

Royaume du Maroc  
UNIVERSITÉ MOHAMED V - RABAT  
ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE  
ET D'ANALISE DES SYSTÈMES



## Rapport de Projet de Fin de 1-ère année

# La Sécurité d'un Raspberry Pi

**Filière :** Sécurité des Systèmes d'Information (SSI)

**Présenté par :**  
AGOULZI Imane  
JOUIJATE Rim

**Sous la direction de :**  
M. BERQIA Amine

Année universitaire : 2021 - 2022



## Remerciement

---

Louange à ALLAH seul, que ses bénédictions soient sur notre seigneur et maître Mohamed et sur les siens.

Avant de commencer ce rapport,nous tenons à exprimer nos vifs remerciements et notre profonde reconnaissance à notre professeur, BERQIA Amine, qui n'a épargné aucun effort pour que ce travail prenne forme. Nous le remercions pour l'attention particulière qu'il a porté à ce travail et la confiance qu'il nous a accordé tout au long de ce parcours, ainsi que pour son soutien, ses remarques pertinentes et son encouragement.

On voudrait aussi exprimer nos remerciements les plus loyales envers tous les enseignants et le personnel de l'École Nationale Supérieure D'Informatique et D'Analyse des Systèmes (ENSIAS), ainsi que tous ceux qui ont participé à notre formation. On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.

## Résumé

---

Le sujet de notre projet de fin d'année consistait à découvrir le Raspberry Pi en tant que célèbre nano-ordinateur ; et nous avons travaillé surtout sur le fait de lier ce Raspberry pi avec des aspects de sécurité que nous avons étudié.

La combinaison du faible cout de Raspberry Pi et de sa capacité à supporter plusieurs logiciels de sécurité donne une solution possible pour surveiller un réseau local contre toute attaque. On peut atteindre cet objectif en utilisant le Raspberry Pi en tant qu'un système de détection d'intrusion.

En fait, dans ce rapport, notre travail repose sur la préparation de Raspberry Pi afin qu'il soit prêt pour être intégrer dans un réseau à surveiller.

## Abstract

---

In this project, we discovered the famous Nano-computer which is the Raspberry Pi; and we tried to link this Raspberry pi with security aspects that we have studied.

The combination of lower cost of Raspberry Pi and its ability to support multiple security software provides a possible solution to monitor a local network against any attack. This can be achieved by using the Raspberry Pi as an intrusion detection system.

Our work, in this report, is based on making Raspberry Pi ready to be integrated into a network that we want to monitor.

# Table des matières

<b>1 Généralité sur Raspberry Pi :</b>	<b>13</b>
1.1 Le Raspberry Pi : . . . . .	13
1.2 Aspect matériel : . . . . .	13
1.2.1 Les composants d'un Raspberry PI : . . . . .	13
1.2.2 Les modèles de Raspberry PI : . . . . .	14
1.3 Software . . . . .	14
1.4 Les domaines d'utilisation d'un Raspberry Pi : . . . . .	15
<b>2 La sécurité et le Raspberry Pi :</b>	<b>16</b>
2.1 La sécurité : . . . . .	16
2.2 La sécurité d'un Raspberry Pi : . . . . .	16
2.3 Intrusion Detection System (IDS) : . . . . .	17
2.3.1 Définition : . . . . .	17
2.3.2 Classification des IDS : . . . . .	18
2.4 IPS et IDS : . . . . .	19
2.4.1 Intrusion Prevention System (IPS) : . . . . .	19
2.4.2 IPS VS IDS : . . . . .	19
<b>3 Configuration et prise en main de Raspberry Pi</b>	<b>20</b>
3.1 Matériels utilisés : . . . . .	20
3.2 Installation de Raspberry Pi OS : . . . . .	21
3.3 Configuration de Raspberry Pi : . . . . .	23
<b>4 Suricata : Logiciel de détection d'intrusion (IDS)</b>	<b>26</b>
4.1 Définition : . . . . .	26
4.2 Fonctionnalités : . . . . .	26
4.3 Les règles de Suricata : . . . . .	27

4.4	Installation du Suricata : . . . . .	27
4.5	Configuration du Suricata : . . . . .	32
4.6	Lancement du Suricata : . . . . .	33
4.7	Test du Suricata : . . . . .	33
4.8	Utilisation du Suricata en mode service : . . . . .	34
4.9	Utilisation des ressources du Raspberry Pi : . . . . .	34

# Table des figures

1.1	L'architecture de Raspberry Pi 4 . . . . .	14
3.1	Raspberry Pi 4 modele B . . . . .	20
3.2	Le matériels utilisés . . . . .	21
3.3	Le site officiel du Raspberry Pi . . . . .	21
3.4	Raspberry Pi Imager . . . . .	22
3.5	Caption . . . . .	22
3.6	configuration 1 . . . . .	23
3.7	configuration 2 . . . . .	23
3.8	Ping de Raspberry Pi par la terminale de Windows . . . . .	23
3.9	Connection via SSH . . . . .	24
3.10	Recherche des mises à jour . . . . .	24
3.11	Recherche des mises à jour . . . . .	25
3.12	Créer un nouveau utilisateur . . . . .	25
4.1	Suricata Logo . . . . .	26
4.2	exepmle d'une règle de Suricata . . . . .	27
4.3	Installation des packages de suricata . . . . .	28
4.4	Instalation des dépendances nécessaires . . . . .	28
4.5	Instalation des sources de Suricata . . . . .	29
4.6	Suricata-6.0.4.tar.gz . . . . .	29
4.7	Décompression des sources . . . . .	29
4.8	Configuration du logiciel . . . . .	30
4.9	Compilation Suricata . . . . .	30
4.10	Installation Suricata . . . . .	31
4.11	Compilation suricata-update . . . . .	31
4.12	Installation suricata-update . . . . .	31

4.13	Installation des règles de Suricata . . . . .	32
4.14	Mis à jour des règles de suricata . . . . .	32
4.15	Accès au fichier suricata/yaml . . . . .	32
4.16	suricata.yaml . . . . .	33
4.17	Lancement de suricata . . . . .	33
4.18	resultat de test du suricata . . . . .	34
4.19	Contenu de fichier suricata.service . . . . .	34
4.20	activation de suricata.service . . . . .	34
4.21	Consommation du CPU . . . . .	35
4.22	Consommation du mémoire . . . . .	35
4.23	Consommation du carte SD . . . . .	35

# Liste des tableaux

1.1	Les modèles de Raspberry PI . . . . .	14
1.2	Les OS utilisées . . . . .	15
2.1	Les modèles de Raspberry PI . . . . .	17
2.2	Les modèles de Raspberry PI . . . . .	17

## Liste des abréviations

---

- 1 AIDS Anomaly-based Intrusion Detection System
- 2 CPU Central Process Unit
- 3 GPU Graphic Process Unit
- 4 HIDS Host-based Intrusion Detection System
- 5 IDS Intrusion Detection System
- 6 IPS Intrusion Prevention System
- 7 JSON JavaScript Object Notation
- 8 NIDS Network Intrusion Detection System
- 9 OS Operating System
- 10 RAM Random Access Memory
- 11 ROM Read Only Memory
- 12 RPi Raspberry Pi
- 13 SIDS Signature-based Intrusion Detection System
- 14 SoC System on a Chip
- 15 SSH Secure Shell

## Introduction Générale

---

Dans la vie quotidienne, les gens utilisent des objets connectés à internet pour de nombreuses applications : santé, loisirs, sécurité, sport, domotique...

Un objet connecté est un appareil sans fil capable de se connecter à un internet via un réseau Wi-Fi, la 4G ou le Bluetooth. Selon sa fonction, il peut stocker, recevoir, traiter et transmettre des données.

Prenant l'exemple de nos maisons où on trouve plusieurs appareils connectés au réseau domestique. Ces appareils stockent numériquement nos données privées telles que des photos, des vidéos et des informations financières. Et par conséquent, si la sécurité de notre réseau n'est pas prise en considération, nos données seront exposées à des utilisateurs malveillants qui cherchent à détruire ou à exploiter notre style de vie numérique.

Afin de pouvoir protéger efficacement un réseau contre ces activités malveillantes, On doit savoir que le réseau est attaqué, donc un système de détection d'intrusion (IDS) peut faire face. Souvent, l'IDS n'est pas conçu pour être utilisé dans les réseaux domestiques puisqu'il est plus professionnel et difficile à implémenter par les utilisateurs à domicile en tant que solution de sécurité ainsi qu'il peut être trop coûteux.

Il existe une solution à petite échelle avec un faible coût de mise en œuvre et de maintenance, qui consiste à exécuter un logiciel de détection d'intrusion, par exemple Suricata, sur un Raspberry Pi. Ce dernier est un appareil abordable et flexible qui peut jouer un grand rôle dans la protection de réseau.

Dans le cadre de nombreux projets proposés par l'ENSIAS ; afin de permettre aux étudiants de développer leurs connaissances, le projet de fin d'année est l'occasion idéale de mettre en pratique tout ce qui a été assimilé pendant l'année ou de découvrir de nouvelles notions.

Le présent document est articulé en 4 chapitres. Le premier consiste à découvrir le Raspberry Pi et ses caractéristiques. Le deuxième lie les notions de sécurité avec le Raspberry Pi. Le troisième montre la configuration et la pris en main de Raspberry Pi. Et le quatrième représente l'installation de l'IDS Suricata au sein de Raspberry pi.

# **Chapitre 1**

## **Généralité sur Raspberry Pi :**

### **1.1 Le Raspberry Pi :**

Partout dans le monde, les gens utilisent le Raspberry Pi pour acquérir des compétences en programmation, créer des projets matériels, faire de la domotique et les utiliser dans des applications industrielles.

Raspberry Pi est le nom d'une série d'ordinateurs à carte unique lancée en 2012 par la Raspberry Pi Foundation, une organisation caritative britannique qui encourage l'apprentissage, l'expérimentation et l'innovation pour les élèves.

