ESPRIT UNIVERSITY

and

Landshut Iot Lab

Surveillance et Sécurité DNS

November 18, 2024

Rapport de stage

Dhia el hak Rached

Supervised by:
Khelil Abdelmajid

Contents

1 Introduction

À l'ère numérique actuelle, les menaces informatiques sont omniprésentes. Les attaques DNS représentent une menace persistante. Ce rapport se concentre sur la détection de ces attaques en utilisant l'apprentissage profond. Avec l'IoT Innovation Lab à la Hochschule Landshut, j'ai travaillé sur ce sujet. Ce rapport résume mes travaux, méthodes, résultats et recommandations.

2 Remerciment

Je tiens à exprimer ma plus profonde gratitude à Monsieur Abdelmajid, un maître de stage remarquable, au sein de l'IoT Innovation Lab de l'Université des Sciences Appliquées de Landshut. Dès le premier jour, il m'a accueilli avec une chaleur exceptionnelle et a démontré un engagement indéfectible à mon égard, ce qui a grandement enrichi mon expérience de stage. Son mentorat exemplaire et sa profonde connaissance du domaine de l'Internet des Objets ont été essentiels à mon développement académique et professionnel.

Sous sa tutelle, j'ai acquis une compréhension approfondie des technologies de pointe et développé des compétences pratiques qui sont cruciales dans le secteur innovant de l'IoT. La patience et la perspicacité de Monsieur Abdelmajid m'ont permis de naviguer avec assurance dans mes projets de recherche et de contribuer de manière significative à l'équipe.

Je suis infiniment reconnaissant pour sa disponibilité constante et ses conseils avisés, qui ont façonné de manière indélébile cette période formatrice de ma vie. Monsieur Abdelmajid a non seulement été un excellent mentor, mais aussi une source d'inspiration constante.

Je vous remercie sincèrement, Monsieur Abdelmajid, pour tout ce que vous avez partagé avec moi durant ce stage. Votre influence perdurera bien au-delà de cette expérience enrichissante.

3 Presentation de Laboratoire

L'IoT Innovation Lab à la Hochschule Landshut est un laboratoire de recherche et d'enseignement spécialisé dans les technologies de l'Internet des objets (IoT). Il est situé sur le campus de la Hochschule Landshut à Landshut, en Allemagne. Le laboratoire est équipé d'une variété de matériel et de logiciels IoT, y compris des capteurs, des actionneurs, des gateways, des serveurs, des applications et des outils de développement. Il est utilisé par les étudiants, les chercheurs et les partenaires industriels pour développer des prototypes et des solutions IoT innovantes. L'IoT Innovation Lab de l'Université des Sciences Appliquées de Landshut offre un environnement d'apprentissage et de recherche véritablement innovant. Sa singularité réside dans sa capacité à dépasser les limites géographiques, ce qui permet à des étudiants du monde entier d'y accéder. La mondialisation encourage une coopération sans précédent entre des individus créatifs provenant de différentes cultures. Au cœur de ce laboratoire, des technologies de pointe, telles que la réalité augmentée, sont utilisées pour améliorer l'expérience des chercheurs et des étudiants. La réalité augmentée crée un pont entre le monde numérique et la réalité physique en permettant aux utilisateurs d'interagir de manière immersive avec des environnements virtuels. Cette méthode innovante permet de tester, de simuler et d'expérimenter des concepts IoT de manière plus tangible

4 Presentation de Stage

Mon stage avait pour objectif de m'immerger dans l'analyse détaillée des paquets DNS et la détection des attaques de DNS spoofing. J'ai eu l'opportunité de travailler au sein de cette laboratoire , où j'ai découvert une équipe passionnée et des défis captivants. Chaque jour, j'approfondissais ma compréhension des techniques avancées pour analyser le trafic DNS et prévenir des attaques sophistiquées. Cette expérience a été bien plus qu'une simple formation technique. J'ai appris à anticiper et bloquer les menaces avec l'aide de mes collègues, ce qui m'a permis de mieux comprendre les enjeux réels de la cybersécurité. Ce stage m'a non seulement offert des compétences précieuses dans un domaine en constante évolution, mais il m'a aussi ouvert les yeux sur l'importance cruciale de protéger les infrastructures numériques dans le monde d'aujourd'hui.

Figure 1: Landshut Iot Lab

5 TÂCHES ET RESPONSABILITÉS

5.1 -ANALYSE DE DOCUMENTS SUR LES ATTAQUES DNS ET LE DNS SPOOFING

Mon travail a principalement consisté à étudier en profondeur les articles de recherche portant sur les attaques DNS, en particulier le DNS spoofing, et les techniques récentes de protection. Ces recherches m'ont permis d'explorer des stratégies avancées utilisées par les attaquants pour manipuler les requêtes DNS et les solutions déployées pour prévenir ces menaces. J'ai passé de nombreuses heures à analyser des travaux académiques afin de mieux comprendre les vulnérabilités des systèmes DNS et les méthodes de détection et de blocage de ces attaques. Cette revue de la littérature m'a offert une vision claire des défis actuels en matière de cybersécurité et des innovations récentes pour renforcer la résilience des infrastructures DNS face aux attaques. Cette démarche a également renforcé ma capacité à anticiper et à répondre aux évolutions des menaces dans un domaine en constante mutation .

