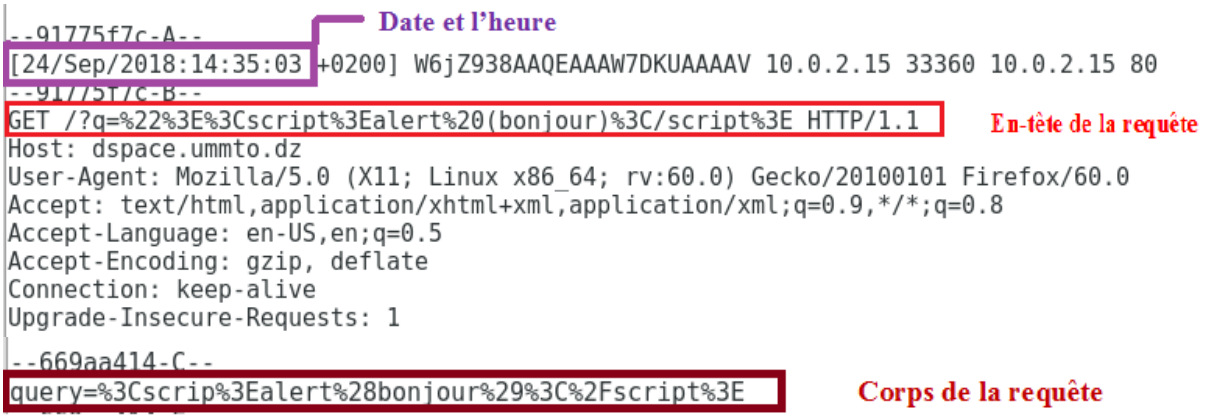
Maintenant nous utilisons le Firewall Applicatif ModSecurity pour détecter et bloquer cette attaque contre l'application web de type XSS.

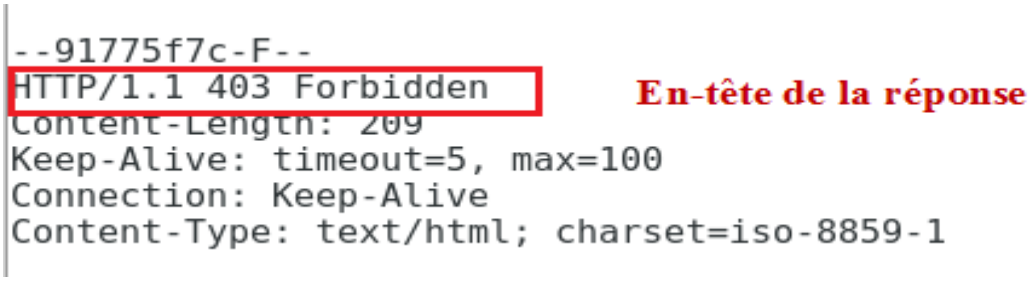
On introduit le code **:< script > alert (1) < /script >**

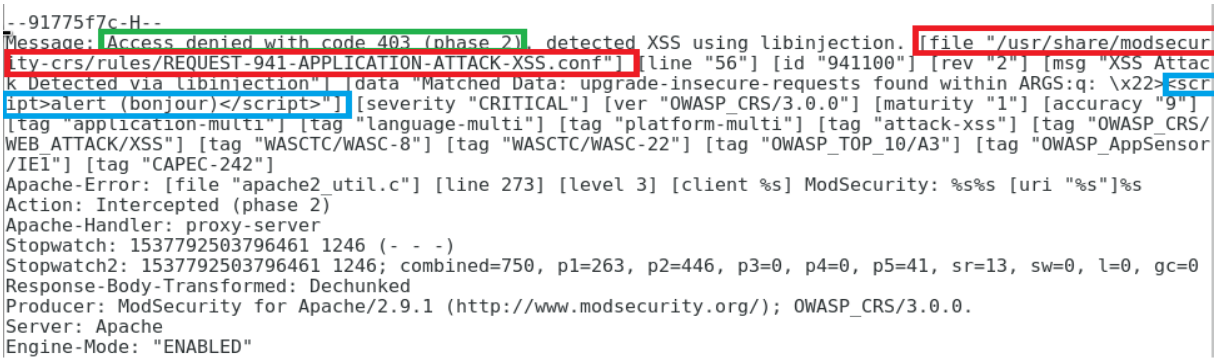
La Capture d'écran du Navigateur au moment d'une attaque XSS après activation des règles de blocage XSS :



Dans cette capture ModSecurity nous signale une interdiction d'accès avec le code d'erreur 403. Il indique qu'après l'analyse de la phase 2, c'est à dire le contenu de la requête il a détecté la présence de l'instruction <script>alert (bonjour) </script>. Cette signature a été détectée par REQUEST-941-APPLICATION-ATTACK-XSS.conf particulièrement la ligne '56'. Le contenu du journal d'audit de ModSecurity montre la raison du rejet de la demande d'apache.







