Annex 1: Instal·lacions i configuracions WAF's i eines

Instal·lació Nginx proxy server i WAF ModSecurity amb OWASP Core Rule Set

Com a primera tasca, es realitza la desinstal·lació de la versió Nginx que proporciona la VM predefinida de Dojo.

```
apt purge nginx nginx-common
 apt autoremove

apt remove --purge nginx*
apt autoremove
apt update
```

A continuació, s'instal·len les dependències necessàries per Nginx

```
apt install make gcc build-essential autoconf automake libtool libfuzzy-dev ssdeep gettext pkg-config libcurl4-openssl-dev liblua5.3-dev libpcre3 libpcre3-dev libxml2 libxml2-dev libyajl-dev doxygen libcurl4 libgeoip-dev libssl-dev zlib1g-dev libxslt-dev liblmdb-dev libpcre++-dev libgd-dev
```

S'afegeix el repositori que conté el Nginx adequat, s'actualitza la llista de paquets i es realitza la instal·lació

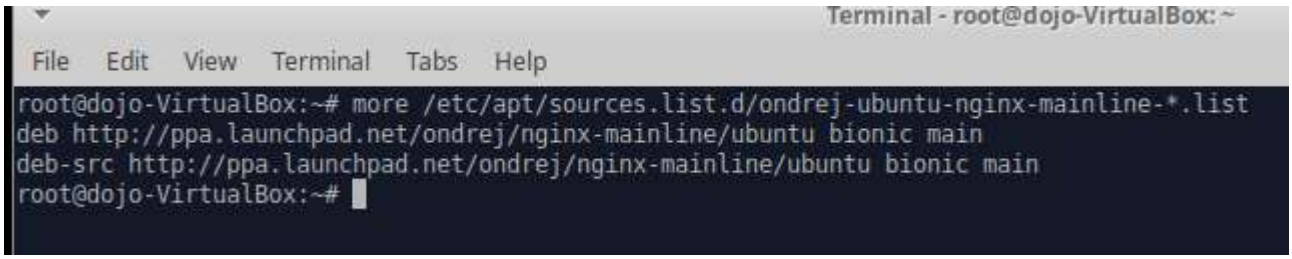
```
add-apt-repository ppa:ondrej/nginx-mainline -y

apt update
apt install nginx-core nginx-common nginx nginx-full
```

S'activa la possibilitat d'accedir al repositori de codi font del Nginx

```
vim /etc/apt/sources.list.d/ondrej-ubuntu-nginx-mainline-*.list
Locate and uncomment this line to enable the source code repository:

# deb-src http://ppa.launchpad.net/ondrej/nginx-mainline/ubuntu/ focal main
```



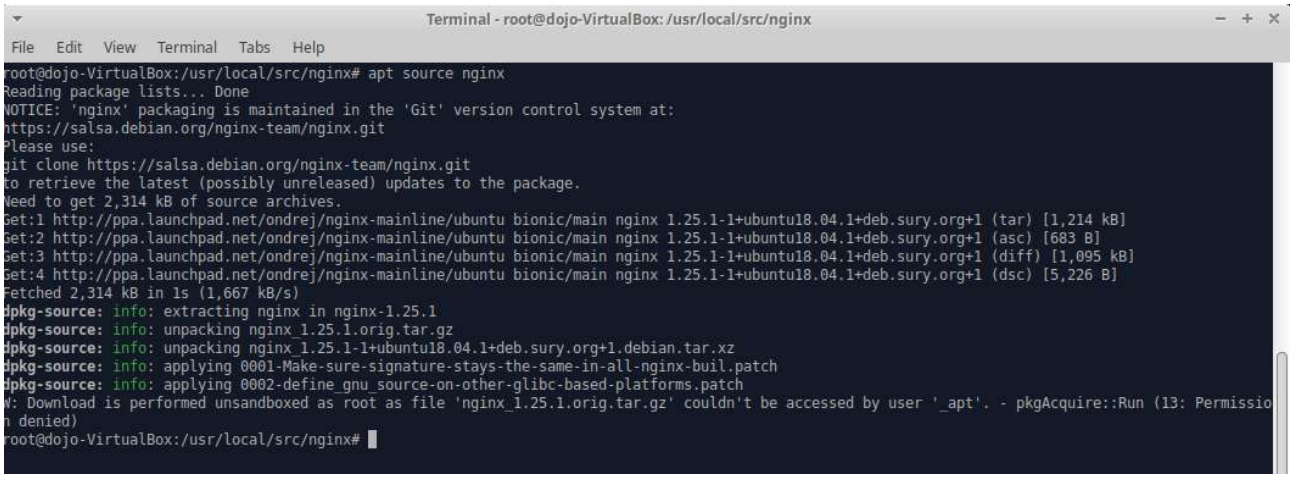
I s'actualitza el package index.

```
apt update
```