Rasspberry Pi peut se brancher sur un écran d'ordinateur ou un téléviseur, et utilise un clavier et une souris standard. C'est un petit appareil qui permet aux personnes de tous âges d'explorer l'informatique et d'apprendre à programmer dans des langages comme Scratch et Python. Comme un ordinateur bureau, elle est capable de naviguer sur Internet et lire des vidéos HD en passant par la création des feuilles de calcul, le traitement de texte et les jeux.

### **1.2 Aspect matériel :**

#### **1.2.1 Les composants d'un Raspberry PI :**

Le cœur d'une carte Raspberry est constitué d'un circuit intégré appelé SoC (System on a Chip : tout le système sur une puce). Il regroupe les composants essentiels de tout ordinateur, c'est-à-dire le CPU (Central Process Unit : processeur central), la mémoire RAM (Random Access Memory : mémoire à accès aléatoire, communément appelée mémoire vive), le GPU (Graphic Process Unit : processeur graphique), et un port USB.

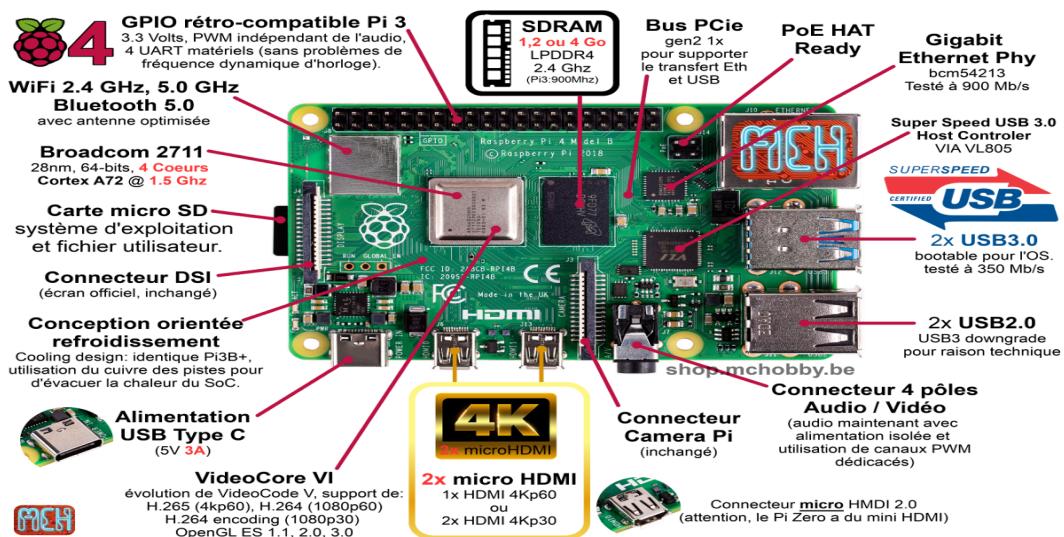


FIGURE 1.1 – L'architecture de Raspberry Pi 4

### 1.2.2 Les modèles de Raspberry PI :

Raspberry Pi	Date de sortie	USB Ports	CPU	RAM	BT	WiFi
Raspberry Pi 1 B	févr-12	2x USB 2.0	700MHZ	512MB	Non	Non
Raspberry Pi 1 B+	juil-14	4x USB 2.0	700MHz	512MB	Non	Non
Raspberry Pi 1 A+	nov-14	1x USB	700MHz	512MB	Non	Non
Raspberry Pi 2	févr-15	4x USB	900MHz	1GB	Non	Non
Raspberry Pi Zero	nov-15	1x Micro USB	1GHz	512MB	Non	Non
Raspberry Pi 3	févr-16	4x USB	1.2GHz	1GB	4.1 LE	Oui
Raspberry Pi Zero W	févr-17	1x Micro USB	1GHz	512MB	4.1	Oui
Raspberry Pi 3 B+	mars-18	4x USB 2.0	1.4GHz	1GB	4.2,BLE	Oui
Raspberry Pi 3 A+	nov-2018	1x USB 2.0	1.4GHz	512MB	4.2,BLE	Oui
Raspberry Pi 4	juin-19	2X USB 3.0, 2X USB 2.0	1.5GHz	1G, 2GB, 4GB or 8GB	5.0, BLE	Oui
Raspberry Pi 400	nov-20	2X USB 3.0, 2X USB 2.0	1.8GHz	4GB	5.0,BLE	Oui
Raspberry Pi Pico	janv-21	USB C	133MHz	264KB	Non	Non

TABLE 1.1 – Les modèles de Raspberry PI

### 1.3 Software

La Raspberry Pi n'est pas livré avec un système d'exploitation préinstallé. Cela signifie que vous pouvez choisir parmi une large sélection de systèmes d'exploitation (OS). N'importe lequel d'entre eux peut être flashé sur la carte SD de votre Raspberry Pi.

De nombreux systèmes d'exploitation sont disponibles pour Raspberry Pi, y compris Raspberry Pi OS, le système d'exploitation officiel de la Raspberry Pi Foundation pris en charge, et les systèmes d'exploitation d'autres organisations.

	Année de publication	Développeur	Éditeur	Basé sur	Caractéristiques distinctives
Arch Linux ARM	2010	Arch Linux Project	Arch Linux		En cycle Rolling-Release (développement continu)
Free BSD	1993	FreeBSD Projekt	BSD		Capacités de réseau et de stockage de premier ordre
Kali Linux	2013	Offensive Security	Debian		Divers outils pour des contrôles de sécurité performants
Pidora	2014	CDOT	Fedora		Mode Headless
Rasp - bian	2012	Mike Thompson, Peter Green	Debian		Système d'exploitation standard officiel de Raspberry Pi
Windows 10 IoT Core	2015	Microsoft	Windows 10		Propriétaire (mais gratuit)

TABLE 1.2 – Les OS utilisées

## 1.4 Les domaines d'utilisation d'un Raspberry Pi :

Les domaines d'intervention de Raspberry Pi sont très vastes. En plus des nombreuses options pratiques qui s'offrent à vous, il est également possible de mettre en place toutes sortes de projets qui sortent de l'ordinaire. Le degré de connaissances préalable pour mener à bien ces projets varie grandement. Avec un peu de patience et de curiosité, il est possible d'arriver à de très bons résultats, en expérimentant avec la carte mère et en acquérant de nouvelles connaissances dans le domaine de l'informatique.

### A) Utiliser Raspberry Pi comme un ordinateur :

Le premier usage possible de la Raspberry Pi est simplement celui d'ordinateur.

Si vous avez besoin d'un ordinateur de petite taille, pas cher, pour lire vos mails et écrire vos documents, ou encore pour coder ou lire des vidéos, alors la Raspberry Pi est largement suffisante.

### B) Utiliser Raspberry Pi comme un média-centre :

Le second usage de la Raspberry, c'est comme serveur multi-média, et plus particulièrement en tant que média-center.

En effet, elle présente une puissance largement suffisante pour un tel usage, et sa taille réduite (plus réduite en fait que la majeure partie des média-centres commerciaux) lui permettra de facilement s'intégrer de façon discrète et élégante n'importe où.

### C) Utiliser Raspberry Pi comme console de rétro-gaming :

Autre usage possible de la Raspberry Pi, la création d'une console de rétro-gaming qui vous permette de jouer à de vieux jeux.

## Chapitre 2

# La sécurité et le Raspberry Pi :

### 2.1 La sécurité :

Etymologie : du latin *securitas*, absence de soucis, tranquillité de l'âme, dérivé de *securus*, exempt de soucis, exempt de crainte, tranquille.

La sécurité est l'absence de danger, c'est-à-dire une situation dans laquelle quelqu'un ou quelque chose n'est pas exposé à des évènements critiques ou à des risques.

En informatique, La sécurité protège l'intégrité des technologies de l'information comme les systèmes, les réseaux et les données informatiques contre les attaques, les dommages ou les accès non autorisés. Elle vise généralement cinq principaux objectifs :

- **L'intégrité** : garantir que les données sont bien celles que l'on croit être.
- **La disponibilité** : maintenir le bon fonctionnement du système d'information.
- **La confidentialité** : rendre l'information inintelligible à d'autres personnes que les seuls acteurs d'une transaction.
- **La non répudiation** : garantir qu'une transaction ne peut être niée.
- **L'authentification** : assurer que seules les personnes autorisées aient accès aux ressources.

### 2.2 La sécurité d'un Raspberry Pi :

Un système exposé à Internet est toujours un risque pour son environnement et pour lui-même. Tout appareil connecté est vulnérable. Si un seul appareil de son environnement est facilement piratable, il sera plus facile pour un attaquant de prendre le contrôle de plus d'appareils. Les utilisateurs doivent donc comprendre le risque de ne pas consacrer de temps à la sécurisation d'un système.

Le Raspberry Pi est un ordinateur complet à petit prix, ce qui implique quelques concessions de conception et de fabrication, alors cela conduit à certaines limitations et faiblesses qui doivent être prises en compte lors de son utilisation.

Une configuration par défaut peut laisser le système dans un état vulnérable, ce qui peut être exploité par des tiers malveillants. Donc il faut être conscient des vulnérabilités de cet appareil pour éviter ou minimiser les risques d'attaques.

Les tableaux ci-dessous représentent différentes vulnérabilités matérielles et logicielles trouvées dans un Raspberry Pi lors de l'utilisation d'une installation par défaut de différents systèmes d'exploitation disponibles :

Vulnerability	Description
USB power and backfeeding	-Powered USB hub required for USB devices that consumes more than 500 mA -Raspberry Pi can be powered by its USB due to the lack of USB protections
Overclocking	-Bad configuration can burn the device
GPIO logic levels and serial access	-GPIO uses 3.3V logic. 5V logic may destroy the SoC
Real time clock absence	-Raspberry Pi cannot keep the time after being powered off
Xenon flash shyness	-RPi 2 reboots if it is pointed with a laser or a xenon flash

TABLE 2.1 – Les modèles de Raspberry PI

Operating System	vulnerability
Raspbian	-Default user and password
Windows 10	-Unsecude Windows Device Portal
OpenELEC LiberELEC	-Default user and password(and cannot be charged) -HTTP Samba unsecured services
Ubuntu	-Nothing to report
RiscOS	-Nothing to report

TABLE 2.2 – Les modèles de Raspberry PI

## 2.3 Intrusion Detection System (IDS) :

### 2.3.1 Définition :

Un système de détection d'intrusion (IDS) est une solution logicielle qui surveille un système ou le réseau pour les intrusions, les violations de politique ou les activités malveillantes. Et lorsqu'il détecte une intrusion ou une violation, le logiciel le signale à l'administrateur ou au personnel de sécurité. Il les aide à enquêter sur l'incident signalé et à prendre les mesures appropriées.