Fichier modsec_audit .log après l'activation de la règle de blocage

IV.4. Conclusion

Ce chapitre a décrit la réalisation de notre WAF et les outils utilisés. Nous avons exploité les technologies Nftables, Modsecurity et Reverse proxy qui nous ont permis de configurer notre pare-feu, et également profité de cette phase pour approfondir nos connaissances en Apache et apprendre à utiliser Debian qui est un système d'exploitation très puissant, ainsi nous avons aussi appris à mieux utiliser, analyser et gérer les fichiers journaux en expliquant leur fonctionnement contre les attaques connues.

Conclusion Générale

Un pare-feu donc a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés ou interdits. Les recherches pour faire évoluer les technologies de filtrage sont nées du besoin de sécuriser les échanges réseaux. Pour améliorer ce filtrage il a été nécessaire de remonter dans les couches OSI, ce qui a été rendu possible grâce à une technologie logicielle et matérielle de plus en plus rapide. Comme on peut le constater, les firewalls possèdent de multiples capacités qui peuvent différer en fonction de leurs types. Cette multitude de solutions impose donc une étude rigoureuse de la sécurité avant être mise en place.

Notre démarche consiste à implémenter une solution basée sur les pare-feu pour filtrer le trafic qui circule dans le réseau.

Dans un premier temps, nous avons présenté les concepts fondamentaux des réseaux locaux, dans lequel nous avons fait un petit aperçu sur les différentes topologies et les équipements d'interconnexion des réseaux locaux. puis, nous avons mis un accent sur le modèle de référence OSI qui est la base de référence pour les réseaux locaux puis une étude sur la sécurité informatique nous a permis d'exposer un large panorama sur les différentes attaques qui peuvent affecter notre réseau, sans oublier les stratégies de sécurité, à savoir les pare-feu et leurs fonctionnement, les zones démilitarisées DMZs, les proxy, etc.

Ensuite, on a abordé les pare-feu leurs types, leurs fonctionnements et ces différentes architectures.

Puis on a présenté l'organisme d'accueil au sein duquel on a effectué notre stage, d'où on a eu l'occasion de voir leur architecture réseau et leur présenter une nouvelle architecture plus sécurisé en utilisant un WAF, puis on a consulté les différentes solutions WAF qui dominent sur le marché pour en extraire la solution qui répond à nos besoins.

En effet, la mise en place d'un pare-feu WAF nous a permis d'assurer la sécurité du réseau et notamment des serveurs web en filtrant les entrées et en contrôlant les sorties selon des règles définies par l'administrateur du réseau.

Donc Ce projet nous a permis d’acquérir une expérience personnelle et professionnelle et ne peut être que bénéfique. Ce fut une occasion pour se familiariser avec l’environnement du travail et de la vie professionnelle ainsi que d’élargir et d’approfondir nos connaissances sur l’administration des réseaux informatiques.

Listes des références

Bibliographie :

- **[14]** : A. JAQUEMIN, A. MERCIER, Les firewalls , 2ème édition, 2012.
- **[1]** : Boucherba Kadidja,Ziane Saloua << mémoire fin de cycle, mise en place d'un pare-feu d'entreprise>>, Bedjaia 2015 Open source.
- **[4]** : José DORDOIGNE. réseaux informatiques. Edition ENI, Février 2011.
- **[10]** : Jean-François pillou et Jean-Philippe. sécurité informatique.3eme Edition Dunod, Paris 2013.
- **[11]** : Tarek Abbes <<Thèse Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusion>>, IAEML Lorraine 2004.
- **[12]** : Yazid Fekhar <<mémoire fin d'étude ing.inf, la sécurité dans les réseaux>> Ummto 2006 /2007.
- **[15]** : Jean-François Pillou et Jean-Philippe Bay. Sécurité informatique.3ième édition, Dunod, Paris 2013.
- **[16] :** D. DROMARD, D. SERET, Architecture des réseaux, Pearson, 2ème édition, 2010.

Webographie :

- **[2]** : cours CISCO en ligne CyberSécurité.

  [3] : https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/techniques-et-systemes-de-transmission-en-reseaux-et-telecoms-42293210/protocoles-de-transmission-de-donnees-e7150/.
- **[5]** : http://hautrive.free.fr/reseaux/architectures/organisation-des-reseaux.html
- **[6]** : http://hautrive.free.fr/reseaux/architectures/protocoles-de-reseaux.html.
- **[7]** : http://mrim.forumpro.fr/t801-comparaison-des-modeles-osi-et-tcp-ip.
- **[8]** : http://www.case.lu .

