ParkFinder Inc.

Audit de Sécurité

24/06/2021

Table des matières

1- Introduction

L'audit de sécurité d'un système d'information (SI) est une vue à un instant T de l'intégralité ou d'une partie du SI, permettant de comparer l'état du SI à un référentiel. L'audit répertorie les points forts, et surtout les points faibles (vulnérabilités) de l'intégralité ou d'une partie du système. Cela nous permettra aussi de dresser également une série de recommandations pour supprimer les vulnérabilités découvertes.

2 – Audit

2.1 – Contexte de l'Audit

Cet audit a pour but de détecter la plupart des vulnérabilités dans notre application web / mobile et d'établir des contre-mesures pour éviter tout accès indésirable dans notre réseau ou bien même une perte de données clientèles. Pour ainsi être conforme aux normes iso.

2.2 – Périmètre de l'Audit

Cet audit va être effectué sur l'intégralité du système d'information et aura pour but de détecter les failles qui jouent le rôle de points d'entrée fréquents pour les attaques malicieuses.

Il ne va pas s'intéresser au social engineering mais plutôt sur les failles présentes sur notre infrastructure et sur les technologies utilisées dans cette dernière.

2.3 – Préparation de l'Audit

Pour préparer notre audit on aura besoin d'effectuer une recherche sur les différentes vulnérabilités des différentes technologies utilisées dans notre SI. Après avoir effectué une recherche et une analyse on passera ensuite à un audit du code source de l'application pour finir avec un test d'intrusion.

2.3.1 – Audit des Technologies

Cet audit nous permettra de définir les vulnérabilités existantes au niveau des technologies utilisées dans notre système d'informations.

Au niveau de notre système d'information on retrouve différentes technologies qui sont définis comme ceci :

Nodejs (v15.14.0), MySql (v2.18.1), Express (v4.16.1), Html, CSS, Jquery (3.3.1 & 3.6.0), Bootstrap, Laravel (8.47.0), JavaScript, PhpMyAdmin, Python (3.6).

Dans ces différentes technologies certaines sont exposées à différents type d'attaque :

- JQuery : La version de JQuery 3.3.1 est exposé à 3 vulnérabilités d'une gravité intermédiaire (CVE-2020-11023 : Vulnerable to Cross-site Scripting (XSS)), (CVE-2020-11022 : Vulnerable to Cross-site Scripting (XSS)), (CVE-2019-11358 : Prototype Pollution)

- JavaScript : JavaScript est exposé à 8 vulnérabilités communes exploité par les hackers, parmi ces 8 vulnérabilités on en retrouve 2 qui sont critiques (Cross-site Scripting (XSS) & Cross-site Request Forgery (CSRF)

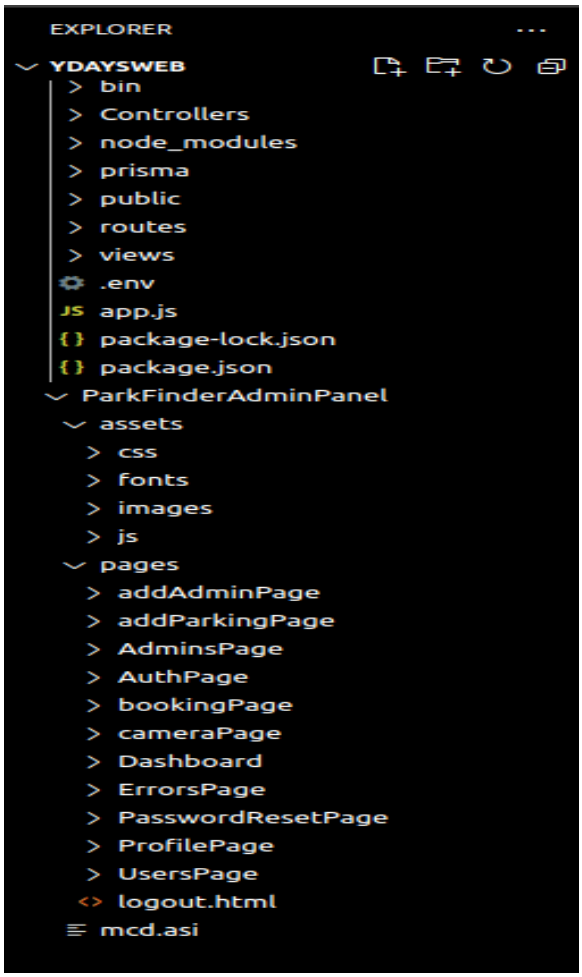
2.3.2– Audit du Code

Dans cette partie-là on va analyser le code source de notre application en utilisant une analyse statique du code source.