Cette solution de surveillance passive peut vous alerter de la détection d'une menace, mais elle ne peut pas prendre de mesures directes contre celle-ci.

Un système IDS vise à détecter une menace avant qu'elle ne s'infiltre dans un réseau. Il nous donne le pouvoir de jeter un œil à notre réseau sans obstruer le flux de trafic réseau. En plus de détecter les violations de politique, il peut se prémunir contre les menaces telles que les fuites d'informations, les accès non autorisés, les erreurs de configuration, les chevaux de Troie et les virus.

### 2.3.2 Classification des IDS :

L'IDS est divisé en fonction de l'endroit où se produit la détection de la menace ou de la méthode de détection utilisée.

#### 1. Selon la cible surveillée :

##### – **Systèmes de détection d'intrusion dans le réseau (NIDS) :**

Le NIDS fait partie de l'infrastructure réseau et surveille les paquets qui la traversent. Il coexiste avec les appareils avec une capacité de prise, de répartition ou de mise en miroir comme les commutateurs. NIDS est positionné à un ou plusieurs points stratégiques d'un réseau pour surveiller le trafic entrant et sortant de tous les appareils connectés.

Il analyse le trafic traversant l'ensemble du sous-réseau, en faisant correspondre le trafic passant les sous-réseaux à la bibliothèque d'attaque connue. Une fois que NIDS a identifié les attaques et détecté un comportement abnormal, il alerte l'administrateur du réseau.

##### – **Systèmes de détection d'intrusion basés sur l'hôte (HIDS) :**

Les HIDS sont la solution qui s'exécute sur des appareils ou des hôtes distincts sur un réseau. Il ne peut surveiller que les paquets de données entrants et sortants des appareils connectés et alerter l'administrateur ou les utilisateurs en cas de détection d'une activité suspecte. Il surveille les appels système, les modifications de fichiers, les journaux d'application, etc.

Le HIDS prend des instantanés des fichiers actuels dans le système et les fait correspondre aux précédents. S'il constate qu'un fichier critique est supprimé ou modifié, le HIDS envoie une alerte à l'administrateur pour enquêter sur le problème.

#### 2. Selon le principe de détection :

##### – **Système de détection d'intrusion par signatures (SIDS) :**

C'est l'approche la plus basique et la plus ancienne. Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Cette démarche appliquée à la détection d'intrusion, est très similaire à celle des outils antivirus et présente les mêmes inconvénients que celle-ci. Il est aisément de comprendre que ce type d'IDS est purement réactif; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour quotidiennes.

##### – **Système de détection d'intrusion par anomalies (AIDS) :**

La mise en œuvre de cette approche comprend toujours une phase d'apprentissage au cours de laquelle ces IDS vont découvrir le fonctionnement

normal des éléments surveillés. Une fois cet apprentissage effectué, ils signaleront les divergences par rapport au fonctionnement de référence. Ces systèmes peuvent être élaborés à partir d'analyses statistiques ou de techniques proches de l'intelligence artificielle. La principale promesse des AIDS est la détection des nouveaux type d'attaque. En effet, ils ne sont pas programmés pour reconnaître des attaques spécifiques mais signalent toute activité anormale.

## 2.4 IPS et IDS :

### 2.4.1 Intrusion Prevention System (IPS) :

D'abord, le système de prévention des intrusions (IPS) qui est également appelé système de détection et de prévention des intrusions (IDPS), est une solution logicielle qui surveille les activités d'un système ou d'un réseau à la recherche d'incidents malveillants, enregistre des informations sur ces activités, les signale à l'administrateur ou au personnel de sécurité et tente de les arrêter ou de les bloquer.

### 2.4.2 IPS VS IDS :

La principale différence entre les deux tient au fait que l'IDS est un système de surveillance, alors que l'IPS est un système de contrôle.

Avec l'IDS, il est nécessaire qu'un humain ou un autre système prenne ensuite le relais pour examiner les résultats et déterminer les actions à mettre en œuvre, ce qui peut représenter un travail à temps complet selon la quantité quotidienne de trafic généré.

Pour sa part, l'objectif de l'IPS est de capturer les paquets dangereux et de les retirer avant qu'ils n'atteignent leur cible. Il est plus passif qu'un IDS et exige simplement de mettre régulièrement à jour la base de données pour y intégrer les informations relatives aux nouvelles menaces.

Nous avons choisi de travailler avec suricata

## Chapitre 3

# Configuration et prise en main de Raspberry Pi

Raspberry Pi est livré avec une sécurité médiocre par défaut. Si vous l'utilisez à la maison ou dans un petit réseau, ce n'est pas grave, mais si vous ouvrez des ports sur Internet, utilisez-le comme point d'accès Wi-Fi, ou si vous l'installez sur un réseau plus large, vous devez prendre des mesures de sécurité pour protéger votre Raspberry Pi.

### 3.1 Matériels utilisés :

- Raspberry Pi 4 Model B 8 GB RAM

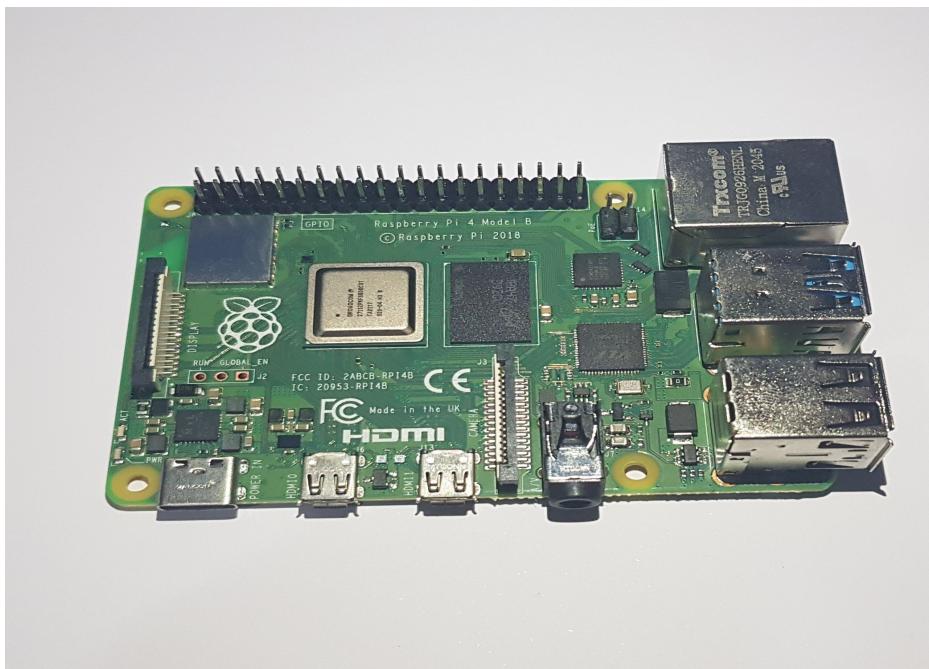


FIGURE 3.1 – Raspberry Pi 4 modele B

- la carte Micro SD 32Go
- Cable HDMI

- Chargeur type C



FIGURE 3.2 – Le matériels utilisés

### 3.2 Installation de Raspberry Pi OS :

On a choisi le Raspberry Pi OS comme système d'exploitation de notre Raspberry Pi 4 Model B 8 GB RAM.

Sur le site officiel, on a téléchargé la dernière image de Raspberry Pi OS ainsi que le logiciel Raspberry Pi Imager v1.7.2.

1. Téléchargement de Raspberry Pi imager trouvé dans le site officiel de raspberry pi

We use cookies to ensure that we give you the best experience on our websites. By continuing to visit this site you agree to our use of cookies. [Read our cookie policy.](#)

**Raspberry Pi OS**

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.

FIGURE 3.3 – Le site officiel du Raspberry Pi

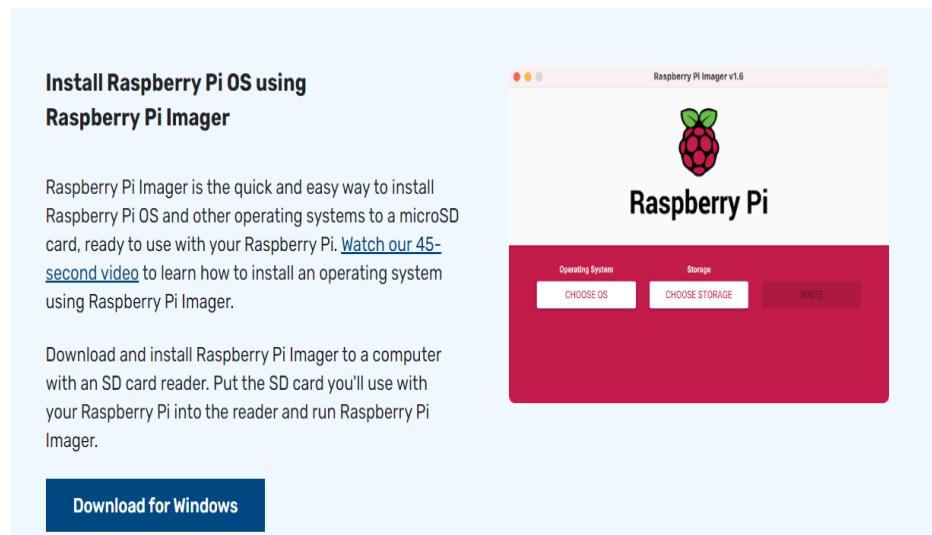


FIGURE 3.4 – Raspberry Pi Imager

2. Préparation de la carte Micro SD : on a utilisé une catre de taille 32Go formatée.
3. Démarrage de Raspberry Pi Imager v1.7.2.