Per compilar el mòdul dinàmic de Modsecurity, cal descarregar el codi font de Nginx. Es crea un directori per allotjar aquest codi font i es descarrega en aquest directori mitjançant comanda apt i es verifica la versió instal·lada.

```
mkdir -p /usr/local/src/nginx
 cd /usr/local/src/nginx

apt source nginx
```



A continuació, s'instal·la la llibreria Libmodsecurity que gestiona el filtrat de les peticions HTTP. Es baixa la versió específica mitjançant comanda wget.

```
git clone --depth 1 -b v3/master --single-branch https://github.com/SpiderLabs/ModSecurity
/usr/local/src/ModSecurity/
```

Des de el directori clonat, s'instal·len els submòduls.

```
cd /usr/local/src/ModSecurity/

git submodule init
git submodule update
```

```
root@dojo-VirtualBox:~# cd /usr/local/src/ModSecurity/
root@dojo-VirtualBox:/usr/local/src/ModSecurity# git submodule init
Submodule 'bindings/python' (https://github.com/SpiderLabs/ModSecurity-Python-bindings.git) registered for path 'bindings/python'
Submodule 'others/libinjection' (https://github.com/libinjection/libinjection.git) registered for path 'others/libinjection'
Submodule 'test/test-cases/secrules-language-tests' (https://github.com/SpiderLabs/secrules-language-tests) registered for path 'test/test-cases/secr
ules-language-tests'
root@dojo-VirtualBox:/usr/local/src/ModSecurity# git submodule update
Cloning into '/usr/local/src/ModSecurity/bindings/python'...
Cloning into '/usr/local/src/ModSecurity/others/libinjection'...
Cloning into '/usr/local/src/ModSecurity/test/test-cases/secrules-language-tests'...
Submodule path 'bindings/python': checked out 'bc625d5bb0bac6a64bcce8dc9902208612399348'
Submodule path 'others/libinjection': checked out 'bfba51f5af8f1f6cf5d6c4bf862f1e2474e018e3'
Submodule path 'test/test-cases/secrules-language-tests': checked out 'a3d4405e5a2c90488c387e589c5534974575e35b'
root@dojo-VirtualBox:/usr/local/src/ModSecurity#
```

I es crea l'entorn a partir de les següents comandes.

```
./build.sh
./configure


Nota:
Ignore the error displayed below.
fatal: No names found, cannot describe anything.
```

Es compila el codi font i s'instal·len altres utilitats amb comanda make. Després s'instal·len les llibreries.

```
$ sudo make -j4

$ sudo make install
```

La següent tasca es baixar i compilar el conector de Modsecurity amb Nginx (ModSecurity v3 Nginx Connector). Aquest connector permet entrellaçar la llibrería LibModSecurity amb el servidor web Nginx.

```
git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git /usr/local/src/ModSecurity-
nginx/
```

```
root@dojo-VirtualBox:/usr/local/src/ModSecurity# git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git /usr/local/src/ModSecurity-nginx/
Cloning into '/usr/local/src/ModSecurity-nginx'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 40 (delta 11), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (40/40), done.
root@dojo-VirtualBox:/usr/local/src/ModSecurity#
```

Des de el directori clonat, es construeixen les dependències.

```
cd /usr/local/src/nginx/nginx-1.21.3/


apt build-dep nginx
apt install uuid-dev
```