- **[9]** : https://tcpip.goffinet.org/modeles_tcpip_et_osi.html#1-mod%C3%A8le-tcpip.
- **[13]:** cours ccna1 version 6 routing and switching
- **[17] :** http://firewalls.chez.com/chapitre2.html : le firewall, une technique de protection.
- **[18]**: http://fr.wikipedia.org/wiki/Application_web.
- **[19] :** Yakaferci, Définition : Serveur Web . http://www.yakaferci .com/definition/serveur-web/22/ .
- **[20] :** TELECOM LILLE 1,Les Serveurs d'Application .http://wapi ti.telecom-lille1.eu/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2001ttv0rveurs%20d'applications/1.html .
- **[21] :** comment Ça marche.net, Le Protocol HTTP . http://www.commentcamarche.net/contents/520-le-protocole-http .
- **[22] :** Adventures in the programming jungle,Web Application Firewall (WAF) – l'etat de l'art (2). http://itblog.adrian.citu.name/2012/07/15/waf2/ .
- **[23] :** Jérôme Saiz, Bien choisir son WAF.http://www.securityvibes.fr.
- **[24]:** IMPERVA, SecureSphere Web Application Firewall-Real-time protection against web at-tacks .http://www.imperva.com/Products/WebApplicationFirewall
- **[25]:** f5, Web Application Firewall. https://f5.com/glossary/web-application-firewall.
- **[26]:** Trustwave,ModSecurity Overview http://modsecurity.org/about.html
- **[27] :** AQTRONiX, AQTRONiX . https://www.aqtronix.com/?PageID=99
- **[28]**: virtualbox, https://www.virtualbox.org/
- **[29]:** Debian 9 Stretch released June 17th, 2017https://www.debian.org/News/2017/20170617
- **[30] :** Apache http server project, http://httpd.apache.org/
- **[31] :** It-connect,https://www.it-connect.fr/chapitres/presentation-de-nftables/
- **[32]:** Ivan, R. (2012). MODSECURITY HANDBOOK : The Complete Guide to Securing YourWeb Applications. Feisty Duck

- **[33] :** Dimitri ségard, Fonctionnement et mise en place d’un reverse proxy sécurisé avec Apache, 8 mai 2011.

Résumé

La mise en ligne des services via les applications web est devenue une action indispensable voir naturelle pour n'importe quelle entreprise, banque, magasin, ou fournisseur désirant la disponibilité permanente vis-à-vis de ses clients. Dans le but de satisfaire la disponibilité permanente, leurs serveurs sont régulièrement victimes d'attaques visant soit à les rendre défaillants soit à accéder aux données sensibles qu'ils contiennent. Il est devenu primordial de les protéger contre des actes malveillants et aux attaques applicatives.

Les attaques de types applicatifs à l'occurrence de XSS et SQLI sont reconnues comme étant des vulnérabilités qui ne dépondent pas du réseau mais plutôt de l'application. Ce qui implique que des mécanismes de sécurité comme les pare-feu ou les IDS/IPS sont efficaces pour éliminer un volume important de menaces ciblant les couches inférieures ils s'avèrent nettement moins
adaptés à la protection contre les menaces de plus en plus ciblées et spécialisées qui affectent désormais les applications des entreprises.

En effet, les pare-feu comme Iptable, Nftable offrent une protection par rapport à l'en-tête, ils ont en charge d'autoriser la transmission des paquets en fonction de leur source et leur destination, mais n'arrivent pas à analyser leur contenu. Ce qui fait, ils sont incapables d'obtenir des informations sur les attaques et les attaquants.

Aussi, les IDS/IPS n'ont pas l'habilité de comprendre la logique du protocole HTTP et donc n'arrivent pas à détecter si une requête est normale ou malicieuse dans la couche application. Et par la suite n'arrivent pas à détecter ni à prévenir les nouvelles attaques sans signatures. Devant une telle situation, l'utilisation d'un pare-feu applicatif (WAF : Web Application Firewall en anglais) s'avère très nécessaire pour assurer la disponibilité des organismes quellesque soit leurs activités vis-à-vis leurs clients.

L'utilisation des WAFs permet le «Virtual Patching », donc, chaque requête est contrôléeavant d'être envoyée au serveur. Si la requête est considérée comme valide, elle sera relayéeau serveur. En revanche, si le WAF découvre un contenu dangereux dans cette requête, il y répondra sans solliciter le serveur concerné.

Le centre de calcul et réseau est l'un des organismes qui a compris cette problématique ,et qui travaille sans relâche pour mettre en œuvre cette nouvelle technologie de traitement, pour assurer une gestion plus sécurisé, plus fiable, plus rigoureuse, moins fastidieuse, et pour minimiser le risque d'erreurs et assurer la disponibilité de l'information à toute éventuelle demande.

Les mots clés :

Filtrage, Trafic réseau, Les réseaux, Filtrage trafic réseaux, Filtrage des paquets, Sécurité des réseaux, Les pare-feu, Les firewalls, Pare-feu applicatif, Pare-feu réseau.