Cette analyse nous permettra de « sanitize » notre code source pour filtrer les users, API, services web data inputs de tous les caractères et chaines indésirables qui mèneront à des injections de code malicieux dans notre système.

Pour effectuer cette analyse on s'est basé sur le principe du linting et l'outil utilisé pour analyser notre code est ESlint, ce dernier analysera la structure de notre code en se basant sur différentes règles prédéfini et les corrigera.

La structure de notre code est définie comme ceci :



En implémentant ESlint dans Visual Code on retrouvera un output de tout les problèmes au niveau de notre code qu'on devra corriger :

```
PS C:\Users\shaux\Desktop\YdaysWeb\BackEndWeb> ./node_modules/.bin/eslint --init
√ How would you like to use ESLint? · style
√ What type of modules does your project use? · esm
√ Which framework does your project use? · none
√ Does your project use TypeScript? · No / Yes
√ Where does your code run? · node
√ How would you like to define a style for your project? · guide
√ Which style guide do you want to follow? · airbnb
√ What format do you want your config file to be in? · JSON
Checking peerDependencies of eslint-config-airbnb-base@latest
The config that you've selected requires the following dependencies:

eslint-config-airbnb-base@latest eslint@^5.16.0 || ^6.8.0 || ^7.2.0 eslint-plugin-import@^2.22.1
√ Would you like to install them now with npm? · No / Yes
Installing eslint-config-airbnb-base@latest, eslint@^5.16.0 || ^6.8.0 || ^7.2.0, eslint-plugin-import@^2.22.1

up to date, audited 435 packages in 2s

52 packages are looking for funding
  run `npm fund` for details
```

Par exemple en scannant notre fichier app.js on retrouve différentes erreurs :

```
PS C:\Users\shaux\Desktop\YdaysWeb\BackEndWeb> ./node_modules/.bin/eslint ./app.js

C:\Users\shaux\Desktop\YdaysWeb\BackEndWeb\app.js
   1:1   error    Unexpected var, use let or const instead                                                    no-var
   2:1   error    Unexpected var, use let or const instead                                                    no-var
   3:1   error    Unexpected var, use let or const instead                                                    no-var
   4:1   error    Unexpected var, use let or const instead                                                    no-var
   5:1   error    Unexpected var, use let or const instead                                                    no-var
   6:1   error    Unexpected var, use let or const instead                                                    no-var
   6:27  error    Missing semicolon                                                                           semi
   7:1   error    Unexpected var, use let or const instead                                                    no-var
   8:1   error    Unexpected var, use let or const instead                                                    no-var
   8:5   error    'bodyParser' is assigned a value but never used                                             no-unused-vars
   8:18  error    'body-parser' should be listed in the project's dependencies. Run 'npm i -S body-parser' to add it  import/no-extraneous-dependencies
   8:40  error    Missing semicolon                                                                           semi
  10:22  error    Missing space before value for key 'path'                                                   key-spacing
  10:31  error    A space is required before '}'                                                              object-curly-spacing
  10:33  error    Missing semicolon                                                                           semi
  12:1   error    All 'var' declarations must be at the top of the function scope                             vars-on-top
  12:1   error    Unexpected var, use let or const instead                                                    no-var
  13:1   error    All 'var' declarations must be at the top of the function scope                             vars-on-top
  13:1   error    Unexpected var, use let or const instead                                                    no-var
  15:1   error    More than 1 blank line not allowed                                                          no-multiple-empty-lines
  16:1   error    All 'var' declarations must be at the top of the function scope                             vars-on-top
  16:1   error    Unexpected var, use let or const instead                                                    no-var
  24:1   error    More than 1 blank line not allowed                                                          no-multiple-empty-lines
  31:1   error    More than 1 blank line not allowed                                                          no-multiple-empty-lines
  32:19  error    Trailing spaces not allowed                                                                 no-trailing-spaces
  42:9   error    Unexpected function expression                                                              prefer-arrow-callback
  42:9   warning  Unexpected unnamed function                                                                 func-names
  42:17  error    Missing space before function parentheses                                                   space-before-function-paren
  47:9   error    Unexpected function expression                                                              prefer-arrow-callback
  47:9   warning  Unexpected unnamed function                                                                 func-names
  47:17  error    Missing space before function parentheses                                                   space-before-function-paren
  47:33  error    'next' is defined but never used                                                            no-unused-vars

✖ 33 problems (31 errors, 2 warnings)
  25 errors and 0 warnings potentially fixable with the `--fix` option.
```

```
'bodyParser' is declared but its value is never read. ts(6133)

(alias) function bodyParser(options?: bodyParser.OptionsJson & bodyParser.OptionsText & bodyParser.OptionsUrlencoded):
NextHandleFunction
(alias) namespace bodyParser
import bodyParser

@deprecated

Quick Fix... (Ctrl+Shift+.)
```