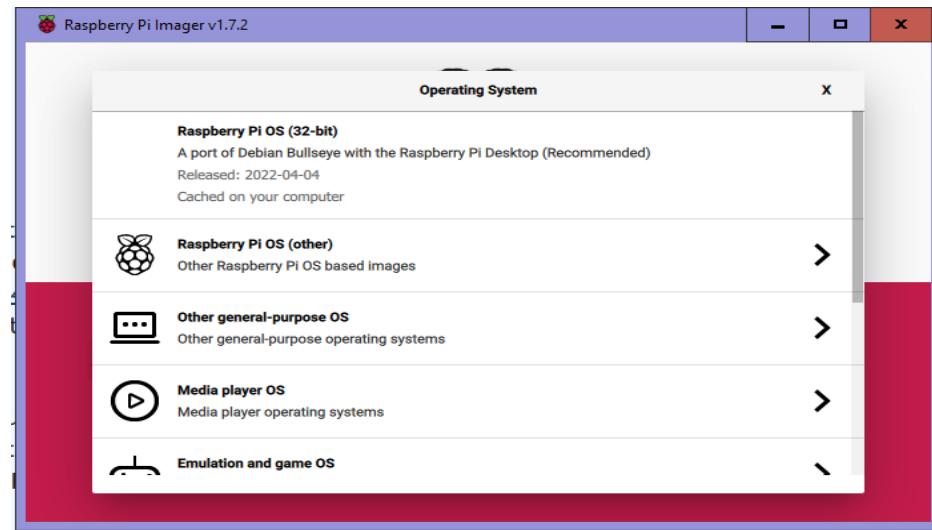


FIGURE 3.5 – Caption

4. Configuration de Raspberry pi : Notre but est d'utiliser le SSH pour connecter le Raspberry Pi via la terminale de Windows.

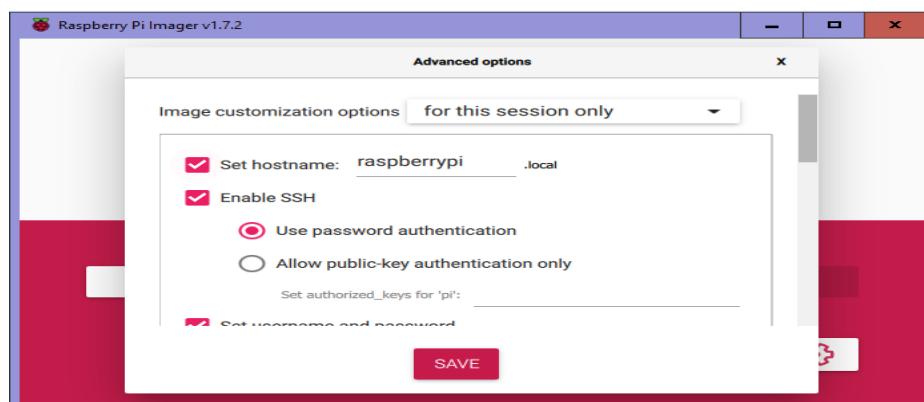


FIGURE 3.6 – configuration 1

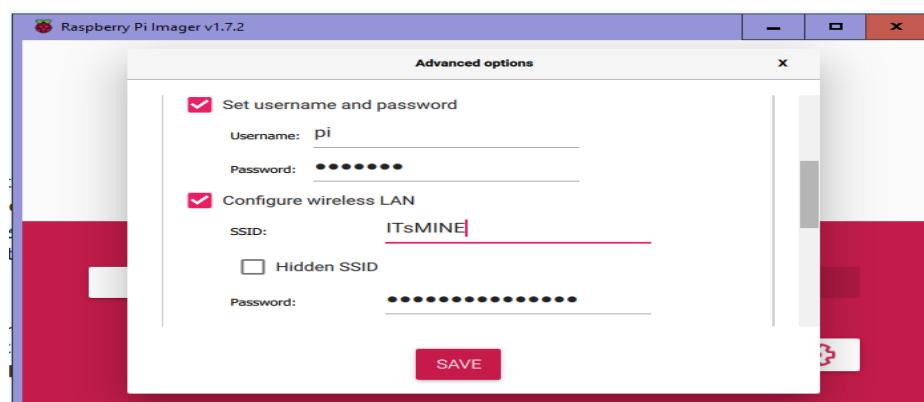


FIGURE 3.7 – configuration 2

### 3.3 Configuration de Raspberry Pi :

1. On insère la carte SD dans le Raspberry Pi, puis On le démarre afin de valider les paramètres du système pour son premier démarrage.
2. On attend quelques secondes avant de faire le ping de Raspberry Pi.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ping raspberrypi.local

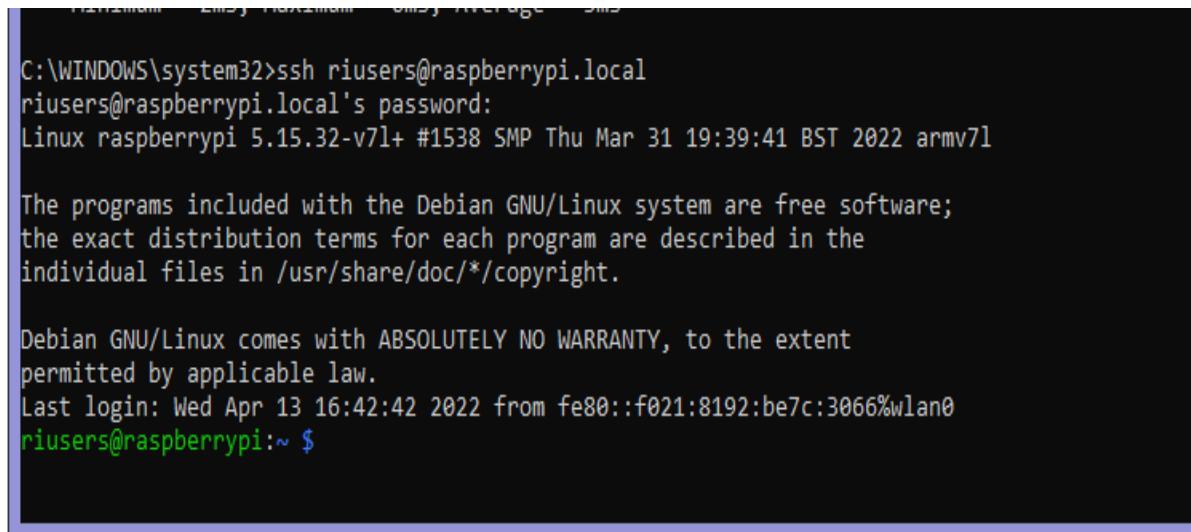
Pinging raspberrypi.local [fe80::c65:e1c1:728a:33c9%19] with 32 bytes of data:
Reply from fe80::c65:e1c1:728a:33c9%19: time=2ms
Reply from fe80::c65:e1c1:728a:33c9%19: time=3ms
Reply from fe80::c65:e1c1:728a:33c9%19: time=6ms
Reply from fe80::c65:e1c1:728a:33c9%19: time=2ms

Ping statistics for fe80::c65:e1c1:728a:33c9%19:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 3ms

C:\WINDOWS\system32>
```

FIGURE 3.8 – Ping de Raspberry Pi par la terminale de Windows

3. Après la réponse de Raspberry Pi, on va donc se connecter via SSH.



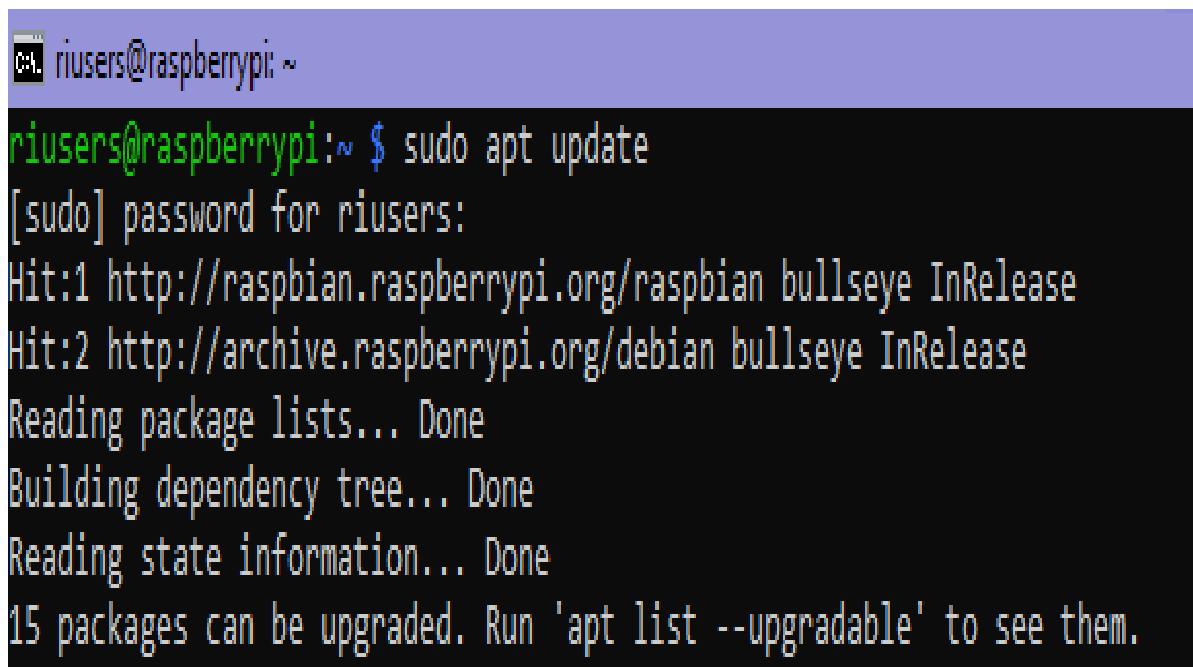
```
C:\WINDOWS\system32>ssh riusers@raspberrypi.local
riusers@raspberrypi.local's password:
Linux raspberrypi 5.15.32-v7l+ #1538 SMP Thu Mar 31 19:39:41 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 13 16:42:42 2022 from fe80::f021:8192:be7c:3066%wlan0
riusers@raspberrypi:~ $
```

FIGURE 3.9 – Connection via SSH

4. On cherche des mises à jour à faire pour notre Raspberry Pi OS.