Es compila el modul connector amb l'opció --with compat per que aquest sigui compatible amb la llibreria Nginx descarregada. A continuació es construeix el modul amb comanda "make" i es copia l'objecte creat al directori dels moduls de Nginx.

```
./configure --with-compat --add-dynamic-module=/usr/local/src/ModSecurity-nginx

make modules
cp objs/ngx_http_modsecurity_module.so /usr/share/nginx/modules/
```

S'accedeix a la configuració de Nginx per carregar el modul connector acabat de crear. A més, s'incorpora en la section http{...} les linies que activen Modsecurity per tots els virtual hosts definits a Nginx.

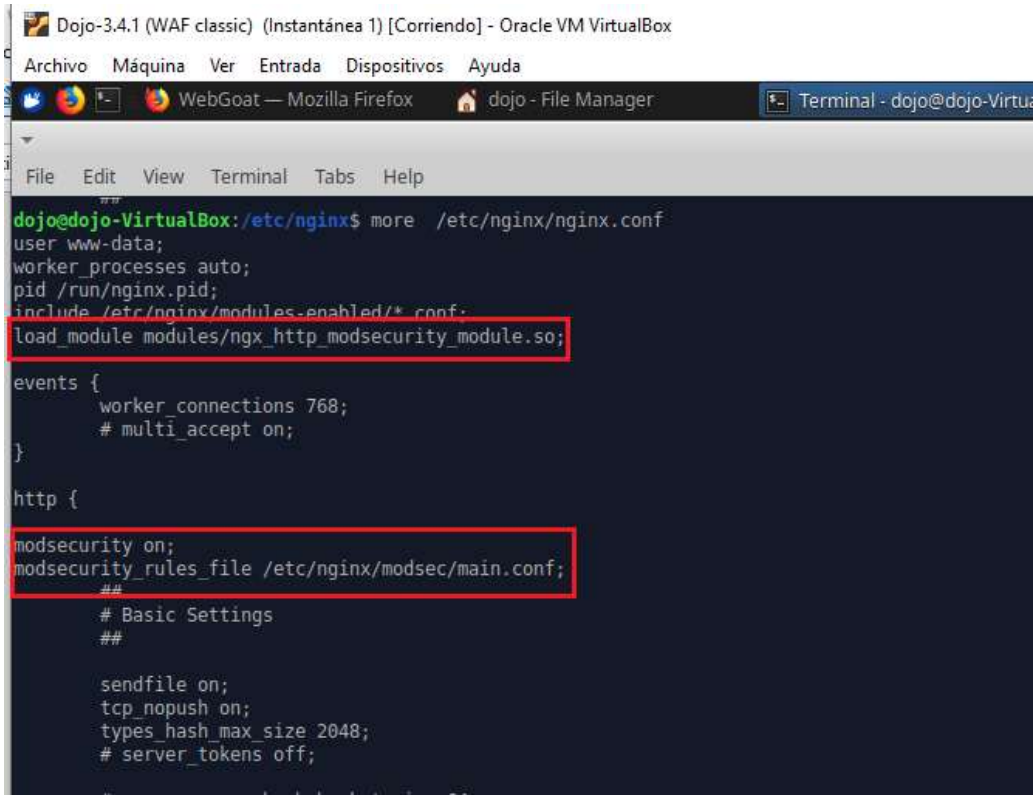
```
 vim /etc/nginx/nginx.conf

#Append the following line just below the first few lines

load_module modules/ngx_http_modsecurity_module.so;

#In addition, append the following lines in the http {...} section. This enables ModSecurity for
#all Nginx virtual hosts.

modsecurity on;
modsecurity_rules_file /etc/nginx/modsec/main.conf;
```



A continuació es crea el directori que contindrà la configuració per ModSecurity i es copia la configuració recomanada en aquest directori. Es varia la directiva "SecRuleEngine" per que bloquegi els atacs web.

```
# Next up, create the /etc/nginx/modsec/ directory which will store ModSecurity configuration.

mkdir /etc/nginx/modsec/

#Next, copy the ModSecurity configuration file as follows.

cp /usr/local/src/ModSecurity/modsecurity.conf-recommended /etc/nginx/modsec/modsecurity.conf

#Then open the configuration file.
```

```
vim /etc/nginx/modsec/modsecurity.conf
#Locate the line beginning with the SecRuleEngine directive.

SecRuleEngine DetectionOnly

#Change the line to the line below

SecRuleEngine On
```