Après avoir détecter les problèmes ESlint nous permettra de les corriger que ce soit en rapport avec des variables non utilisés, des champs vides ou bien même des virgules ou quottes manquantes.

2.3.3 – Test d'intrusion

Dans cette partie-là on effectuera des tests d'intrusion sur notre système d'information, pour cela on va utiliser kali linux pour faire un scan entier du réseau de notre infrastructure et identifier les risques potentiels sur notre système.

En premier lieu on va effectuer un scan Nmap pour connaitre les ports utilisés (ouvert) au niveau de notre infrastructure

```
└─# nmap -p- -sV 192.168.179.134
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-26 14:28 +01
Nmap scan report for 192.168.179.134
Host is up (0.0030s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain     ISC BIND 9.16.1 (Ubuntu Linux)
80/tcp   open  http       Apache httpd 2.4.41
9100/tcp open  jetdirect?
MAC Address: 00:0C:29:B8:48:E1 (VMware)
Service Info: Host: parkfinder.ynov; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.42 seconds
```

Après avoir effectué le scan on remarque que 3 ports sont ouverts :

- Port 22 (SSH) est ouvert et peut être exploité pour établir une connexion distante vers le serveur parkfinder.

- Port 53 (DNS) qui peut être exploité pour établir une attaque d'énumération et récolter des informations.

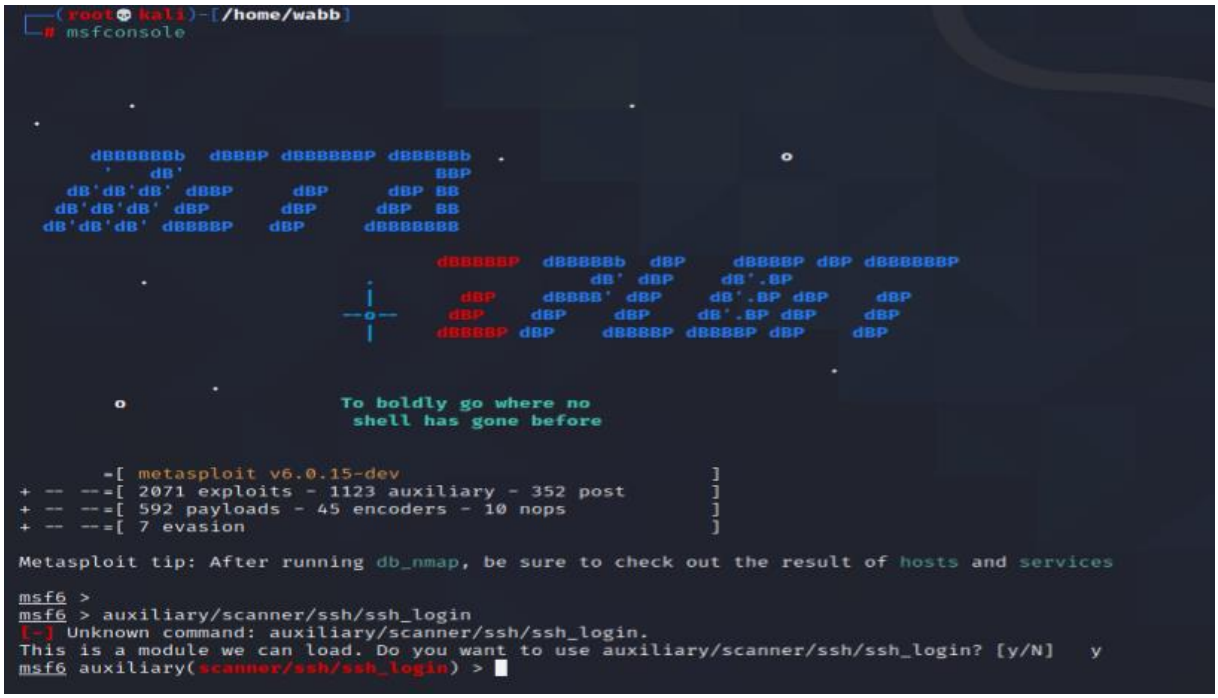
- Port 80 (HTTP) qui peut être exploité pour recueillir des informations sur notre serveur.