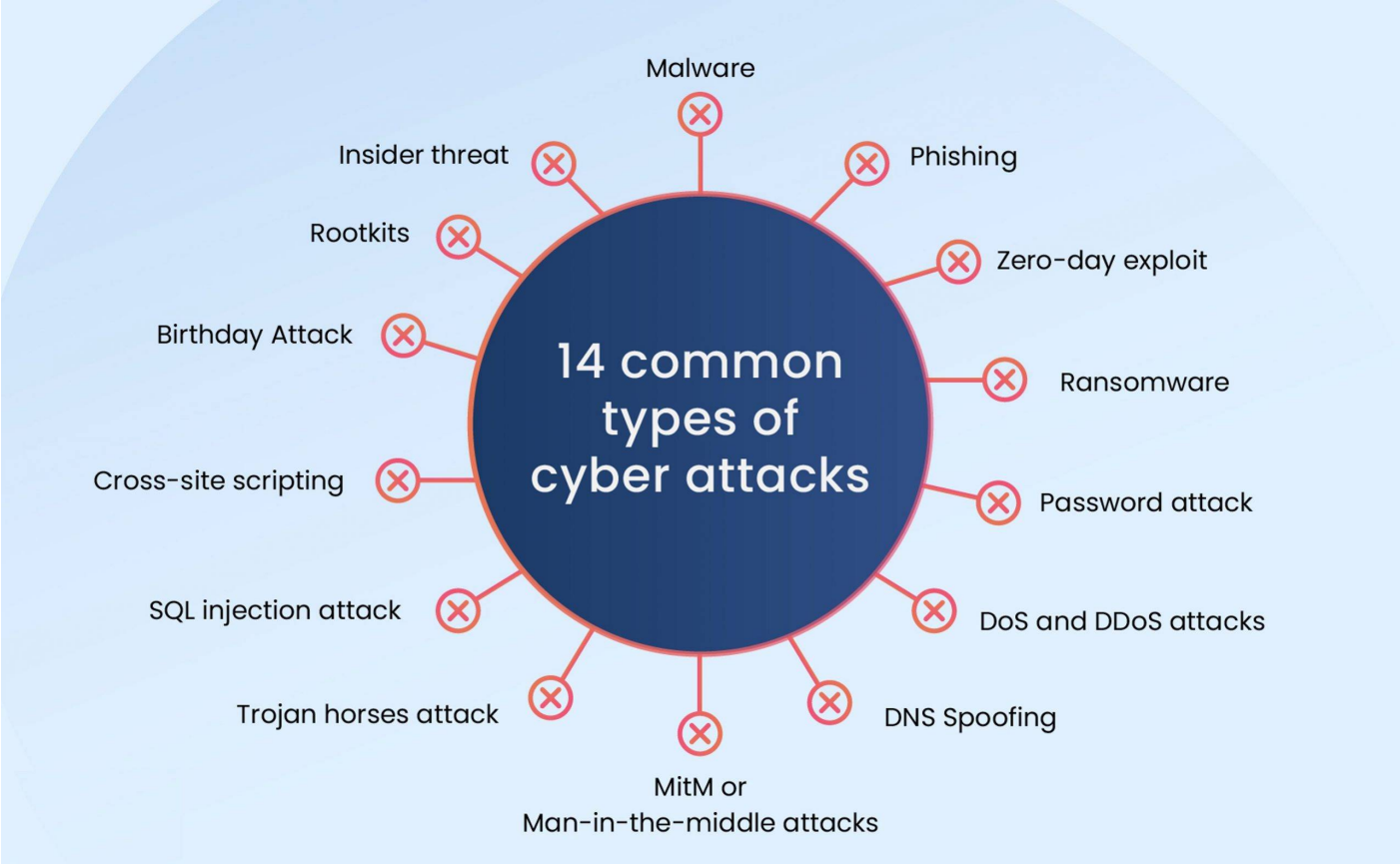


Figure 2: Types de cyberattaques

5.2 -COLLECTE ET SYNTHÈSE D'INFORMATIONS

Je me suis plongé dans la lecture des articles pertinents, récoltant les informations clés que j'ai ensuite synthétisées pour les partager avec mes collègues de recherche. Chaque détail nécessitait une attention particulière, exigeant une réflexion approfondie et une compréhension précise des sujets abordés. Mon rôle consistait à trier méticuleusement les données, à identifier les idées les plus importantes et à les présenter de manière claire et concise. En travaillant en étroite collaboration avec l'équipe, nous avons enrichi nos discussions et nos réflexions sur les meilleures pratiques et les tendances actuelles en matière de détection des malwares. C'était gratifiant de constater que mes efforts

contribuaient à orienter nos recherches vers des voies prometteuses, renforçant ainsi notre expertise collective.

5.3 CONTRIBUTIONS À LA RECHERCHE

En m'appuyant sur mes lectures, j'ai contribué à la création de recommandations et de pistes de recherche pour améliorer les techniques de détection de DNS Spoofing attacks face à l'utilisation croissante par les attaquants.

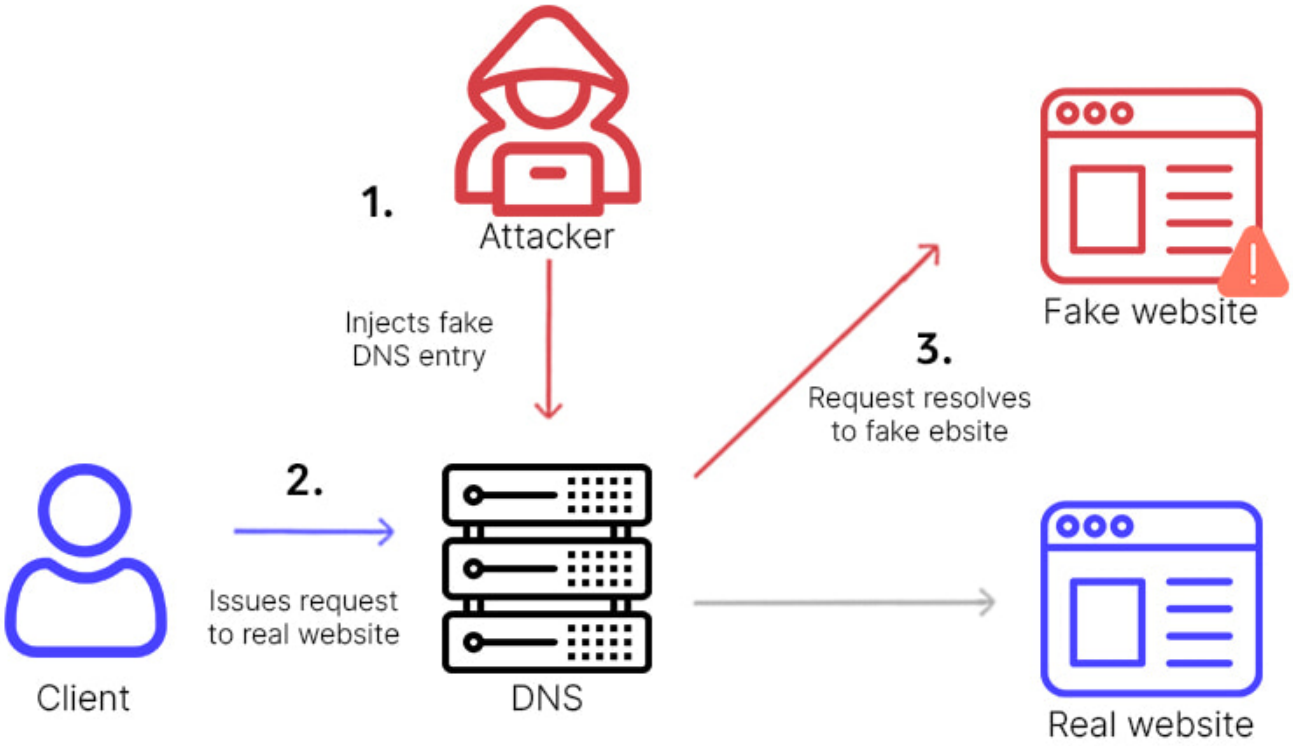


Figure 3: DNS Spoofing

5.4 -Decouvrir Qu'est-ce que le DNS?

Le DNS est un annuaire pour internet. Comme un annuaire téléphonique qui associe le nom d'une personne à son numéro de téléphone, le DNS associe les noms de domaine aux adresses IP, permettant ainsi la navigation sur internet sans nécessiter la mémorisation de séquences numériques complexes correspondant à chaque site web.