Es crea el fitxer main.conf i s'afegeix la referencia del fitxer modsecurity.conf

```
vim  /etc/nginx/modsec/main.conf

#Append this line to reference the /etc/nginx/modsec/modsecurity.conf configuration file.

Include /etc/nginx/modsec/modsecurity.conf
```

Es copia el fitxer de mapeig d'Unicode i es testeja la que la configuració de Nginx sigui correcta.

```
cp /usr/local/src/ModSecurity/unicode.mapping /etc/nginx/modsec/

#Then test Nginx configuration.

nginx -t
```

```
root@dojo-VirtualBox:/usr/local/src/nginx/nginx-1.25.1# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@dojo-VirtualBox:/usr/local/src/nginx/nginx-1.25.1#
```

Es configura el servidor web Nginx perque escolti pel port 8888 i derivi les peticions a les aplicacions que calgui en cada cas.

Al fitxer `/etc/nginx/sites-enabled/default` es canvïa

```
server {
    listen       80;
    server_name  localhost;

    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
```

per

```
server {
    listen       8888;
    server_name  localhost;

    #access_log  /var/log/nginx/host.access.log  main;
```

```
   location / {
      root   /usr/share/nginx/html;
     index  index.html index.htm;
   }
location /WebGoat { proxy_pass http://webgoat.local:8081/WebGoat; }
```

Es re-arrenca el Nginx per aplicar tots els canvis efectuats.

```
systemctl restart nginx
```

Com a tasca final, previa al testeig final, s'incorporen les regles del OWASP Corerule Set. Es baixen de Github a partir de comanda "wget".

```
wget https://github.com/coreruleset/coreruleset/archive/v3.3.5.tar.gz
```

Es descomprimeix el gzip i es mou el directori al path adient. S'usa el exemple del crs-setup com a fitxer inicial de configuració de regles.

```
tar xvf v3.3.5.tar.gz

#Ensure to move the uncompressed directory to the /etc/nginx/modsec/ path.

mv coreruleset-3.3.5/ /etc/nginx/modsec/


#Then rename the crs-setup.conf.example file to crs-setup.conf.

mv /etc/nginx/modsec/coreruleset-3.3.5/crs-setup.conf.example
/etc/nginx/modsec/coreruleset-3.3.5/crs-setup.conf
```

Es configura el main.conf del servidor Nginx incorporant els directoris/fitxers de regles.

```
vim /etc/nginx/modsec/main.conf
# And append the following lines.

Include /etc/nginx/modsec/coreruleset-3.3.5/crs-setup.conf
Include /etc/nginx/modsec/coreruleset-3.3.5/rules/*.conf
```

Es re-arrenca el Nginx per aplicar tots els canvis efectuats.

```
systemctl restart nginx
```