En utilisant un script NSE (NMAP Scripting Engine) qui regroupe plusieurs CVE stockés dans des bases de données local on pourra avoir plus de détails sur comment exploité ces 3 ports à travers différentes techniques.

```
SecurityTracker - https://www.securitytracker.com:
[1028187] OpenSSH pam_ssh_agent_auth Module on Red Hat Enterprise Linux Lets Remote Users Execute Arbitrary Code
[1026593] OpenSSH Lets Remote Authenticated Users Obtain Potentially Sensitive Information
[1025739] OpenSSH on FreeBSD Has Buffer Overflow in pam_thread() That Lets Remote Users Execute Arbitrary Code
[1025482] OpenSSH ssh-keysign Utility Lets Local Users Gain Elevated Privileges
[1025020] OpenSSH Legacy Certificates May Disclose Stack Contents to Remote Users
[1022967] OpenSSH on Red Hat Enterprise Linux Lets Remote Authenticated Users Gain Elevated Privileges
[1021235] OpenSSH CBC Mode Error Handling May Let Certain Remote Users Obtain Plain Text in Certain Cases
[1020891] OpenSSH on Debian Lets Remote Users Prevent Logins
[1020730] OpenSSH for Red Hat Enterprise Linux Packages May Have Been Compromised
[1020537] OpenSSH on HP-UX Lets Local Users Hijack X11 Sessions
[1019733] OpenSSH Unsafe Default Configuration May Let Local Users Execute Arbitrary Commands
[1019707] OpenSSH Lets Local Users Hijack Forwarded X Sessions in Certain Cases
[1017756] Apple OpenSSH Key Generation Process Lets Remote Users Deny Service
[1017183] OpenSSH Privilege Separation Monitor Validation Error May Cause the Monitor to Fail to Properly Control the Unprivileged Process
[1016948] OpenSSH Race Condition in Signal Handler Lets Remote Users Deny Service and May Potentially Permit Code Execution
[1016939] OpenSSH GSSAPI Authentication Abort Error Lets Remote Users Determine Valid Usernames
[1016931] OpenSSH SSH v1 CRC Attack Detection Implementation Lets Remote Users Deny Service
[1016672] OpenSSH on Mac OS X Lets Remote Users Deny Service
[1015780] OpenSSH Interaction With OpenPAM Lets Remote Users Deny Service
[1015540] OpenSSH scp Double Shell Character Expansion During Local-to-Local Copying May Let Local Users Gain Elevated Privileges in Certain Cases
[1014845] OpenSSH May Unexpectedly Activate GatewayPorts and Also May Disclose GSSAPI Credentials in Certain Cases
[1011103] OpenSSH scp Directory Traversal Flaw Lets Remote SSH Servers Overwrite Files in Certain Cases
[1011163] OpenSSH Default Configuration May Be Unsafe When Used With Anonymous SSH Services
[1007791] Portable OpenSSH PAM free() Bug May Let Remote Users Execute Root Code
[1007716] OpenSSH buffer_append_space() and Other Buffer Management Errors May Let Remote Users Execute Arbitrary Code
[1006926] OpenSSH Host Access Restrictions Can Be Bypassed By Remote Users
[1006688] OpenSSH Timing Flaw With Pluggable Authentication Modules Can Disclose Valid User Account Names to Remote Users
[1004818] OpenSSH's Secure Shell (SSH) Implementation Weakness May Disclose User Passwords to Remote Users During Man-in-the-Middle Attacks
[1004616] OpenSSH Integer Overflow and Buffer Overflow May Allow Remote Users to Gain Root Access to the System
[1004391] OpenSSH 'BSD_AUTH' Access Control Bug May Allow Unauthorized Remote Users to Authenticated to the System
[1004115] OpenSSH Buffer Overflow in Kerberos Ticket and AFS Token Processing Lets Local Users Execute Arbitrary Code With Root Level Permissions
[1003750] OpenSSH OFF-by-one 'Channels' Bug May Let Authorized Remote Users Execute Arbitrary Code with Root Privileges
[1002895] OpenSSH UseLogin Environment Variable Bug Lets Local Users Execute Commands and Gain Root Access
[1002748] OpenSSH 3.0 Denial of Service Condition May Allow Remote Users to Crash the sshd Daemon and KerberosV Configuration Error May Allow Remote Users to Partially Authenticate When Authentication Should Not Be Permitted
[1002734] OpenSSH's S/Key Implementation Information Disclosure Flaw Provides Remote Users With Information About Valid User Accounts
[1002455] OpenSSH May Fail to Properly Restrict IP Addresses in Certain Configurations
[1002432] OpenSSH's Sftp-server Subsystem Lets Authorized Remote Users with Restricted Keypairs Obtain Additional Access on the Server
[1001683] OpenSSH Allows Authorized Users to Delete Other User Files Named Cookies
```