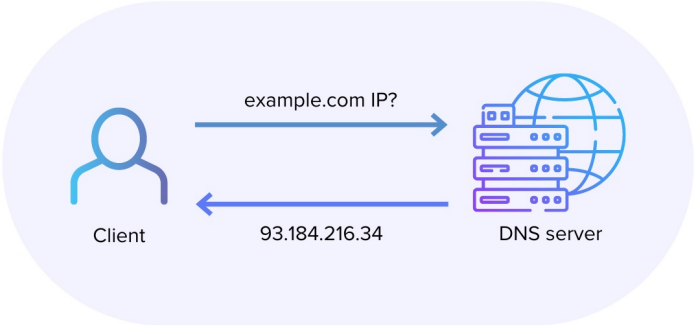


Figure 4: Serveur DNS

5.5 Analyse approfondie du fonctionnement du DNS?

Le processus de résolution DNS implique la conversion d'un nom d'hôte en adresse IP « au format informatique". Chaque appareil relié à Internet se voit attribuer une telle adresse. C'est cette adresse qui permet de le localiser sur Internet, c'est-à-dire de trouver l'appareil approprié, en quelque sorte comme une adresse dans une rue nous permet de trouver un domicile. Cependant, lorsque l'utilisateur recherche une page web spécifique, une traduction doit être effectuée entre l'adresse de la page web que l'utilisateur saisit dans son navigateur , dans ce cas ci « example.com » et l'adresse que la machine va utiliser pour rechercher pour nous, la page web. Alors, comment cela fonctionne, dans ce cas ci; « example.com » ? Pour comprendre comment cela fonctionne, il est important de comprendre les différents composants physiques auquel une requête DNS est confrontée. Du point de vue de l'utilisateur, la recherche DNS s'effectue « en arrière-plan » et ne nécessite pas d'interaction de l'ordinateur de l'utilisateur à l'exception de la demande initiale.

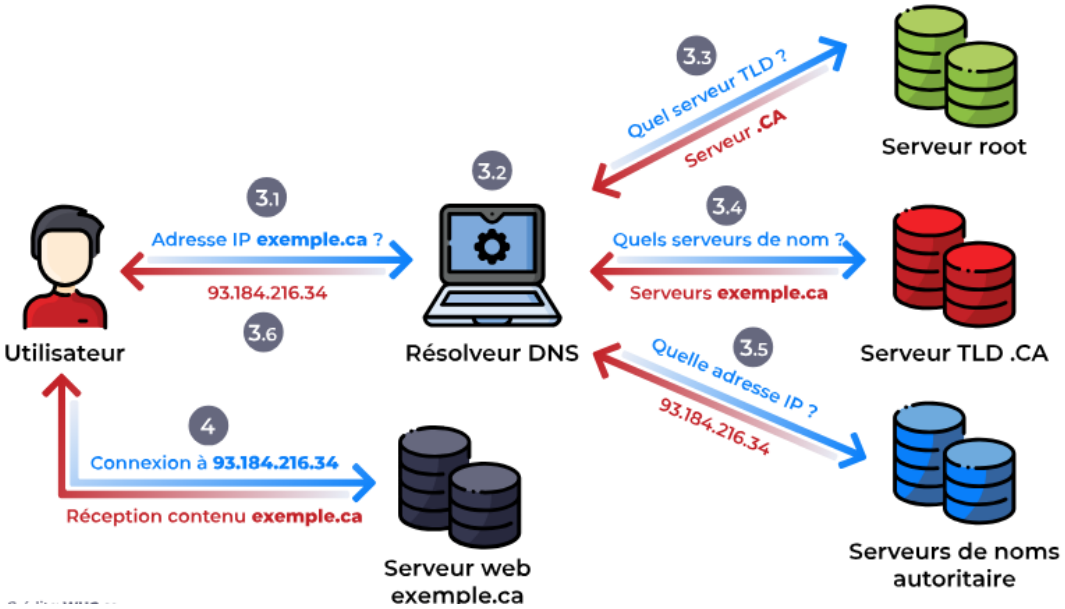


Figure 5: le mechanism de Serveur DNS

5.5.1 Quatre serveurs DNS sont impliqués dans le chargement d'une page web

Récurseur DNS : le récurseur peut être considéré comme un bibliothécaire à qui l'on demande d'aller chercher un livre particulier quelque part dans une bibliothèque. Il s'agit d'un serveur conçu pour recevoir les requêtes des ordinateurs client par l'intermédiaire d'applications, comme les navigateurs web. Généralement, le récurseur se charge ensuite d'effectuer les requêtes supplémentaires nécessaires à la résolution de la requête DNS du client.
Serveur de noms racine : le serveur racine constitue la première étape de la traduction (résolution) des noms d'hôtes lisibles par l'humain en adresses IP. Il s'agit en quelque sorte du catalogue d'une bibliothèque : il renvoie vers les différents rayonnages de livres et sert généralement de référence pour trouver d'autres emplacements plus spécifiques.
Serveur de noms TLD : le serveur de domaine de premier niveau (TLD, top level domain) peut être considéré comme un rayonnage spécifique au sein d'une bibliothèque. Étape suivante dans la recherche d'une adresse IP spécifique, ce serveur de noms héberge la dernière partie d'un nom d'hôte (dans « example.com », le serveur TLD est ainsi « .com »).
Serveur de noms de référence : ce dernier serveur de noms peut être considéré comme un dictionnaire situé sur un rayonnage. Il rend possible la traduction d'un nom spécifique sous forme d'une définition. Le serveur de noms de référence constitue la dernière étape d'une requête effectuée au serveur de noms. Si le serveur de noms de référence a

accès à l'enregistrement demandé, il renvoie l'adresse IP du nom d'hôte demandé au récurseur DNS (le bibliothécaire) qui a effectué la requête initiale.

5.6 Approfondir les étapes d'une recherche DNS

Dans la plupart des situations, le DNS se charge de la traduction d'un nom de domaine afin d'aboutir à l'adresse IP appropriée. Pour comprendre le fonctionnement de cette opération, il peut s'avérer utile de suivre le parcours d'une recherche DNS, depuis son émission dans le navigateur web jusqu'à son traitement par le processus de recherche DNS lui-même, avant de revenir à son origine dans le sens inverse. Examinons les différentes étapes.

Remarque : les informations d'une recherche DNS sont souvent mises en cache localement, dans l'ordinateur à l'origine de la requête ou à distance au sein de l'infrastructure DNS. Une recherche DNS se compose généralement de 8 étapes. Lorsque les informations DNS sont mises en cache, le processus de recherche DNS ignore certaines étapes, accélérant d'autant son traitement. L'exemple ci-dessous décrit l'ensemble des 8 étapes en l'absence d'informations mises en cache.

Les 8 étapes d'une recherche DNS :

1. Un utilisateur saisit « example.com » dans un navigateur web. La requête est acheminée via Internet et reçue par un résolveur DNS récursif.
2. Le résolveur interroge alors un serveur de noms racine DNS (.).
3. Ce dernier répond au résolveur avec l'adresse d'un serveur DNS de domaine de premier niveau (TLD) (comme « .com » ou « .net ») sur lequel sont conservées les informations relatives à ses domaines. Ainsi, lors d'une recherche portant sur « example.com », la requête est dirigée vers le TLD « .com ».
4. Le résolveur émet ensuite une requête vers le TLD « .com ».
5. Le serveur TLD répond alors avec l'adresse IP du serveur de noms de domaine : « example.com ».
6. Le résolveur récursif envoie une requête au serveur de noms de domaine.
7. Le serveur de noms de domaine renvoie ensuite l'adresse IP du site example.com vers le résolveur.
8. Enfin, le résolveur DNS répond au navigateur web avec l'adresse IP du domaine initialement demandé.