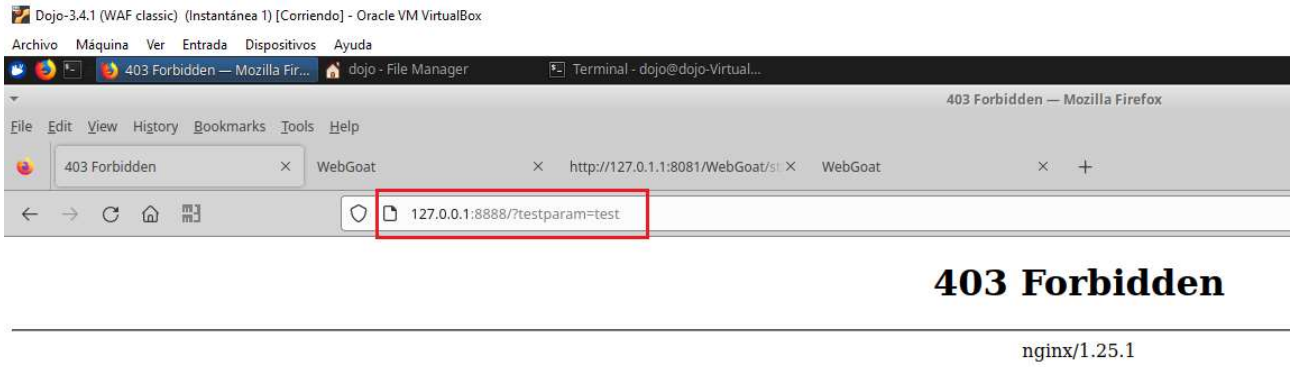
Per comprovar el correcte funcionament de Modsecurity, confirmant que detecta i bloqueja el transit sospitós, es crea una regla que bloqueja una URL concreta quan s'accedeix a través del navegador.

```
#vim /etc/nginx/modsec/modsecurity.conf
Add this line just below the SecRuleEngine On directive

SecRule ARGS:testparam "@contains test" "id:254,deny,status:403,msg:'Test Successful'"
```

S'accedeix a la URL

```
http://127.0.0.1:8888/?testparam=test
```



I es comprova també en el log del Nginx

Referència: https://www.tecmint.com/install-modsecurity-nginx-debian-ubuntu/

Instal·lació Nginx proxy server i WAF open-appsec

El WAF open-appsec requereix de una versió de Nginx 1.22 o superior.

Per això, es baixa la versió específica mitjançant comanda wget:

```
wget https://nginx.org/packages/ubuntu/pool/nginx/n/nginx/nginx_1.22.0-1~bionic_amd64.deb
```

A continuació, s'instal.la la versió baixada:

```
sudo dpkg -i nginx_1.22.0-1~bionic_amd64.deb
```

Es configura el servidor web Nginx perque escolti pel port 8888 i derivi les peticions a les aplicacions que calgui en cada cas.
Al fitxer /etc/nginx/conf.d/default.conf es canvïa

```
server {
    listen       80;
    server_name  localhost;

    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
```

per

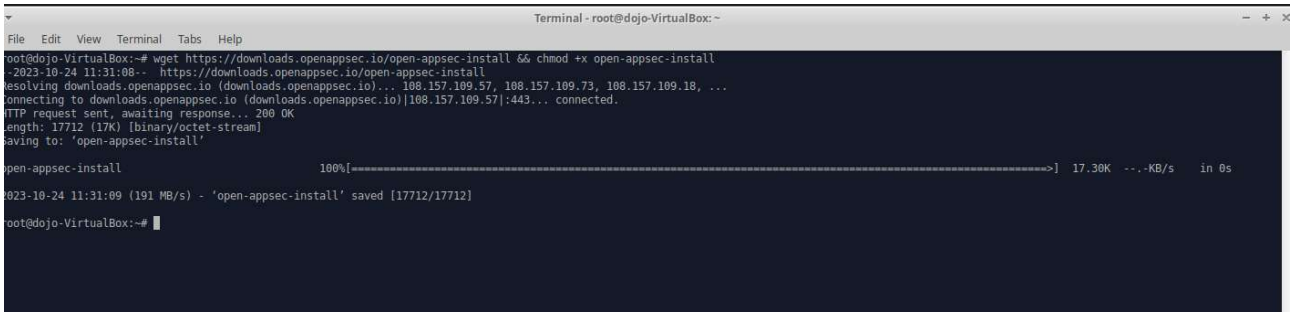
```
server {
    listen       8888;
    server_name  localhost;

    #access_log  /var/log/nginx/host.access.log  main;

   location / {
      root   /usr/share/nginx/html;
     index  index.html index.htm;
   }
location /WebGoat { proxy_pass http://webgoat.local:8081/WebGoat; }
```

Posteriorment, es realitza la instalació del WAF open-appsec. Es baixa mitjançant comanda wget.

```
wget https://downloads.openappsec.io/open-appsec-install && chmod +x open-appsec-install
```



I, s'instal·la el WAF integrant-se automaticament amb el servidor Nginx.