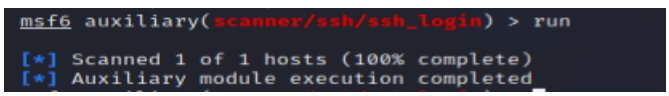
Après avoir scanné notre réseau on essayera d'exploité le port 22 pour établir une connexion distante au serveur Parkfinder.

Pour cela on va créer 2 fichiers : un fichier users et un password ou on va mettre les mots de passe les plus communs (admin etc…) et ensuite on executera un module de test de connexion SSH.



Ensuite on va définir les 2 fichiers users et password qu'on a créé auparavant et préciser l'adresse IP du serveur ou on veut tester notre connexion SSH.

Après avoir défini ces éléments là on aura juste à exécuter le module et ce dernier tentera de se connecter au serveur Parkfinder en utilisant les crédentials mentionné dans les 2 fichiers users et password.



La connexion a été établie avec succès et une session SSH a été créé. Il suffit d'ouvrir la session et on sera sur le Shell du serveur parkfinder.

Ceci est un exemple d'une des méthodes qu'on peut utiliser pour exploiter le port 22 ouvert d'un serveur et il en existe plusieurs (Hydra, script Nmap pour bruteforce…).

2.2.4 – Sécurisation de l'infrastructure

Dans cette dernière partie on sécurisera notre infrastructure en se basant sur le processus du hardening. Ce dernier nous permettra de sécuriser la connexion SSH (qu'on a exploité dans les tests d'intrusion), de garder le système d'information à jour mais aussi d'établir un firewall pour contrôler le traffic et de monitorer ce dernier (ainsi que les ressources matérielles).

2.2.4.1- Processus du hardening

Au niveau de cette partie on va premièrement intégrer des mises à jour régulière pour garder notre système à jour. Ensuite on va implémenter des password policy et sécuriser la connexion SSH, et dernièrement on va implémenter un FireWall pour contrôler le traffic et bloquer l'utilisation de certains ports.

On va devoir créer un nouvel utilisateur qui jouera le rôle d'admin et qui pourra se connecter à distance au serveur. On va générer une clé RSA et on va enlever la connexion via Root et la connexion ou on demande un password.

```
root@parkfinder-virtual-machine:/var/www/parkfinder.ynov#  chmod 700 ~/.ssh
root@parkfinder-virtual-machine:/var/www/parkfinder.ynov# ssh-keygen -b 8192
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)?
root@parkfinder-virtual-machine:/var/www/parkfinder.ynov# cd .ssh
bash: cd: .ssh: No such file or directory
root@parkfinder-virtual-machine:/var/www/parkfinder.ynov# ls
BackEndAPI  ParkFinderAdminPanel
root@parkfinder-virtual-machine:/var/www/parkfinder.ynov# cd  ~/.ssh
root@parkfinder-virtual-machine:~/.ssh# ssh-copy-id parkadmin@192.168.180.179
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
parkadmin@192.168.180.179's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'parkadmin@192.168.180.179'"
and check to make sure that only the key(s) you wanted were added.

root@parkfinder-virtual-machine:~/.ssh# ssh parkadmin@192.168.180.179
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-59-generic x86_64)
```

```
root@parkfinder-virtual-machine:/# sudo nano /etc/ssh/sshd_config
root@parkfinder-virtual-machine:/# sudo systemctl restart sshd
root@parkfinder-virtual-machine:/# ssh parkadmin@192.168.180.179 -p 6969
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Sun Jun 27 13:11:46 2021 from 192.168.180.179
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

parkadmin@parkfinder-virtual-machine:~$ exit
logout
```

