Cyber Défense des infrastructures.

Start date : 10/12/2022

Duration: 30 days

For the Company: POWERZIO

From the team : “C'est down ou quoi ?”

Création et update du DNS, installation de bind9 etc…

Créations des différentes zones

```
#!/bin/bash

#Updates the system
sudo apt update -y && apt-get upgrade -y

#Installs dns bind9
sudo apt install bind9 bind9utils -y

#Set the domain name for the docker instance
echo "powerzio.net" | sudo tee "/etc/hostname" > /dev/null

#Add the localhost with the domain name
echo "127.0.0.1 powerzio.net" | sudo tee -a "/etc/hosts" > /dev/null

cat ./configs/db.powerzio.net | sudo tee "/etc/bind/db.powerzio.net" > /dev/null
cat ./configs/db.10 | sudo tee "/etc/bind/db.10" > /dev/null

#Dns principal
echo \
 "zone \"powerzio.net\" {\
  type master;\
  file \"/etc/bind/db.powerzio.net\";\
};" | sudo tee -a "/etc/bind/named.conf.local"

#Reverse dns
echo \
  "zone \"0.10.10.in-addr.arpa\" {\
  type master;\
  notify no;\
  file \"/etc/bind/db.10\";\
};" | sudo tee -a "/etc/bind/named.conf.local"

sudo systemctl restart bind9
sudo systemctl enable bind9
```

Reverse DNS

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.powerzio.net. root.powerzio.net. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.powerzio.net.
10      IN      PTR     ns.powerzio.net.
```

Création des sous domaines et liens entre IP et domaines

```
$TTL    604800
@       IN      SOA     ns.powerzio.net. root.powerzio.net. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )    ; Negative Cache TTL
;
@               IN      NS      ns.powerzio.net.
f               IN      CNAME   files.powzezio.net
beta-f          IN      CNAME   beta.files.powerzio.net
g               IN      A       8.8.8.8
bank            IN      A       10.10.0.6
files           IN      A       10.10.0.6
beta.files      IN      A       10.10.0.6
monitoring      IN      A       10.10.0.6
dev             IN      TXT     "hello world"
```

Initialisation de Docker et mise en place :

```
#!/bin/bash

#Update the system
sudo apt update -y
sudo apt upgrade -y

#Install the docker dependencies
sudo apt-get install \
     ca-certificates \
     curl \
     gnupg \
     lsb-release

#Add Docker official GPG key
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

#Setup Docker repo
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

#Install docker engine
sudo apt-get update -y
sudo apt-get install docker-ce docker-ce-cli docker-compose containerd.io docker-compose-plugin -y
sudo systemctl enable docker
```

Création d'un script pour installer le packet pour monitorer le system.

```
#!/bin/bash
#doc https://linuxhandbook.com/cockpit/

#Update system
sudo apt update -y
sudo apt upgrade -y

#Install cockpit monitoring dashboard
sudo apt -y install cockpit

#Access dashboard on localhost:9090
```

Script qui permet de créer un antiDDoS

```
#!/bin/bash

#Block invalid packets
iptables -t mangle -A PREROUTING -m conntrack --ctstate INVALID -j DROP

#Block new packets that are not SYN
iptables -t mangle -A PREROUTING -p tcp ! --syn -m conntrack --ctstate NEW -j DROP

#Block uncommon MSS values
iptables -t mangle -A PREROUTING -p tcp -m conntrack --ctstate NEW -m tcpmss ! --mss 536:65535 -j DROP

#Block packets with bogus flags
iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,SYN FIN,SYN -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags FIN,ACK FIN -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags ACK,URG URG -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags ACK,PSH PSH -j DROP
iptables -t mangle -A PREROUTING -p tcp --tcp-flags ALL NONE -j DROP

#Bloquer les connexions qui ont plus de X connexions
iptables -A INPUT -p tcp -m connlimit --connlimit-above 100 -j REJECT --reject-with tcp-reset

#Limite les requêtes par client
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m limit --limit 60/s --limit-burst 30 -j ACCEPT
iptables -A INPUT -p tcp -m conntrack --ctstate NEW -j DROP