```
riusers@raspberrypi:~ $ sudo apt update
[sudo] password for riusers:
Hit:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Hit:2 http://archive.raspberrypi.org/debian bullseye InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
15 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

FIGURE 3.10 – Recherche des mises à jour

5. On installe tous les mises à jour disponibles.

```

pi@raspberrypi:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  gzip libcamera-tools liblzma5 libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0 pi-bluetooth piwiz raspi-config realvnc-vnc-server
  userconf-pi xcompngr xz-utils
15 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 10.2 MB of archives.
After this operation, 6,129 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian bullseye/main armhf libcamera0 armhf 0~git20220426+18e68a9b-1 [548 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye/main armhf libcamera-tools armhf 0~git20220426+18e68a9b-1 [75.4 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian.bullseye/main armhf gzip armhf 1.10-4+deb11u1 [124 kB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi-dev armhf 1:2+git20220324~090146+c4fd1b8-1 [177 kB]
Get:5 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi-bin armhf 1:2+git20220324~090146+c4fd1b8-1 [149 kB]
Get:6 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi0 armhf 1:2+git20220324~090146+c4fd1b8-1 [167 kB]
Get:7 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi-doc all 1:2+git20220324~090146+c4fd1b8-1 [1,908 kB]
Get:8 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi0 armhf 1:2+git20220324~090146+c4fd1b8-1 [167 kB]
Get:9 http://archive.raspberrypi.org/debian bullseye/main armhf libraspberrypi-doc all 1:2+git20220324~090146+c4fd1b8-1 [1,908 kB]
Get:10 http://archive.raspberrypi.org/debian bullseye/main armhf pi-bluetooth all 0.1.19 [5,764 kB]
Get:11 http://archive.raspberrypi.org/debian bullseye/main armhf raspi-config all 20220425 [30.3 kB]
Get:12 http://archive.raspberrypi.org/debian bullseye/main armhf userconf-pi all 0.2 [5,736 kB]
Get:13 http://archive.raspberrypi.org/debian bullseye/main armhf piwiz armhf 0.48 [167 kB]
Get:14 http://archive.raspberrypi.org/debian bullseye/main armhf realvnc-vnc-server armhf 6.9.1.46706 [8,343 kB]
Get:15 http://raspbian.raspberrypi.org/raspbian.bullseye/main armhf liblzma5 armhf 5.2.5-2.1~deb11u1 [159 kB]
Get:16 http://raspbian.raspberrypi.org/raspbian.bullseye/main armhf xz-utils armhf 5.2.5-2.1~deb11u1 [216 kB]
Get:17 http://archive.raspberrypi.org/debian bullseye/main armhf xcompngr armhf 1.1.8-1+rpt1 [25.6 kB]
Fetched 10.2 MB in 1min 23s (123 kB/s)
Reading changelogs... Done
(Reading database ... 101501 files and directories currently installed.)
Preparing to unpack .../gzip_1.10-4+deb11u1_armhf.deb ...
Unpacking gzip (1.10-4+deb11u1) over (1.10-4) ...
Setting up gzip (1.10-4+deb11u1) ...
(Reading database ... 101501 files and directories currently installed.)
Preparing to unpack .../liblzma5_5.2.5-2.1~deb11u1_armhf.deb ...
Unpacking liblzma5:armhf (5.2.5-2.1~deb11u1) over (5.2.5-2) ...
Setting up liblzma5:armhf (5.2.5-2.1~deb11u1) ...
(Reading database ... 101501 files and directories currently installed.)
Preparing to unpack .../00-xz-utils_5.2.5-2.1~deb11u1_armhf.deb ...
Unpacking xz-utils (5.2.5-2.1~deb11u1) over (5.2.5-2) ...

```

FIGURE 3.11 – Recherche des mises à jour

## 6. On change les paramètres par défaut.

```

pi@raspberrypi:~ $ sudo /usr/sbin/useradd --groups sudo -m riusers
pi@raspberrypi:~ $ whoiam
-bash: whoiam: command not found
pi@raspberrypi:~ $ whoami
pi
pi@raspberrypi:~ $ sudo passwd riusers
New password:
Retype new password:
passwd: password updated successfully
pi@raspberrypi:~ $

```

FIGURE 3.12 – Crée un nouveau utilisateur

## Chapitre 4

# Suricata : Logiciel de détection d'intrusion (IDS)

### 4.1 Définition :

Suricata est un logiciel open source de détection d'intrusion (IDS), de prévention d'intrusion (IPS), et de supervision de sécurité réseau (NSM). Il est développé par la fondation OISF (Open Information Security Foundation).

Suricata permet l'inspection des Paquets en Profondeur (DPI). De nombreux cas d'utilisations déontologiques peuvent être mis en place permettant notamment la remontée d'informations qualitatives et quantitatives.

Suricata analyse le trafic sur une ou plusieurs interfaces réseaux en fonction de règles activées. Il génère, par défaut, un fichier JSON. Celui-ci peut être ensuite utilisé par le logiciel de type Extract-transform-load comme par exemple logstash souvent utilisé avec Elasticsearch.



FIGURE 4.1 – Suricata Logo

### 4.2 Fonctionnalités :

Liste des principales fonctionnalités :

- IDS/IPS.

- Performances élevées : Multi-threading, utilisation des GPU (accélération graphique).
- Détection automatique de protocole (IPv4/6, TCP, UDP, ICMP, HTTP, TLS, FTP, SMB, DNS).
- NSM : journalisation DNS, module de journalisation HTTP, enregistrement des certificats et extraction de fichiers, vérification de somme de contrôle md5.
- Librairie HTP indépendante.
- Nombreux formats de sortie Unified2, JSON, Prelude.
- Écriture de scripts en Lua pour l'analyse avancée.

### 4.3 Les règles de Suricata :

Les signatures/les règles jouent un rôle majeur dans Suricata, et surtout pour déclencher les alertes.

On peut trouver un ensemble de règles déjà existantes dans Suricata-update, comme on peut modifier ou créer des nouvelles règles.

Une règle dans Suricata est constituée des éléments suivants :

- The action : détermine le comportement à adopter en cas de détection d'intrusion.
- The header : définit le protocole, les adresses IP source et destination, la direction du trafic (entrant ->, sortant <- ou bidirectionnel <>).
- The options : définit les spécificités de la règle.

Voici un exemple d'une règle :

---

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";  
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK .*USA.*[0-9]  
{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124;  
rev:2;)
```

FIGURE 4.2 – exemple d'une règle de Suricata

La partie rouge correspond à l'action, la verte au header et la bleue aux options.

### 4.4 Installation du Suricata :

Dans cette partie, on cite les étapes d'installation et de configuration de Suricata dans le Raspberry Pi 4.

## 1. La recherche et l'installation de package Suricata :

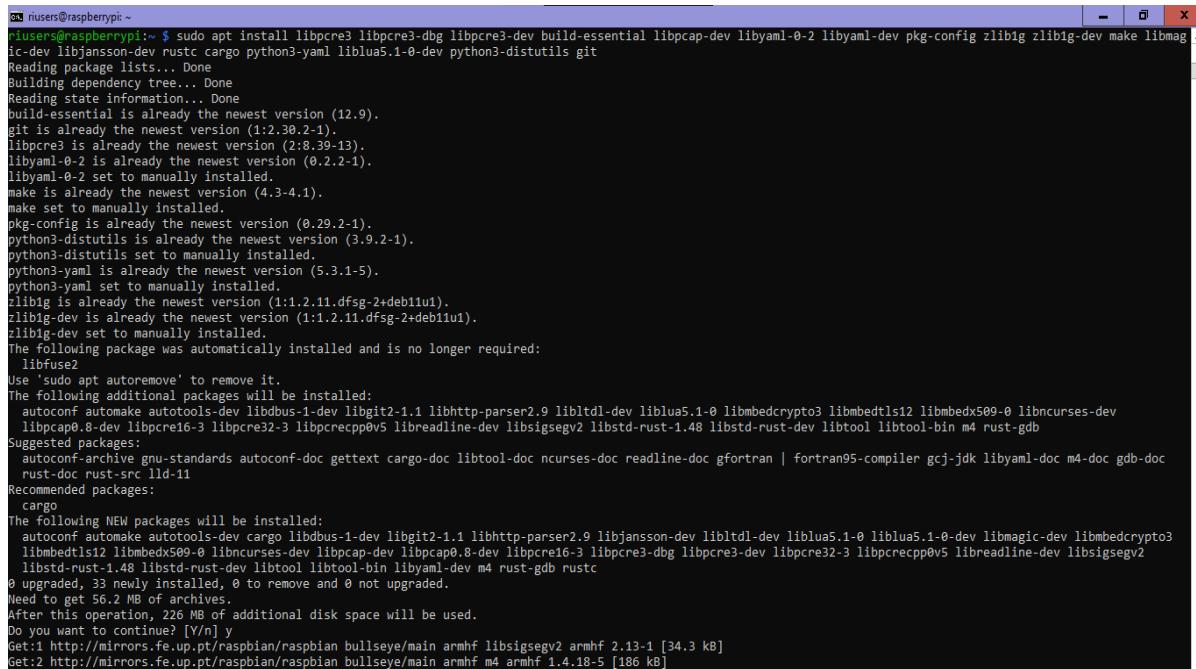
```