Une fois que les 8 étapes de la recherche DNS ont renvoyé l'adresse IP d'example.com, le navigateur peut émettre la requête à la page web :

9- Le navigateur envoie une requête HTTP à l'adresse IP. **10-** Le serveur situé à cette adresse IP renvoie la page web à afficher dans le navigateur (étape 10).

6 DNS Packet Structure

Un paquet DNS est composé de plusieurs sections clés. D'abord, l'en-tête qui contient des informations essentielles telles que l'identifiant de transaction et les drapeaux indiquant s'il s'agit d'une requête ou d'une réponse. Ensuite, la section de question spécifie le domaine concerné ainsi que le type de requête (par exemple, une adresse IP). La section de réponse fournit les données demandées, comme l'adresse IP du domaine. Enfin, les sections d'autorité et d'informations supplémentaires peuvent aussi apparaître, offrant des détails sur les serveurs de noms ou des informations supplémentaires nécessaires au traitement de la requête

6.1 En-tête

L'en-tête est présent dans chaque paquet DNS, qu'il s'agisse d'une requête ou d'une réponse, et contient des informations générales sur les opérations DNS.

6.1.1 Identification (16 bits)

Un identifiant unique pour faire correspondre les requêtes et les réponses.

6.1.2 QR

Position : Bit 0
Indique si le paquet est une requête (0) ou une réponse (1).

6.1.3 Opcode

Position : Bits 1–4
Spécifie le type de requête. Les valeurs courantes incluent :

- 0 : Requête standard (la plus courante).
- 1 : Requête inverse (rarement utilisée).
- 2 : Demande de statut du serveur.

6.1.4 AA

Position : Bit 5
Indique si la réponse provient d'un serveur autoritaire. Mis à 1 si le serveur est autoritaire.

6.1.5 TC

Position : Bit 6
Tronqué : Indique si le message DNS est trop volumineux pour un seul paquet et a été tronqué. Mis à 1 en cas de troncature.

6.1.6 RD

Position : Bit 7
Récursion demandée : Mis à 1 par le client pour demander la récursion lors d'une requête.

6.1.7 RA

Position : Bit 8
Récursion disponible : Indique, dans une réponse, si le serveur supporte les requêtes récursives. Mis à 1 si la récursion est disponible sur le serveur.

6.1.8 Z

Position : Bits 9–11
Bits réservés pour un usage futur. Ils doivent toujours être mis à 0.

6.1.9 RCode

Position : Bits 12–15
Code de réponse : Indique le résultat de la requête. Les valeurs possibles incluent :

- 0 : Pas d'erreur.
- 1 : Erreur de format (le serveur n'a pas pu comprendre la requête).
- 2 : Échec du serveur (le serveur n'a pas pu traiter la requête).
- 3 : Erreur de nom (le nom de domaine n'existe pas).
- 4 : Non implémenté (le type de requête n'est pas pris en charge).
- 5 : Refusé (le serveur refuse de répondre à la requête).

6.2 En-tête

L'en-tête est présent dans chaque paquet DNS, qu'il s'agisse d'une requête ou d'une réponse, et contient des informations générales sur les opérations DNS.

6.2.1 Identification (16 bits)

Un identifiant unique pour faire correspondre les requêtes et les réponses.

6.2.2 QR

Position : Bit 0
Indique si le paquet est une requête (0) ou une réponse (1).

6.2.3 Opcode

Position : Bits 1–4
Spécifie le type de requête. Les valeurs courantes incluent :

- 0 : Requête standard (la plus courante).
- 1 : Requête inverse (rarement utilisée).
- 2 : Demande de statut du serveur.

6.2.4 AA

Position : Bit 5
Indique si la réponse provient d'un serveur autoritaire. Mis à 1 si le serveur est autoritaire.

6.2.5 TC

Position : Bit 6

Tronqué : Indique si le message DNS est trop volumineux pour un seul paquet et a été tronqué. Mis à 1 en cas de troncature.

6.2.6 RD

Position : Bit 7

Récursion demandée : Mis à 1 par le client pour demander la récursion lors d'une requête.

6.2.7 RA

Position : Bit 8

Récursion disponible : Indique, dans une réponse, si le serveur supporte les requêtes récursives. Mis à 1 si la récursion est disponible sur le serveur.

6.2.8 Z

Position : Bits 9–11

Bits réservés pour un usage futur. Ils doivent toujours être mis à 0.

6.2.9 RCode

Position : Bits 12–15

Code de réponse : Indique le résultat de la requête. Les valeurs possibles incluent :

- 0 : Pas d'erreur.
- 1 : Erreur de format (le serveur n'a pas pu comprendre la requête).
- 2 : Échec du serveur (le serveur n'a pas pu traiter la requête).
- 3 : Erreur de nom (le nom de domaine n'existe pas).
- 4 : Non implémenté (le type de requête n'est pas pris en charge).
- 5 : Refusé (le serveur refuse de répondre à la requête).

6.3 Questions

La section des questions contient des demandes spécifiques concernant un nom de domaine. Chaque question se compose de :

- **Nom** : Le nom de domaine pour lequel la requête est effectuée.
- **Type** : Indique le type d'enregistrement (par exemple, A, AAAA, MX).
- **Classe** : Généralement IN pour Internet.

6.4 Réponses

La section des réponses contient les réponses aux questions posées. Chaque réponse se compose de :

- **Nom** : Le nom de domaine auquel l'enregistrement se rapporte.
- **Type** : Le type d'enregistrement (par exemple, A, AAAA, CNAME).
- **Classe** : Généralement IN pour Internet.
- **TTL** : Time To Live, le temps pendant lequel l'enregistrement peut être mis en cache.
- **Données de l'enregistrement** : Contient l'adresse IP ou d'autres données selon le type d'enregistrement.

6.5 Autorité

La section d'autorité fournit des informations sur les serveurs de noms qui sont autoritaires pour le domaine. Elle se compose de :

6.5.1 Éléments d'autorité

- **Nom** : Le nom de domaine auquel l'enregistrement se rapporte.
- **Type** : Indique le type d'enregistrement (généralement NS pour les serveurs de noms).
- **Classe** : Généralement IN pour Internet.
- **TTL** : Time To Live, le temps pendant lequel l'enregistrement peut être mis en cache.
- **Données de l'enregistrement** : Contient le nom du serveur de noms autoritaire.

6.6 Additionnel

La section additionnelle contient des enregistrements supplémentaires qui peuvent être utiles. Par exemple, cela peut inclure des enregistrements A pour les serveurs de noms mentionnés dans la section d'autorité. Elle est structurée de manière similaire à la section de réponses.