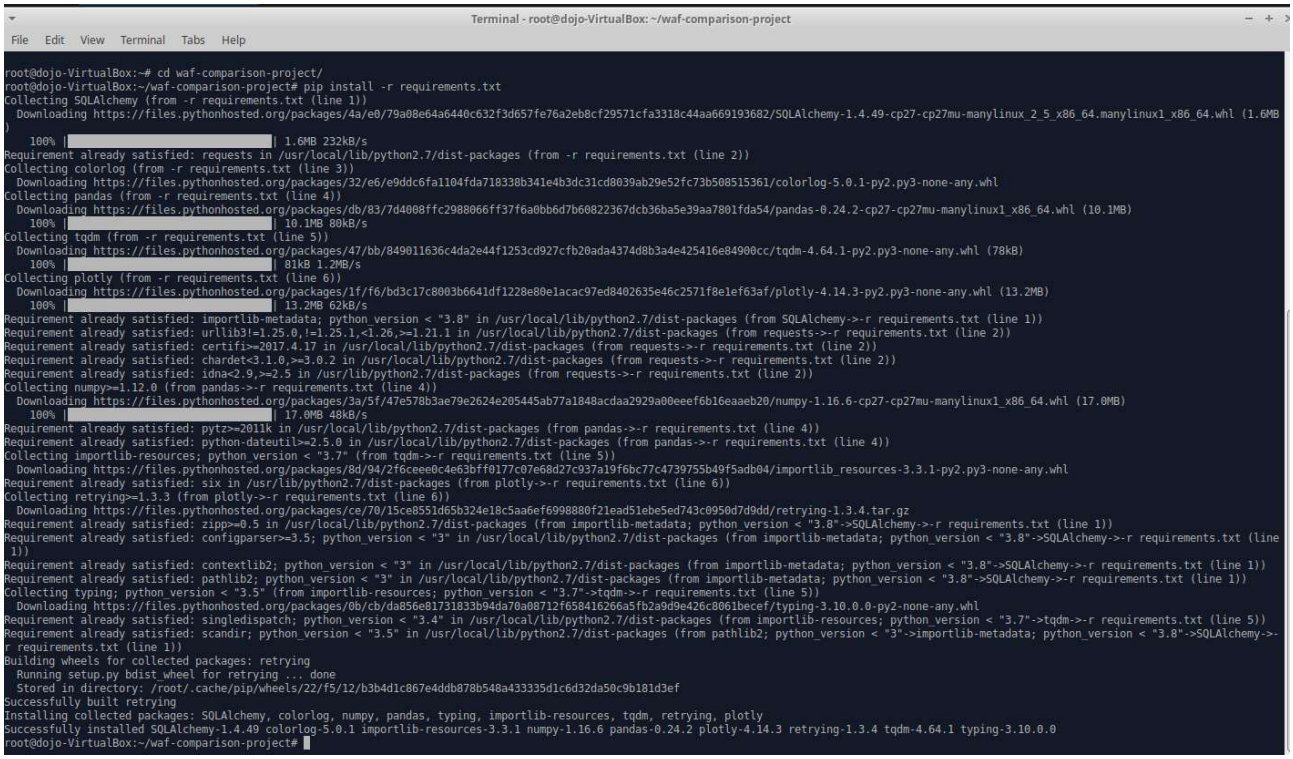
```
./open-appsec-install --auto --prevent
```

Referència: https://medium.com/@bhojport/installing-a-specific-version-of-nginx-on-ubuntu-or-any-other-platform-4fed8e859534