pi@raspberrypi:~ $ sudo apt install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfile-listing-perl libfont-afm-perl
  libhtml-form-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp2 libhttp-cookies-perl libhttp-daemon-perl
  libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl liblwp-mediatypes-perl liblwp-protocol-https-perl
  libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnet1 libnetfilter-log1 libnetfilter-queue1 libnfnetlink0 libpcap0.8 libtimedate-perl
  libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-openssl-defaults python3-yaml snort-rules-default suricata-update
Suggested packages:
  libdigest-hmac-perl libgssapi-perl libcrypt-ssleay-perl libauthen-ntlm-perl snort | snort-pgsql | snort-mysql libtcmalloc-minimal4
The following NEW packages will be installed:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfile-listing-perl libfont-afm-perl
  libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp2 libhttp-cookies-perl libhttp-daemon-perl
  libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl liblwp-mediatypes-perl liblwp-protocol-https-perl
  libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnet1 libnetfilter-log1 libnetfilter-queue1 libnfnetlink0 libpcap0.8 libtimedate-perl
  libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-openssl-defaults python3-yaml snort-rules-default suricata-update
0 upgraded, 44 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,525 kB of archives.
After this operation, 14.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libevent-core-2.1-7 armhf 2.1.12-stable-1 [126 kB]
Get:3 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libredis0.14 armhf 0.14.1-1 [30.8 kB]
Get:4 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libhttp2 armhf 1:0.5.36-4 [58.7 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libnet1 armhf 1.1.6+dfsg-3.1 [53.9 kB]
Get:6 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libnftnlink0 armhf 1.0.1-3+b1 [12.4 kB]
Get:7 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libnetfilter-log1 armhf 1.0.1-3 [9,944 B]
Get:2 http://mirror.as4328.net/raspbian/raspbian bullseye/main armhf libevent-pthreads-2.1-7 armhf 2.1.12-stable-1 [56.8 kB]
Get:8 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libnetfilter-queue1 armhf 1.0.5-2 [12.1 kB]
Get:9 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libpcap0.8 armhf 1.10.0-2 [143 kB]

```

FIGURE 4.3 – Installation des packages de suricata

## 2. L'installation des dépendances nécessaires :



```

pi@raspberrypi:~ $ sudo apt install libpcre3 libpcre3-dbg libpcre3-dev build-essential libpcap-dev libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev make libmag ...
ic-dev libjansson-dev rustc cargo python3-yaml liblua5.1-0-dev python3-distutils git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9).
git is already the newest version (1:2.30.2-1).
libpcre3 is already the newest version (2:8.39-13).
libyaml-0-2 is already the newest version (0.2.2-1).
libyaml-0-2 set to manually installed.
make is already the newest version (4.3-4.1).
make set to manually installed.
pkg-config is already the newest version (0.29.2-1).
python3-distutils is already the newest version (3.9.2-1).
python3-distutils set to manually installed.
python3-yaml is already the newest version (5.3.1-5).
python3-yaml set to manually installed.
zlib1g is already the newest version (1:1.2.11.dfsg-2+deb11u1).
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2+deb11u1).
zlib1g-dev set to manually installed.
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  autoconf automake autotools-dev libldbus-1-dev libgit2-1.1 libhttp-parser2.9 libltdl-dev liblua5.1-0 libmbcrypto3 libmbdts12 libmbdex509-0 libncurses-dev
  libpcap0.8-dev libpcre16-3 libpcre32-3 libpcrecpp0v5 libreadline-dev libsigsegv2 libstdc++-rust-1.48 libstd-rust-dev libtool libtool-bin m4 rust-gdb
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc gettext cargo-doc libtool-doc ncurses-doc readline-doc gfortran | fortran95-compiler gcj-jdk libyaml-doc m4-doc gdb-doc
  rust-doc rust-src lld-11
Recommended packages:
  cargo
The following NEW packages will be installed:
  autoconf automake autotools-dev cargo libldbus-1-dev libgit2-1.1 libhttp-parser2.9 libjansson-dev libltdl-dev liblua5.1-0 liblua5.1-0-dev libmagic-dev libmbcrypto3
  libmbdts12 libmbdex509-0 libncurses-dev libpcap-dev libpcap0.8-dev libpcre16-3 libpcre3-dbg libpcre3-dev libpcre32-3 libpcrecpp0v5 libreadline-dev libsigsegv2
  libstdc++-rust-1.48 libstdc++-rust-dev libtool libtool-bin libyaml-dev m4 rust-gdb rustc
0 upgraded, 33 newly installed, 0 to remove and 0 not upgraded.
Need to get 56.2 MB of archives.
After this operation, 226 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf libsigsegv2 armhf 2.13-1 [34.3 kB]
Get:2 http://mirrors.fe.up.pt/raspbian/raspbian bullseye/main armhf m4 armhf 1.4.18-5 [186 kB]

```

FIGURE 4.4 – Instalation des dépendances nécessaires

## 3. Téléchargement des sources de Suricata :

```

riusers@raspberrypi:~ $ wget https://www.openinfosecfoundation.org/download/suricata-6.0.4.tar.gz
--2022-04-26 23:33:53-- https://www.openinfosecfoundation.org/download/suricata-6.0.4.tar.gz
Resolving www.openinfosecfoundation.org (www.openinfosecfoundation.org)... 52.14.249.179, 2600:1f16:db2:4f00:da9d:37d6:e8b9:9802
Connecting to www.openinfosecfoundation.org (www.openinfosecfoundation.org)|52.14.249.179|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32498036 (31M) [application/x-gzip]
Saving to: 'suricata-6.0.4.tar.gz'

suricata-6.0.4.tar.gz          100%[=====] 30.99M  130KB/s   in 4m 28s

2022-04-26 23:38:22 (119 KB/s) - 'suricata-6.0.4.tar.gz' saved [32498036/32498036]

```

FIGURE 4.5 – Instalation des sources de Suricata

4. Ici on trouve le fichier compressé :

```

riusers@raspberrypi:~ $ ls
suricata-6.0.4.tar.gz

```

FIGURE 4.6 – Suricata-6.0.4.tar.gz

Maintenant, on décomprime le fichier des sources en utilisant la commande : **tar** :

```

riusers@raspberrypi:~ $ tar -xvf suricata-6.0.4.tar.gz
suricata-6.0.4/
suricata-6.0.4/config.sub
suricata-6.0.4/acsite.m4
suricata-6.0.4/configure
suricata-6.0.4/rules/
suricata-6.0.4/rules/tls-events.rules
suricata-6.0.4/rules/dns-events.rules
suricata-6.0.4/rules/http2-events.rules
suricata-6.0.4/rules/dnp3-events.rules
suricata-6.0.4/rules/decoder-events.rules
suricata-6.0.4/rules/ipsec-events.rules
suricata-6.0.4/rules/Makefile.in
suricata-6.0.4/rules/mqtt-events.rules
suricata-6.0.4/rules/modbus-events.rules
suricata-6.0.4/rules/ntp-events.rules
suricata-6.0.4/rules/smtp-events.rules
suricata-6.0.4/rules/http-events.rules
suricata-6.0.4/rules/smb-events.rules
suricata-6.0.4/rules/files.rules
suricata-6.0.4/rules/stream-events.rules
suricata-6.0.4/rules/dhcp-events.rules
suricata-6.0.4/rules/nfs-events.rules
suricata-6.0.4/rules/app-layer-events.rules
suricata-6.0.4/rules/Makefile.am
suricata-6.0.4/rules/kerberos-events.rules
suricata-6.0.4/Changelog
suricata-6.0.4/Makefile.in
suricata-6.0.4/rust/
suricata-6.0.4/rust/dist/
suricata-6.0.4/rust/dist/rust-bindings.h
suricata-6.0.4/rust/Makefile.in
suricata-6.0.4/rust/cbindgen.toml
suricata-6.0.4/rust/Cargo.toml.in
suricata-6.0.4/rust/src/
suricata-6.0.4/rust/src/lua.rs
suricata-6.0.4/rust/src/filecontainer.rs

```

FIGURE 4.7 – Décompression des sources

5. Après on se place dans le dossier Suricata pour configurer l'installation du logiciel :

```
riusers@raspberrypi:~/suricata-6.0.4
riusers@raspberrypi:~/suricata-6.0.4 $ ./configure --prefix=/usr/ --sysconfdir=/etc/ --localstatedir=/var/
checking whether make supports nested variables... yes
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk...
checking whether make sets $(MAKE)... yes
checking whether UID '1001' is supported by ustar format... yes
checking whether GID '1001' is supported by ustar format... yes
checking how to create a ustar tar archive... gntar
checking build system type... armv7l-unknown-linux-gnueabihf
checking host system type... armv7l-unknown-linux-gnueabihf
checking how to print strings... printf
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking for gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for a sed that does not truncate output... /usr/bin/sed
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for fgrep... /usr/bin/grep -F
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD- or MS-compatible name lister (nm)... /usr/bin/nm -B
checking the name lister (/usr/bin/nm -B) interface... BSD nm
```

FIGURE 4.8 – Configuration du logiciel

## 6. La compilation du Suricata :

```
riusers@raspberrypi:~/suricata-6.0.1 $ make
Making all in libhttp
make[1]: Entering directory '/home/riusers/suricata-6.0.1/libhttp'
make  all-recurse
make[2]: Entering directory '/home/riusers/suricata-6.0.1/libhttp'
Making all in http
make[3]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
Making all in lzma
make[4]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
/bin/bash ../../libtool  --tag=CC   --mode=compile gcc -DHAVE_CONFIG_H -I. -I../../ -O2 -I
../../ -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzFind.lo -MD -MP -MF .deps/LzFind.Tpo -c -o LzFind.lo LzFind.c
libtool: compile:  gcc -DHAVE_CONFIG_H -I. -I../../ -O2 -I../../ -D_GNU_SOURCE -g -Wall -Wextra -std=gnu99 -pedantic -Wextra -Wno-missing-field-initializers -Wshadow -Wpointer-arith -Wstrict-prototypes -Wmissing-prototypes -Wno-unused-parameter -O2 -Wstrict-overflow=1 -fstack-protector -D_FORTIFY_SOURCE=2 -Wformat -Wformat-security -fPIC -MT LzFind.lo -MD -MP -MF
```

FIGURE 4.9 – Compilation Suricata

## 7. L'installation du Suricata :