6.6.1 Éléments additionnels

- **Nom** : Le nom de domaine associé.
- **Type** : Le type d'enregistrement.
- **Classe** : Généralement IN pour Internet.
- **TTL** : Time To Live.
- **Données de l'enregistrement** : Contient des informations pertinentes selon le type.

7 Conception et Mise en Œuvre

7.0.1 Vue d'Ensemble de l'Architecture

Dans ce projet, j'ai conçu un outil avec une architecture modulaire, permettant ainsi flexibilité et évolutivité. Voici un diagramme de haut niveau de l'architecture du système :

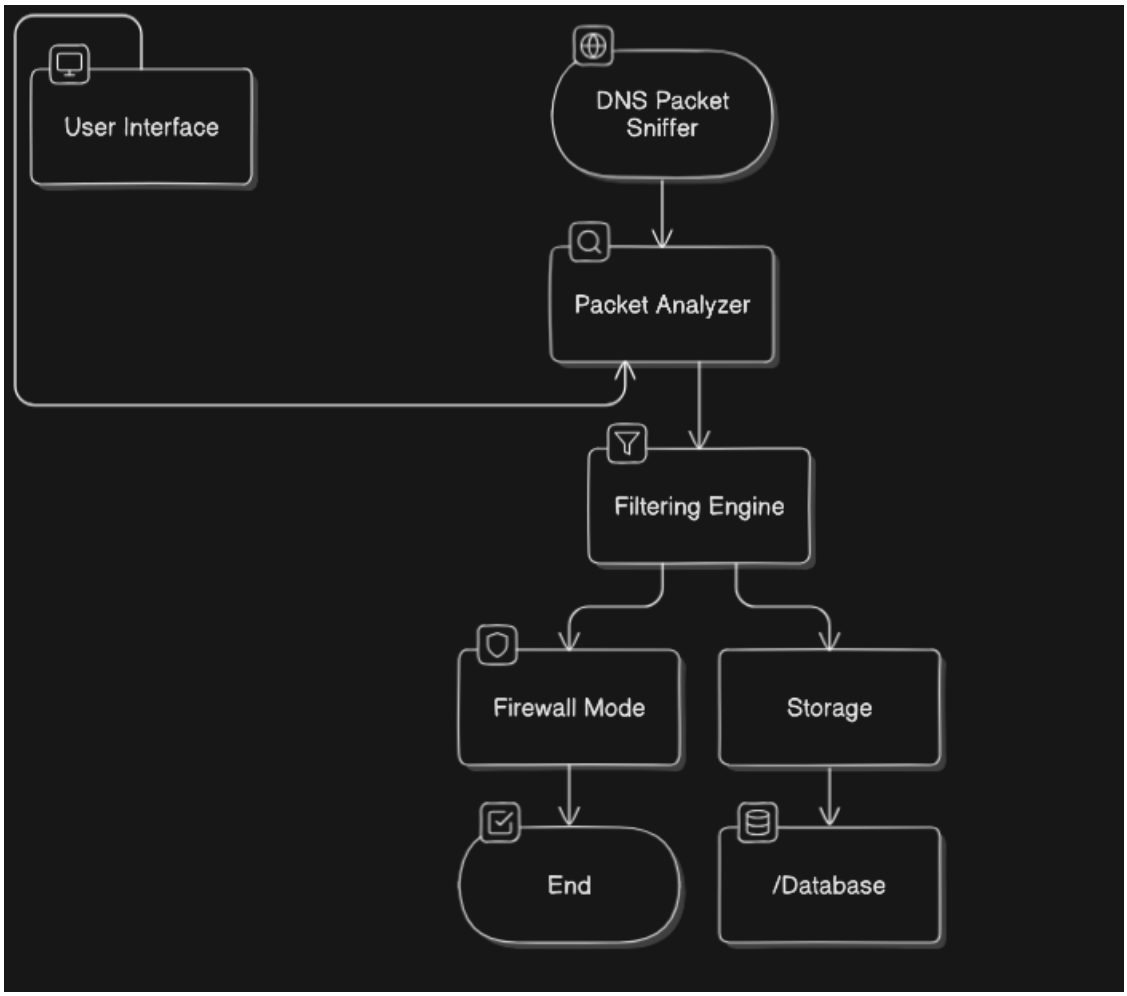


Figure 6: Architecture du Système pour la Surveillance et l'Analyse du Trafic DNS

7.1 Description des Composants

Sniffer de Paquets : Ce composant est responsable de la capture des paquets DNS sur l'interface réseau.

Analyseur : Ce module traite les paquets capturés et extrait les informations DNS pertinentes.

Interface Utilisateur : Une interface en ligne de commande permet de faciliter l'interaction avec l'utilisateur.

Stockage de Données : Ce module permet de sauvegarder les paquets capturés pour une analyse ultérieure.

7.2 Répartition des Composants

7.2.1 Sniffing de Paquets DNS

Pour capturer les paquets DNS dans mon outil, j'utilise la bibliothèque `Scapy`, un puissant outil Python. Voici comment cela fonctionne :

Écoute du Trafic Réseau : J'utilise la fonction `sniff()` de `Scapy` pour capturer les paquets sur l'interface réseau.

Filtrage des Paquets DNS : Le paramètre `filter` permet de capturer uniquement les paquets DNS, en utilisant la syntaxe BPF (Berkeley Packet Filter). Par exemple :

```
packets = sniff(filter='udp port 53', prn=process_packet)
```

Cela permet de capturer les paquets UDP sur le port 53, qui est utilisé pour les requêtes DNS.

Figure 7: DNS UDP (port 53)

7.2.2 Sortie Verbose

J'ai également implémenté un mode verbose pour améliorer l'expérience utilisateur. En activant ce mode, je peux fournir des informations détaillées sur les paquets capturés, telles que les IP source et destination, les types de requête et les codes de réponse. Voici un exemple de sortie en mode verbose :

```
[INFO] Requête DNS capturée : www.example.com
[INFO] IP Source : 192.168.1.5
[INFO] IP Destination : 8.8.8.8
[INFO] Type de Requête : A
```