Ensuite on va établir un FireWall et on va y ajouter des règles pour permettre ou refuser des requêtes venant de certaines adresse IP ou des requêtes utilisant certains ports/protocoles (voir même les bloquer).

```
parkadmin@parkfinder-virtual-machine:~$ sudo ss -tupln
Netid       State        Recv-Q       Send-Q             Local Address:Port             Peer Address:Port
Process
udp         UNCONN       0            0                        0.0.0.0:631                   0.0.0.0:*
 users:(("cups-browsed",pid=905,fd=7))
udp         UNCONN       0            0                        0.0.0.0:5353                  0.0.0.0:*
 users:(("avahi-daemon",pid=740,fd=12))
udp         UNCONN       0            0                  127.0.0.53%lo:53                    0.0.0.0:*
 users:(("systemd-resolve",pid=12435,fd=12))
udp         UNCONN       0            0                192.168.180.179:53                    0.0.0.0:*
 users:(("named",pid=923,fd=30))
udp         UNCONN       0            0                192.168.180.179:53                    0.0.0.0:*
 users:(("named",pid=923,fd=31))
udp         UNCONN       0            0                      127.0.0.1:53                    0.0.0.0:*
 users:(("named",pid=923,fd=24))
udp         UNCONN       0            0                      127.0.0.1:53                    0.0.0.0:*
 users:(("named",pid=923,fd=23))
udp         UNCONN       0            0                        0.0.0.0:47304                 0.0.0.0:*
 users:(("avahi-daemon",pid=740,fd=14))
udp         UNCONN       0            0                           [::]:5353                     [::]:*
 users:(("avahi-daemon",pid=740,fd=13))
udp         UNCONN       0            0                           [::]:38570                    [::]:*
 users:(("avahi-daemon",pid=740,fd=15))
tcp         LISTEN       0            4096               127.0.0.53%lo:53                    0.0.0.0:*
 users:(("systemd-resolve",pid=12435,fd=13))
tcp         LISTEN       0            10               192.168.180.179:53                    0.0.0.0:*
 users:(("named",pid=923,fd=34),("named",pid=923,fd=33),("named",pid=923,fd=32))
tcp         LISTEN       0            10                     127.0.0.1:53                    0.0.0.0:*
 users:(("named",pid=923,fd=28),("named",pid=923,fd=27),("named",pid=923,fd=26))
```

On s’est débarrasser du port 631 qu’on n’utilise pas et on a permis le port SSH qu’on a modifié, on a aussi bloqué les requêtes utilisant le protocole. ICMP.

```
parkadmin@parkfinder-virtual-machine:~$ sudo ufw allow 717
Rules updated
Rules updated (v6)
parkadmin@parkfinder-virtual-machine:~$ sudo ufw deny 631
Rules updated
Rules updated (v6)
parkadmin@parkfinder-virtual-machine:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
parkadmin@parkfinder-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache Full                ALLOW       Anywhere
717                        ALLOW       Anywhere
631                        DENY        Anywhere
Apache Full (v6)           ALLOW       Anywhere (v6)
717 (v6)                   ALLOW       Anywhere (v6)
631 (v6)                   DENY        Anywhere (v6)

parkadmin@parkfinder-virtual-machine:~$ sudo nano /etc/ufw/before.rules
parkadmin@parkfinder-virtual-machine:~$ sudo ufw reload
Firewall reloaded
```

On aura juste à update les règles au niveau du fichier /etc/ufw/before.rules et reload le FireWall pour appliquer les changements.

2.2.4.2- Monitoring

Au niveau du monitoring on a utilisé Grafana et Prometheus pour superviser en temps réels les ressources matérielles et le traffic réseau sur notre serveur parkfinder.

Ce dernier nous permettra d'établir des règles pour générer des alertes a l'administrateur en cas de problèmes.