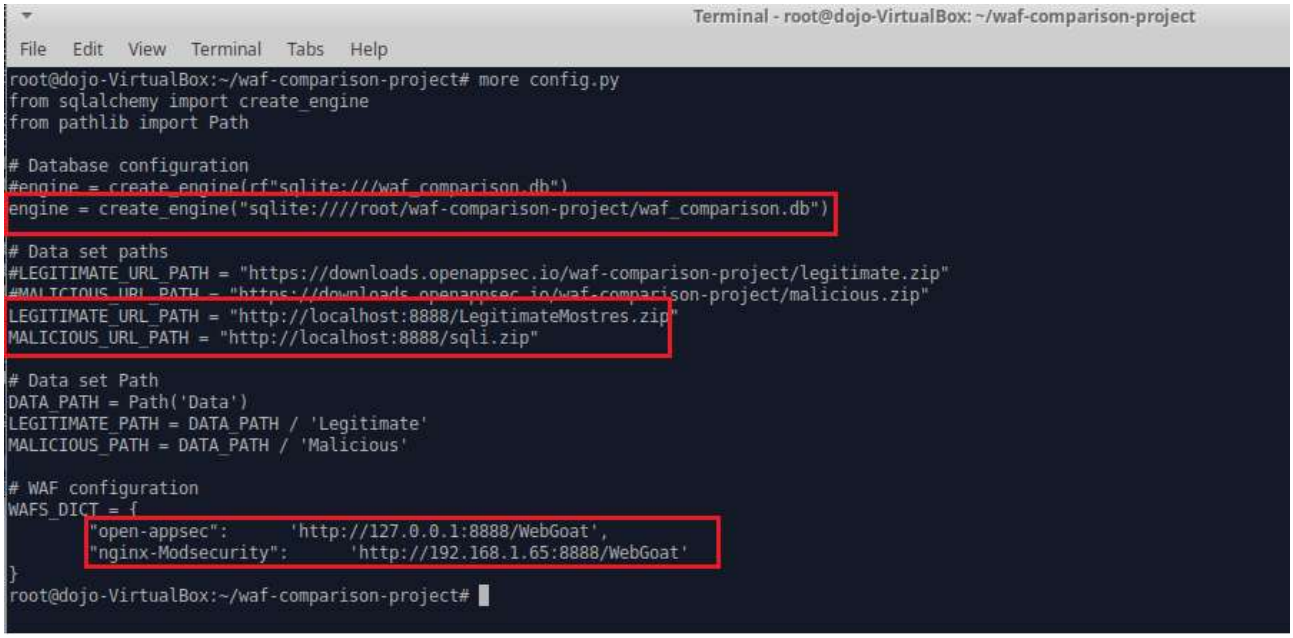
Instal·lació eina waf_comparison_project

En primera instancia, es baixa el projecte desde Github mitjançant comanda "git clone":

Posteriorment, es realitza la instal·lació dels requeriments de Python necessaris per l'eina.



La configuració es realitza en el fitxer config.py. S'especifica el engine de base de dades, les ubicacions dels payloads i les URL's dels WAF's en estudi.


Terminal - root@dojo-VirtualBox: ~/waf-comparison-project
File Edit View Terminal Tabs Help
root@dojo-VirtualBox:~/waf-comparison-project# more config.py
from sqlalchemy import create_engine
from pathlib import Path
# Database configuration
#engine = create_engine(rf"sqlite:///waf_comparison.db")
engine = create_engine("sqlite:////root/waf-comparison-project/waf_comparison.db")
# Data set paths
#LEGITIMATE_URL_PATH = "https://downloads.openappsec.io/waf-comparison-project/legitimate.zip"
#MALICIOUS_URL_PATH = "https://downloads.openappsec.io/waf-comparison-project/malicious.zip"
LEGITIMATE_URL_PATH = "http://localhost:8888/LegitimateMostres.zip"
MALICIOUS_URL_PATH = "http://localhost:8888/sqli.zip"
# Data set Path
DATA_PATH = Path('Data')
LEGITIMATE_PATH = DATA_PATH / 'Legitimate'
MALICIOUS_PATH = DATA_PATH / 'Malicious'
# WAF configuration
WAFS_DICT = {
"open-appsec": 'http://127.0.0.1:8888/WebGoat',
"nginx-Modsecurity": 'http://192.168.1.65:8888/WebGoat'
}
root@dojo-VirtualBox:~/waf-comparison-project#

Instal·lació eina WafNinja

Per la baixada i instalació de l'eina, es clona mitjançant comanda git, i s'executen les dependències de Python.

```
git clone https://github.com/khalilbijjou/WAFNinja && cd WAFNinja
pip install -r requirements.txt
```