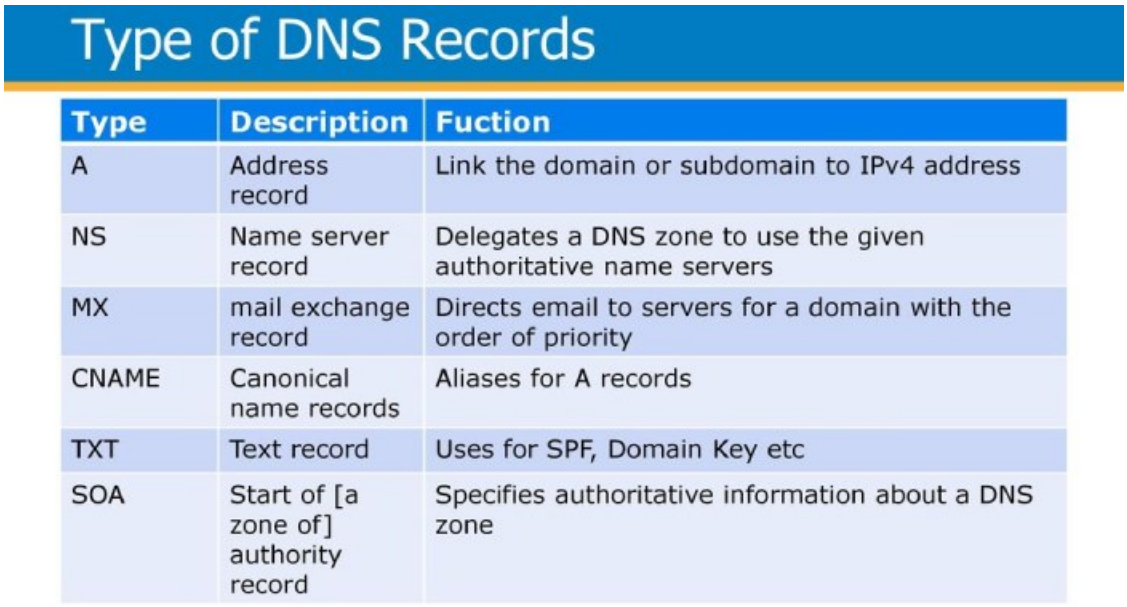
J'utilise également la bibliothèque Colorama pour mettre en évidence les informations, rendant ainsi la sortie plus lisible.

7.2.2.1 Analyse de l'IP Cible Dans ce projet, j'ai ajouté la fonctionnalité qui permet aux utilisateurs de spécifier une adresse IP cible pour surveiller les réponses DNS. Cela me permet de :

7.2.3 Analyse des Types DNS

Je voulais que cet outil puisse analyser différents types de requêtes DNS, comme :

- **A (Adresse)** : Résolution des noms d'hôtes en adresses IP.
- **AAAA (Adresse IPv6)** : Résolution des noms d'hôtes en adresses IPv6.
- **CNAME (Nom Canonique)** : Alias d'un nom de domaine vers un autre.
- **MX (Échange de Mail)** : Spécification des serveurs de messagerie pour un domaine.



Type of DNS Records

Type	Description	Fuction
A	Address record	Link the domain or subdomain to IPv4 address
NS	Name server record	Delegates a DNS zone to use the given authoritative name servers
MX	mail exchange record	Directs email to servers for a domain with the order of priority
CNAME	Canonical name records	Aliases for A records
TXT	Text record	Uses for SPF, Domain Key etc
SOA	Start of [a zone of] authority record	Specifies authoritative information about a DNS zone

Figure 8

Pour cela, j'ai mis en place un mécanisme qui vérifie le champ de type des paquets DNS et affiche les informations pertinentes à l'utilisateur.

7.2.4 DNS sur HTTPS (DoH)

Pour renforcer la sécurité, j'ai intégré la prise en charge du DNS sur HTTPS (DoH), qui crypte les requêtes DNS. J'utilise des bibliothèques comme `requests` pour envoyer des requêtes DNS via HTTPS et analyser les réponses JSON pour en extraire les informations DNS.

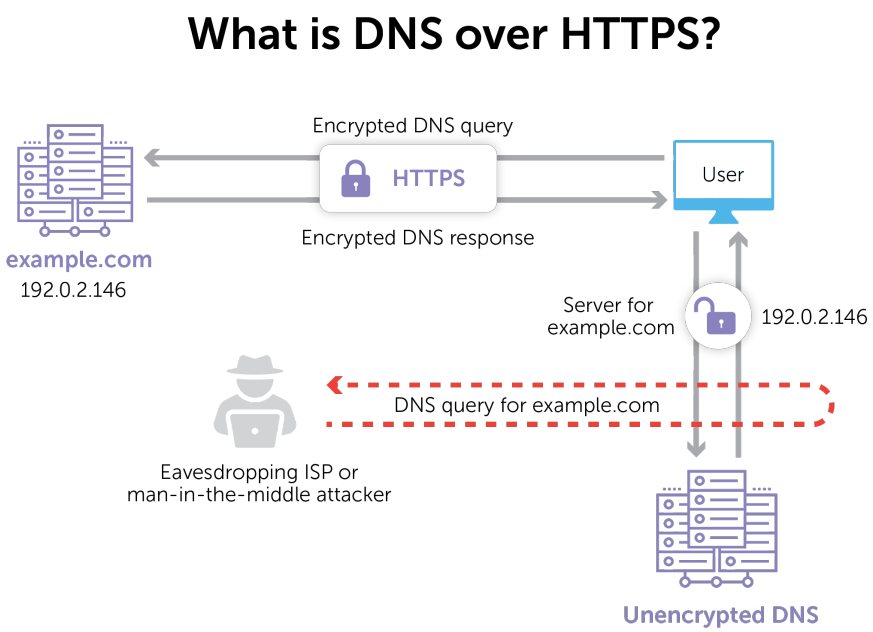


Figure 9

7.2.5 Surveillance de Domaines Cibles

Les utilisateurs de cet outil peuvent spécifier des domaines pour surveiller le trafic DNS associé. J'ai réalisé cela en permettant aux utilisateurs de fournir un nom de domaine via la ligne de commande, par exemple `-domain www.example.com`, ce qui permet de filtrer et d'afficher uniquement les requêtes et réponses DNS relatives au domaine spécifié.

7.2.6 Filtrage par Port et IP

Pour gérer le volume de données capturées, j'ai intégré des options de filtrage :

- **Filtrage par Port** : Les utilisateurs peuvent spécifier les ports à surveiller (comme le port UDP 53 pour le DNS).
- **Filtrage par IP** : Je permets également de filtrer les paquets par adresses IP source et destination, afin de se concentrer sur des hôtes ou services particuliers.