```

riusers@raspberrypi:~/suricata-6.0.1 $ sudo make install
[sudo] password for riusers:
Making install in libhttp
make[1]: Entering directory '/home/riusers/suricata-6.0.1/libhttp'
Making install in http
make[2]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
Making install in lzma
make[3]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[4]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[4]: Nothing to be done for 'install-exec-am'.
/usr/bin/mkdir -p '/usr/include/http/lzma'
/usr/bin/install -c -m 644 LzmaDec.h '/usr/include/http/lzma'
make[4]: Leaving directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[3]: Leaving directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[3]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
make[4]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
/usr/bin/mkdir -p '/usr/lib'
/bin/bash ./libtool --mode=install /usr/bin/install -c libhttp.la '/usr/lib/libtool: i
nstall: /usr/bin/install -c .libs/libhttp.so.2.0.0 /usr/lib/libhttp.so.2.0.0
libtool: install: (cd /usr/lib && { ln -s -f libhttp.so.2.0.0 libhttp.so.2 || { rm -f libhttp.
so.2 && ln -s libhttp.so.2.0.0 libhttp.so.2; }; })
libtool: install: (cd /usr/lib && { ln -s -f libhttp.so.2.0.0 libhttp.so || { rm -f libhttp.so
&& ln -s libhttp.so.2.0.0 libhttp.so; }; })
libtool: install: /usr/bin/install -c .libs/libhttp.lai /usr/lib/libhttp.la
libtool: install: /usr/bin/install -c .libs/libhttp.a /usr/lib/libhttp.a
libtool: install: chmod 644 /usr/lib/libhttp.a
libtool: install: ranlib /usr/lib/libhttp.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
ldconfig -n /usr/lib
-----
Libraries have been installed in:
  /usr/lib

```

FIGURE 4.10 – Installation Suricata

8. On se placer dans le dossier suricata-update puis on l'installe après sa compilation :

```

riusers@raspberrypi:~/suricata-6.0.1 $ cd $HOME/suricata-6.0.1/suricata-update/
riusers@raspberrypi:~/suricata-6.0.1/suricata-update $ sudo python setup.py build
running build_py
creating build
creating build/lib
creating build/lib/suricata
copying suricata/__init__.py -> build/lib/suricata
creating build/lib/suricata/update
copying suricata/update/maps.py -> build/lib/suricata/update
copying suricata/update/rule.py -> build/lib/suricata/update
copying suricata/update/revision.py -> build/lib/suricata/update
copying suricata/update/exceptions.py -> build/lib/suricata/update
copying suricata/update/__init__.py -> build/lib/suricata/update
copying suricata/update/matchers.py -> build/lib/suricata/update
copying suricata/update/config.py -> build/lib/suricata/update
copying suricata/update/net.py -> build/lib/suricata/update
copying suricata/update/parsers.py -> build/lib/suricata/update

```

FIGURE 4.11 – Compilation suricata-update

```

riusers@raspberrypi:~/suricata-6.0.1/suricata-update $ sudo python setup.py install
running install
running build
running build_py
copying suricata/update/revision.py -> build/lib/suricata/update
running build_scripts
running install_lib
copying build/lib/suricata/update/revision.py -> /usr/local/lib/python3.9/dist-packages/sur
icata/update
byte-compiling /usr/local/lib/python3.9/dist-packages/suricata/update/revision.py to revisi
on.cpython-39.pyc
running install_scripts
copying build/scripts-3.9/suricata-update -> /usr/local/bin
changing mode of /usr/local/bin/suricata-update to 755
running install_egg_info
Writing /usr/local/lib/python3.9/dist-packages/suricata_update-1.2.0.egg-info

```

FIGURE 4.12 – Installation suricata-update

9. Dans le dossier Suricata, on finalise l'installation, on inclut les règles de suricata

en les mettant à jour :

```
riusers@raspberrypi:~/suricata-6.0.1$ cd $HOME/suricata-6.0.1/
riusers@raspberrypi:~/suricata-6.0.1$ sudo make install-full
make install
make[1]: Entering directory '/home/riusers/suricata-6.0.1'
Making install in libhttp
make[2]: Entering directory '/home/riusers/suricata-6.0.1/libhttp'
Making install in http
make[3]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
Making install in lzma
make[4]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[5]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[5]: Nothing to be done for 'install-exec-am'.
/usr/bin/mkdir -p '/usr/include/http/lzma'
/usr/bin/install -c -m 644 LzmaDec.h '/usr/include/http/lzma'
make[5]: Leaving directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[4]: Leaving directory '/home/riusers/suricata-6.0.1/libhttp/http/lzma'
make[4]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
make[5]: Entering directory '/home/riusers/suricata-6.0.1/libhttp/http'
/usr/bin/mkdir -p '/usr/lib'
/bin/bash ../libtool --mode=install /usr/bin/install -c libhttp.la '/usr/lib/libtool: i
nstall: /usr/bin/install -c .libs/libhttp.so.2.0.0 /usr/lib/libhttp.so.2.0.0
libtool: install: (cd /usr/lib && { ln -s -f libhttp.so.2.0.0 libhttp.so.2 || { rm -f libhttp.
so.2 && ln -s libhttp.so.2.0.0 libhttp.so.2; }; })
libtool: install: (cd /usr/lib && { ln -s -f libhttp.so.2.0.0 libhttp.so || { rm -f libhttp.so
&& ln -s libhttp.so.2.0.0 libhttp.so; }; })
libtool: install: /usr/bin/install -c .libs/libhttp.lai /usr/lib/libhttp.la
libtool: install: /usr/bin/install -c .libs/libhttp.a /usr/lib/libhttp.a
libtool: install: chmod 644 /usr/lib/libhttp.a
libtool: install: ranlib /usr/lib/libhttp.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
ldconfig -n /usr/lib
```

FIGURE 4.13 – Installation des règles de Suricata

```
riusers@raspberrypi:~/suricata-6.0.1$ sudo suricata-update
31/5/2022 -- 01:39:54 - <Info> -- Using data-directory /var/lib/suricata.
31/5/2022 -- 01:39:54 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
31/5/2022 -- 01:39:54 - <Info> -- Using /usr/share/suricata/rules for Suricata provided rul
es.
31/5/2022 -- 01:39:54 - <Info> -- Found Suricata version 6.0.1 at /usr/bin/suricata.
31/5/2022 -- 01:39:54 - <Info> -- Loading /etc/suricata/suricata.yaml
31/5/2022 -- 01:39:54 - <Error> -- [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - Failed to parse
configuration file at line 1884: found character that cannot start any token

riusers@raspberrypi:~/suricata-6.0.1$
```

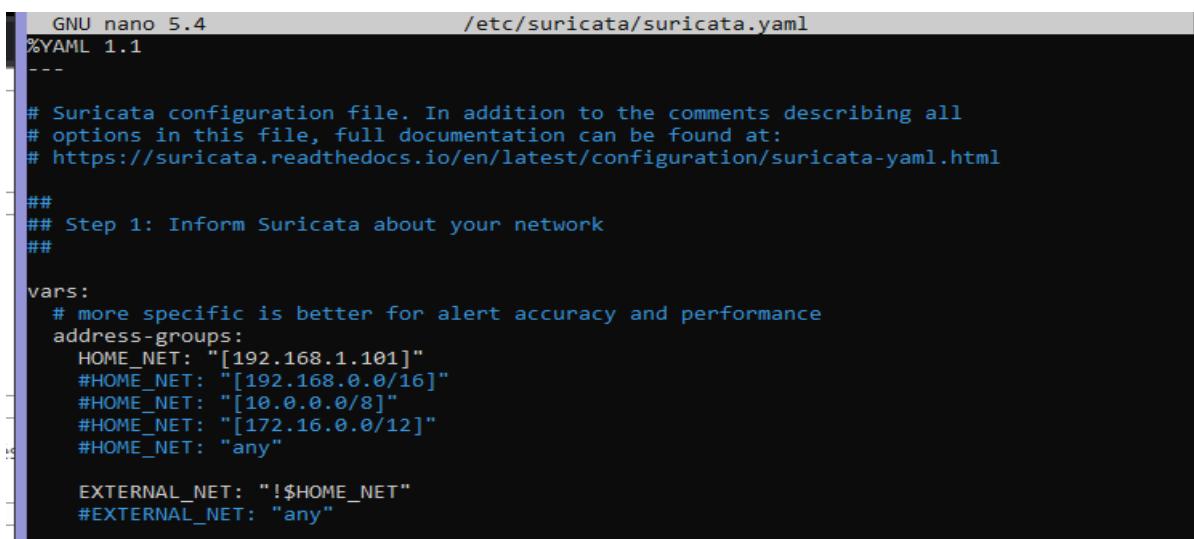
FIGURE 4.14 – Mis à jour des règles de suricata

## 4.5 Configuration du Suricata :

Pour configurer Suricata, il faut tout d'abord connaitre l'adresse locale de notre Raspberry Pi. Ensuite, on modifie la variable HOME NET, qui se trouve dans le fichier suicata.yaml, afin qu'elle contienne notre réseau local :

```
riusers@raspberrypi:~/suricata-6.0.1$ hostname -I
0.1.6.117 172.17.0.1
riusers@raspberrypi:~/suricata-6.0.1$ sudo nano /etc/suricata/suricata.yaml
[sudo] password for riusers:
riusers@raspberrypi:~/suricata-6.0.1$ sudo nano /etc/suricata/suricata.yaml
```

FIGURE 4.15 – Accès au fichier suricata/yaml



```

GNU nano 5.4                               /etc/suricata/suricata.yaml
%YAML 1.1
---

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://suricata.readthedocs.io/en/latest/configuration/suricata-yaml.html

## Step 1: Inform Suricata about your network

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.1.101]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

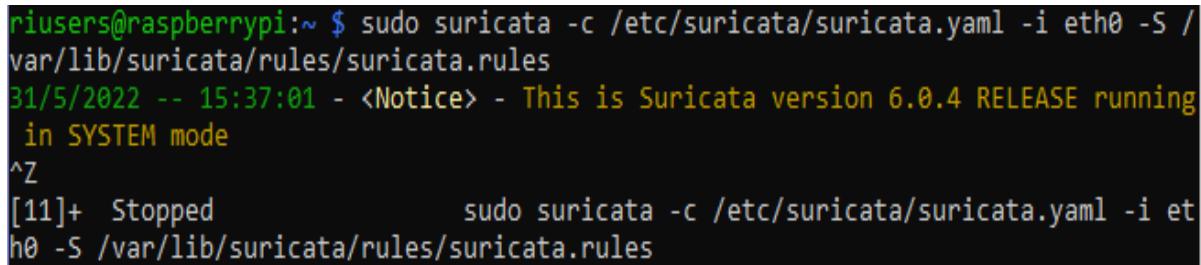
```

FIGURE 4.16 – suricata.yaml

## 4.6 Lancement du Suricata :

Après son installation, on lance Suricata avec la commande représentée dans la figure suivante, avec :

- c : fichier de configuration à utiliser.
- i : interface Ethernet à surveiller.
- S : fichier contenant les règles à utiliser.