#Block fragmentaed packets
iptables -t mangle -A PREROUTING -f -j DROP

### SSH brute-force protection ###
iptables -A INPUT -p tcp --dport ssh -m conntrack --ctstate NEW -m recent --set
iptables -A INPUT -p tcp --dport ssh -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 10 -j DROP

### Protection against port scanning ###
iptables -N port-scanning
iptables -A port-scanning -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 1/s --limit-burst 2 -j RETURN
iptables -A port-scanning -j DROP
```

Il bloque les paquets invalide, ce qui ne sont pas du SYN, les paquets qui son flag « bogus », les connexions répété, limite les requets par clients, bloque les paquets fragmenter et empêche le brut force.

Mise en place du pont VPN

```
#!/bin/bash

#Update system
sudo apt update -y
sudo apt upgrade -y

#Install wiregard vpn
sudo apt install wireguard -y

#Set key to the wireguard path via symlink
sudo cp ./wg0.conf /etc/wireguard/wg0.conf

#Setup new network interface for vpn biding
sudo wg-quick up wg0

#start the vpn service at startup and start it now
sudo systemctl start wg-quick@wg0.service
sudo systemctl enable wg-quick@wg0.service
```

Configuration du VPN:

```
[Interface]
PrivateKey = qNWzx0Evm4RHt3Nmmkh8lTrVzKGINt3xJYE0Vv8Ghm8=
Address = 10.10.0.6/24

[Peer]
PublicKey = ic4d58tu1WSm7gtVOI0h9d5DIQ3pf3GpugbF0aG/cCM=
AllowedIPs = 10.10.0.0/16
Endpoint = 95.179.143.25:41194
PersistentKeepalive = 15
```

Infra docker pour le site bank.powerzio.net

```yaml
version: '3.3'

services:

  bank:
    build:
      context: bank
    volumes:
      - ./bank/db.sqlite:/app/db.sqlite
    restart: always

  proxy:
    build:
      context: nginx
    volumes:
      - ./nginx/files:/var/project-files
    ports:
      - "80:80"
      - "443:443"
    restart: always
```

Script de backup pour la base de donnée de la banque

```bash
#!/bin/bash

SCRIPT=$(readlink -f "$0")
SCRIPT_DIR=$(dirname "$SCRIPT")
BACKUP_NAME="$(date +'%d-%m-%Y_%H:%M:%S')"

cp "$SCRIPT_DIR/bank/db.sqlite" "$SCRIPT_DIR/bank/backups/bank_db_$BACKUP_NAME.sqlite"
```

Image docker pour Nginx

```
#Docker image for proxy for the accesible apps
FROM debian

RUN apt-get update -y && apt-get install nginx -y

ADD certification/1670684367.crt /etc/cert/powerzio/1670684367.crt
ADD certification/super_key.csr /etc/cert/powerzio/super_key.csr

ADD ./configs/bank.powerzio.net /etc/nginx/sites-available/bank.powerzio.net
ADD ./configs/bank.powerzio.net /etc/nginx/sites-enabled/bank.powerzio.net

ADD ./configs/files.powerzio.net /etc/nginx/sites-available/files.powerzio.net
ADD ./configs/files.powerzio.net /etc/nginx/sites-enabled/files.powerzio.net

RUN mkdir -p /var/project-files
RUN chown -R www-data:www-data /var/project-files
RUN chmod 755 /var/project-files

EXPOSE 80
EXPOSE 443

STOPSIGNAL SIGQUIT

CMD ["nginx", "-g", "daemon off;"]
```

Config nginx pour bank

```
server {
    listen 80;
    server_name bank.powerzio.net;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name bank.powerzio.net;

    index index.html index.htm;

    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $http_host;
        proxy_pass http://bank:5000;
        proxy_redirect off;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }

    ssl_certificate /etc/cert/powerzio/1670684367.crt;
    ssl_certificate_key /etc/cert/powerzio/super_key.csr;
}
```

Image docker pour bank (python)

```
#Docker image for bank app
FROM python:3-alpine

WORKDIR /app

COPY . .

RUN python -m pip install --upgrade pip

RUN pip install flask jwt

#RUN pip install flask pyjwt pysqlite3

CMD ["python3", "./server.py"]

EXPOSE 5000
```