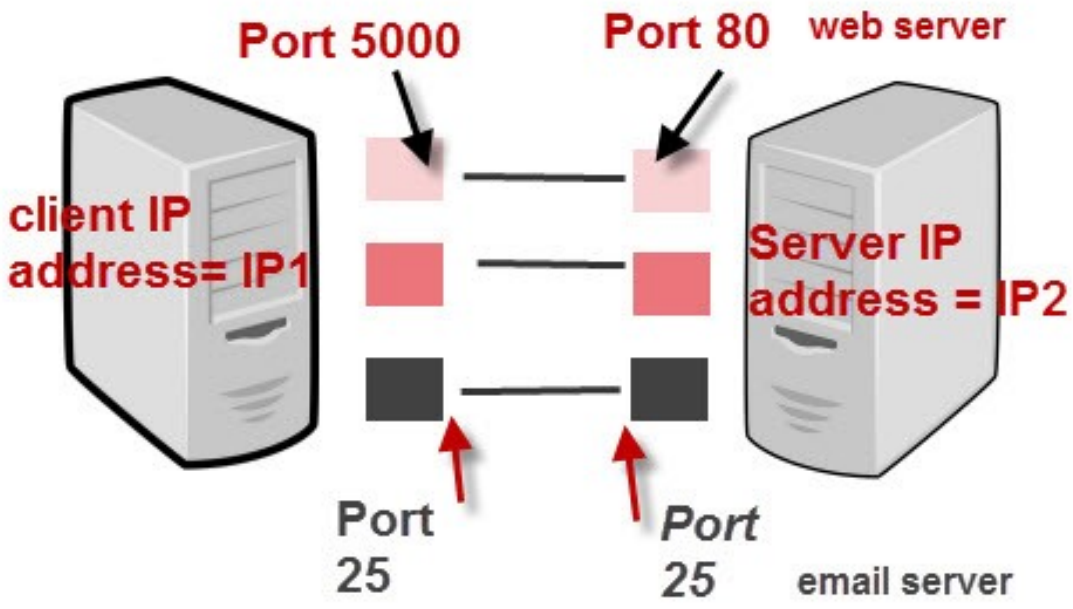


Figure 10

7.2.7 Sauvegarde des Fichiers PCAP

Cette outil que j'ai developpe offre la possibilité de sauvegarder les paquets capturés au format PCAP pour une analyse ultérieure. Cela est particulièrement utile pour :

- **Analyse Postérieure** : Les utilisateurs peuvent utiliser des outils comme Wireshark pour analyser les données.
- **Mise en Œuvre** : Un argument en ligne de commande (par exemple, `-save-pcap output.pcap`) permet de spécifier le nom du fichier pour la sauvegarde des paquets capturés.

Voici un exemple de code pour sauvegarder les paquets :

```
wrpcap('output.pcap', packets)
```

7.3 Composant Pare-feu dans mon `DNSSpoofingDetector`

Dans le cadre de mon projet, j'ai intégré un composant pare-feu dans mon outil `DNSSpoofingDetector`. Ce composant a pour rôle de surveiller et de détecter les tentatives potentielles d'usurpation DNS. Voici les principales fonctionnalités que j'ai mises en place :

7.3.1 Détection Basée sur un Seuil

J'ai implémenté un système de détection basé sur un seuil de requêtes DNS. J'ai configuré l'outil pour qu'il surveille le nombre de requêtes DNS provenant d'une adresse IP sur une période de temps donnée (par défaut, 50 requêtes en 60 secondes). Si ce seuil est dépassé, l'outil déclenche une alerte qui signale une possible attaque par usurpation DNS. Cette approche permet de repérer rapidement les anomalies dans le trafic DNS.

7.3.2 Surveillance en Temps Réel

J'ai développé le pare-feu de manière à ce qu'il traite les paquets DNS en temps réel. Il analyse chaque paquet en inspectant l'adresse IP source et l'ID de la transaction DNS afin de détecter des volumes de trafic inhabituellement élevés. Cela permet de répondre immédiatement à des situations potentiellement dangereuses, comme des attaques par déluge DNS.

7.3.3 Fenêtre Temporelle de Détection

J'ai également ajouté un mécanisme de gestion de la fenêtre temporelle. Le pare-feu utilise une fenêtre de temps configurable (par défaut, 60 secondes) pour évaluer les requêtes DNS. J'ai mis en place une fonction qui nettoie les anciennes requêtes hors de cette période, garantissant ainsi que seules les requêtes récentes soient prises en compte pour l'analyse.

7.3.4 Alertes Détailées

Lorsque le nombre de requêtes DNS dépasse le seuil que j'ai défini, l'outil génère une alerte détaillée. J'ai conçu cette alerte pour inclure des informations importantes, comme :

- L'adresse IP de la source suspecte.
- Le nombre de requêtes DNS effectuées au cours de la fenêtre temporelle.
- Les ID de transaction des requêtes DNS les plus récentes.
- L'horodatage de la dernière requête.

Ces informations permettent d'aider les administrateurs réseau à identifier et réagir rapidement aux tentatives d'attaque.

7.3.5 Paramètres Configurables

Enfin, j'ai rendu les paramètres de détection totalement configurables. J'ai prévu la possibilité de modifier :

- **Le seuil :** J'ai inclus une fonction `set_threshold()` pour ajuster le nombre maximum de requêtes autorisées dans la fenêtre temporelle, ce qui permet de personnaliser la sensibilité de l'outil.
- **La taille de la fenêtre temporelle :** Grâce à la fonction `set_window_size()`, l'utilisateur peut ajuster la durée de la période pendant laquelle le trafic DNS est surveillé, en fonction des besoins spécifiques du réseau.

Grâce à ces fonctionnalités, j'ai réussi à créer un composant pare-feu efficace qui peut identifier en temps réel les attaques par usurpation DNS tout en étant flexible et adaptable à différents environnements réseau.

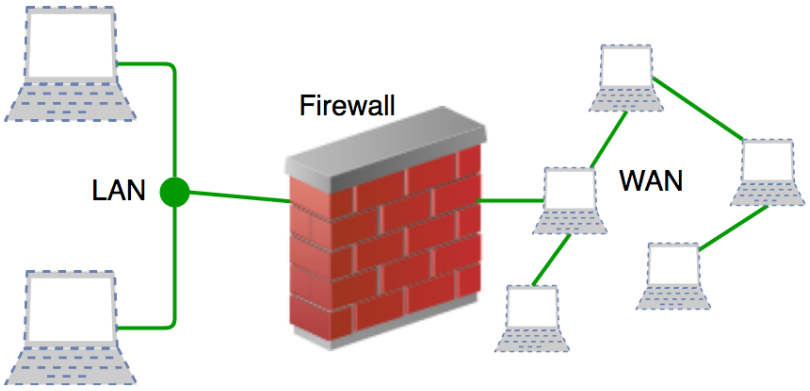


Figure 11

7.4 URL Analysis

J'ai intégré l'analyse des URL dans mon projet en utilisant les services de VirusTotal. Lorsqu'une URL est détectée dans le prompt de l'utilisateur, mon script l'extrait automatiquement à l'aide d'expressions régulières et la prépare pour l'analyse. L'URL est ensuite encodée en base64, ce qui est nécessaire pour l'analyse. Après avoir envoyé la requête, je récupère les résultats indiquant si l'URL présente des risques, tels que "malicieux", "suspicieux", "indétecté" ou "inoffensif". Ces résultats sont ensuite affichés à l'utilisateur pour l'informer sur la sécurité de l'URL et l'alerter en cas de menace. Cela permet de vérifier automatiquement la sécurité des URL et de prévenir l'utilisateur en cas de danger.

```
(.venv) PS C:\Users\MSI\Desktop\Internship\DNSWatch> python ChatBot.py
What do you want to do? analyze https://github.com/
Parsed action: analyze_url --url https://github.com/
Executing command: analyze_url --url https://github.com/
URL: https://github.com/
Analysis Status: {'malicious': 0, 'suspicious': 0, 'undetected': 25, 'harmless': 71, 'timeout': 0}
```

Figure 12

Figure 13

7.5 Développement d'un Chatbot NLP pour l'Analyse de Commandes DNS et la Détection d'URL Malveillantes

J'ai développé un chatbot NLP avancé intégré dans un projet de surveillance DNS, qui permet à l'utilisateur d'interagir de manière fluide avec des systèmes de sécurité réseau. Ce chatbot utilise des modèles de langage pour comprendre les intentions de l'utilisateur et générer des réponses contextuelles. À partir d'une entrée de texte, il peut effectuer plusieurs

actions, telles que l'analyse de trafic DNS, l'activation de pare-feu, ou la collecte et sauvegarde de paquets réseau. L'un des ajouts les plus intéressants est l'analyse en temps réel des URL, où le chatbot peut automatiquement détecter les liens présents dans les commandes de l'utilisateur et les analyser via un service externe pour détecter d'éventuelles menaces de sécurité. Ce processus permet d'ajouter une couche de sécurité supplémentaire en identifiant les risques liés aux sites web proposés.