```

pi@raspberrypi:~ $ sudo suricata -c /etc/suricata/suricata.yaml -i eth0 -S /var/lib/suricata/rules/suricata.rules
31/5/2022 -- 15:37:01 - <Notice> - This is Suricata version 6.0.4 RELEASE running
in SYSTEM mode
^Z
[11]+  Stopped                  sudo suricata -c /etc/suricata/suricata.yaml -i et
h0 -S /var/lib/suricata/rules/suricata.rules

```

FIGURE 4.17 – Lancement de suricata

## 4.7 Test du Suricata :

Pour assurer le bon fonctionnement de Suricata, on fait un test qui consiste à ajouter une règle qui affiche un avertissement à chaque réception d'un ICMP Echo (ping). On se place donc dans le fichier suricata.rules et on ajoute la règle suivante : **alert icmp any any -gt; any any (msg : "ICMP Packet found"; sid : 1; rev : 1;)**.

Et lorqu'on affiche le fichier de log on remarquera l'alerte suivante :

NB : On supprime cette règle après ce test.

```

root@raspberrypi:~ $ curl: (6) Could not resolve host: 3wzrSpoyJumh7skj.onion
root@raspberrypi:~ $ curl: (6) Could not resolve host: 3wzrSpoyJumh7skj.onion
root@raspberrypi:~ $ /var/lib/suricata/rules/suricata.rules
root@raspberrypi:~ $ /var/lib/suricata/rules/suricata.rules
root@raspberrypi:~ $ sudo nano /var/lib/suricata/rules/suricata.rules
root@raspberrypi:~ $ sudo tail -f /var/log/suricata/fast.log
[1]+  Stopped                  sudo tail -f /var/log/suricata/fast.log

```

FIGURE 4.18 – résultat de test du suricata

## 4.8 Utilisation du Suricata en mode service :

Pour utiliser suricata en tant que service, on crée un fichier dans :

**/etc/systemd/system/suricata.service** qui contient le contenu de la figure 4.19  
Puis on l'active.

```

root@raspberrypi:~ $ nano /etc/systemd/system/suricata.service
# Sample Suricata systemd unit file.
[Unit]
Description=Suricata Intrusion Detection Service
After=network.target syslog.target
[Service]
ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml -i eth0 -S /var/lib/suricata/rules/suricata.rules
ExecReload=/bin/kill -HUP $MAINPID
ExecStop=/bin/kill $MAINPID
[Install]
WantedBy=multi-user.target

```

FIGURE 4.19 – Contenu de fichier suricata.service

```

root@raspberrypi:~ $ sudo nano /etc/systemd/system/suricata.service
root@raspberrypi:~ $ sudo systemctl enable suricata.service
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
root@raspberrypi:~ $ sudo systemctl start suricata.service
root@raspberrypi:~ $ sudo systemctl stop suricata.service
root@raspberrypi:~ $ sudo systemctl restart suricata.service
root@raspberrypi:~ $ sudo systemctl status suricata.service
● suricata.service - Suricata Intrusion Detection Service
   Loaded: loaded (/etc/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-05-31 15:53:21 +01; 43s ago
     Main PID: 3129 (Suricata-Main)
        Tasks: 10 (limit: 4915)
       CPU: 39.412s
      CGroup: /system.slice/suricata.service
              └─3129 /usr/bin/suricata -c /etc/suricata/suricata.yaml -i eth0 -S /var/lib/suricata/rules/suricata.rules

May 31 15:53:21 raspberrypi systemd[1]: Started Suricata Intrusion Detection Service.
May 31 15:53:21 raspberrypi suricata[3129]: 31/5/2022 -- 15:53:21 - <Notice> - T>
May 31 15:54:00 raspberrypi suricata[3129]: 31/5/2022 -- 15:54:00 - <Notice> - a>
[12]+  Stopped                  sudo systemctl status suricata.service

```

FIGURE 4.20 – activation de suricata.service

## 4.9 Utilisation des ressources du Raspberry Pi :

Dans les figures suivantes, on trouve la consommation des ressources de notre Raspberry Pi 4 modèle B faisant tourner Suricata :

- Utilisation de processeur :

```
top - 16:01:00 up 9 min, 3 users, load average: 0.03, 0.20, 0.16
Tasks: 190 total, 1 running, 189 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.9 us, 1.4 sy, 0.0 ni, 97.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7898.9 total, 6632.6 free, 660.7 used, 605.6 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used. 6987.4 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 551 root      20   0  577348 457844  8240 S 17.8  5.7  2:12.04 Suricata-M+
 888 root      20  -20      0      0     0 I 0.3  0.0  0:02.00 kworker/u9+
 1523 riusers  20   0   11364  2952  2496 R 0.3  0.0  0:00.29 top
    1 root      20   0   33904  8808  6924 S 0.0  0.1  0:03.13 systemd
    2 root      20   0      0      0     0 S 0.0  0.0  0:00.02 kthreadd
    3 root      20  -20      0      0     0 I 0.0  0.0  0:00.00 rcu_gp
```

FIGURE 4.21 – Consommation du CPU

- Utilisation de la mémoire :

```
riusers@raspberrypi:~ $ free
              total        used        free      shared  buff/cache   available
Mem:       8088496      676720      6792020        22456      619756      7155052
Swap:      102396          0      102396
```

FIGURE 4.22 – Consommation du mémoire

- Utilisation de la carte SD :

```
riusers@raspberrypi:~ $ df
Filesystem  1K-blocks  Used  Available Use% Mounted on
/dev/root    29583108 5633120  22693940  20% /
devtmpfs     3879384      0  3879384   0% /dev
tmpfs        4044248      0  4044248   0% /dev/shm
tmpfs        1617700    1244  1616456   1% /run
tmpfs         5120        4    5116   1% /run/lock
/dev/mmcblk0p1  258095  50413   207683  20% /boot
tmpfs        808848      20  808828   1% /run/user/1000
tmpfs        808848      16  808832   1% /run/user/1001
```

FIGURE 4.23 – Consommation du carte SD

## Conclusion Générale

---

Notre Raspberry Pi est maintenant prêt à détecter les intrusions dans un réseau donné, il doit seulement recevoir une copie de tout le trafic réseau qui doit être analysé.

l'utilisation de Raspberry Pi en tant qu' un IDS est une solution pratique pour surveiller le réseau, et surtout les petits réseaux comme les réseaux domestiques et ceux des petites entreprises. Des améliorations pourraient aussi être apportées à cette solution, grâce à l'évolution continue de RPi au niveau de ses performances, comme le passage de la détection à la prévention et au contrôle pour achever un haut niveau de sécurité.

Malgré les petites difficultés qu'on a rencontrées tout au long de la réalisation de ce projet, ceci fut très intéressant et très instructif pour nous. En outre, ce travail était une belle occasion pour travailler en groupe et de développer nos connaissances et nos compétences et surtout pour découvrir de nouveaux outils.



# Bibliographie

- [1] Article in IEEE Consumer Electronics Magazine, Security Vulnerabilities in Raspberry Pi—Analysis of the System Weaknesses, November 2019, <https://ieeexplore.ieee.org/document/8889544>
- [2] Robin Mitchell, Raspberry Pi For Beginners : The Basics, 23-08-2018, <https://maker.pro/raspberry-pi/tutorial/an-intro-to-raspberry-pi-and-its-fundamentals>
- [3] Oğuzhan Karahan and Berat Kaya, Research Article : Raspberry Pi Firewall and Intrusion Detection System, Journal of Intelligent Systems : Theory and Applications 3(2) 2020 : 21-24, <https://dergipark.org.tr/>
- [4] Raspberry Pi Digital Privacy Risks, 09-01-2018, <https://choosetoencrypt.com/tech/raspberry-pi-digital-privacy>
- [5] Raspberry Pi Documentation, <https://www.raspberrypi.com/documentation/>
- [6] Tom Van De Wiele and Justin Klein Keane, Take These Steps to Secure Your Raspberry Pi Against Attackers, 07-09-2017, <https://makezine.com/>
- [7] What is a Raspberry Pi and How Does it Work ?, <https://www.piday.org/>
- [8] Know all about Raspberry Pi Board Technology, 26-07-2019, <https://www.watelectronics.com/>
- [9] CHRISTIAN CAWLEY, Is Your Raspberry Pi Safe and Secure ?, 14-09-2017, <https://www.makeuseof.com/tag/raspberry-pi-safe-secure/>
- [10] Comparatif Raspberry Pi : quel modèle choisir ?, 31-10-2021, <https://framboisepi.fr/comparatif-raspberry-pi/>
- [11] Thommy SIMONSSON and Andreas ASPERNÄS, IDS on Raspberry Pi—A Performance Evaluation, Diploma Thesis.
- [12] Snort IDPS using Raspberry Pi 4, 07-07-2020, <https://www.ijert.org/>
- [13] Juliana Fajardini, Spot suspicious activity on your local network with Suricata Intrusion Detection System (IDS) on Raspberry Pi, 15-02-202, <https://jufajardini.wordpress.com/>
- [14] Suricata Documentation, <https://suricata.readthedocs.io/en/latest/rules/intro.html>

- [15] Amrita Pathak, IDS vs IPS : un guide complet des solutions de sécurité réseau,  
31-08-2021, <https://geekflare.com/fr/ids-vs-ips-network-security-solutions/>