Le système repose sur une combinaison de techniques de traitement du langage naturel, d'analyse de texte avec spaCy, et d'intégration avec des services tiers pour offrir une solution proactive de gestion de la sécurité réseau. Ce chatbot est ainsi un excellent exemple d'application de l'intelligence artificielle pour améliorer la cybersécurité, automatiser les tâches et offrir une interface conviviale pour l'utilisateur, tout en nécessitant une gestion minutieuse des aspects techniques et de sécurité pour en assurer l'efficacité à long terme.

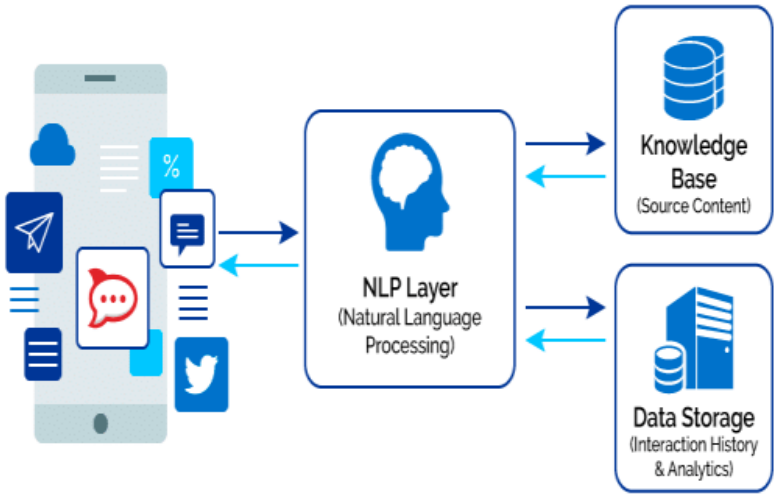


Figure 14

8 Technologies Utilisées

Dans le cadre de ce projet, j'ai employé plusieurs technologies et bibliothèques qui ont été essentielles à la mise en œuvre de mon outil de détection d'usurpation DNS. Voici un aperçu des principales technologies utilisées :

Figure 15

- **Colorama** : Cette bibliothèque a été utilisée pour améliorer la sortie en console avec des couleurs. Elle permet de rendre les messages d'alerte plus visibles, facilitant ainsi l'identification des événements critiques. Grâce à `Fore`, `Style` et `Back`, j'ai pu personnaliser le format des messages affichés.
- **Scapy** : Une bibliothèque puissante pour la manipulation de paquets réseau, Scapy m'a permis d'analyser et d'interagir avec le trafic DNS. J'ai utilisé `sniff` pour capturer les paquets en temps réel, et `wrpcap` pour enregistrer ces paquets dans un fichier au format pcap. Les couches DNS, UDP, et IP ont été essentielles pour déchiffrer et traiter les requêtes DNS.
- **Collections** : J'ai utilisé `defaultdict` de la bibliothèque `collections` pour gérer efficacement les requêtes DNS par adresse IP. Cela a permis de conserver un historique des requêtes sans avoir à initialiser manuellement les clés de dictionnaire.
- **datetime** : Cette bibliothèque a été utilisée pour gérer le temps, permettant de timestamp chaque requête DNS capturée. Elle a été cruciale pour implémenter des fonctionnalités comme le nettoyage des anciennes requêtes dans une fenêtre temporelle définie.
- **OS** : J'ai utilisé le module `os` pour interagir avec le système d'exploitation, notamment pour manipuler les règles du pare-feu à l'aide d'iptables sur les systèmes Linux. Cela m'a permis de bloquer dynamiquement les adresses IP suspectes.
- **Tabulate** : Cette bibliothèque a été utilisée pour afficher les données sous forme de tableaux dans la console, facilitant ainsi la lecture des informations sur les requêtes DNS et les alertes de sécurité.
- **dnsfiglet** et **dnsfirewall** : Ces modules, provenant de mon propre répertoire source, ont été intégrés pour fournir des fonctionnalités personnalisées. `dnsfiglet` est utilisé pour styliser les titres et les messages, tandis que `dnsfirewall` contient la classe `MaginotDNSDetector`, qui constitue le cœur de mon outil de détection.

Ces technologies ont non seulement simplifié le processus de développement, mais elles ont également renforcé l'efficacité de l'outil de détection d'usurpation DNS, permettant une analyse approfondie du trafic réseau et une réponse rapide aux menaces potentielles.

---

9 Conclusion

En conclusion, ce projet présente la conception et la mise en œuvre d'un outil modulaire pour la surveillance et l'analyse du trafic DNS, avec des fonctionnalités avancées pour la capture, l'analyse et la protection contre les attaques DNS. Grâce à l'utilisation de bibliothèques puissantes comme Scapy et Colorama, l'outil capture et filtre efficacement les paquets DNS, tout en offrant une interface utilisateur claire via la ligne de commande. La prise en charge de DNS sur HTTPS (DoH) améliore la sécurité en chiffrant les requêtes DNS, tandis que les options de filtrage et la sauvegarde des fichiers au format PCAP permettent une analyse approfondie et personnalisée du trafic. En intégrant un composant pare-feu à son outil DNSSpoofingDetector, l'auteur a ajouté une couche supplémentaire de sécurité pour détecter les tentatives d'usurpation DNS en temps réel. Avec des mécanismes de surveillance basés sur des seuils configurables et des fenêtres temporelles adaptables, l'outil est capable de réagir rapidement aux anomalies du trafic DNS et de générer des alertes détaillées pour aider les administrateurs réseau à identifier et à atténuer les attaques. Dans l'ensemble, ce projet démontre une solide maîtrise des concepts de capture de trafic réseau et de sécurité DNS, tout en offrant une solution flexible et évolutive qui peut s'adapter à différents besoins de surveillance et de protection réseau.

10 Reference

1. **Reference**

- https://ieeexplore.ieee.org/document/7726376
- https://ieeexplore.ieee.org/document/9651929
- https://ieeexplore.ieee.org/document/10590243
- https://ieeexplore.ieee.org/document/10346289
- https://ieeexplore.ieee.org/document/10531381
- https://dl.acm.org/doi/10.1145/2663716.2663731
- https://www.researchgate.net/publication/309687590_DNS_Protection_against_Spoofing_and_Poisoning_Attacks