PUESTA EN PRODUCCIÓN SEGURA

Unidad 4. Actividad 33

01 DE ABRIL DE 2024

CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Enunciado

Actividad. Configure en un firewall WAF en un servidor Apache, en instalación básica de Apache sin modificación. Instalar el juego de reglas de OWASP. Capture todos los pasos.

Ejercicio.

Primero vamos a configurar el firewall WAF, para esto vamos a instalarnos el mod-security2:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install libapache2-mod-security2
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  gcc-12-base libarmadillo11 libcodec2-1.1 libgcc-12-dev libgumbo1 libgupnp-igd-1.0-4 libjim0.81 libmagickcore-6.q16-6
  libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libnfs13 libobjc-12-dev libstdc++-12-dev libtexluajit2 python3-jdcal
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  liblua5.1-0 modsecurity-crs
Suggested packages:
  lua geoip-database-contrib python
The following NEW packages will be installed:
  libapache2-mod-security2 liblua5.1-0 modsecurity-crs
0 upgraded, 3 newly installed, 0 to remove and 1630 not upgraded.
Need to get 531 kB of archives.
After this operation, 2,458 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Ahora vamos a hacer una copia del archivo de seguridad:

```
┌──(kali㉿kali)-[~]
└─$ sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
```

Ahora editamos el archivo:

```
  GNU nano 7.2                                  /etc/modsecurity/modsecurity.conf *
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On
```

Hago un git clone:

```
┌──(kali㉿kali)-[~]
└─$ sudo git clone https://github.com/Spiderlabs/owasp-modsecurity-crs.git /usr/share/modsecurity-crs
Cloning into '/usr/share/modsecurity-crs' ...
remote: Enumerating objects: 10486, done.
remote: Total 10486 (delta 0), reused 0 (delta 0), pack-reused 10486
Receiving objects: 100% (10486/10486), 3.33 MiB | 6.25 MiB/s, done.
Resolving deltas: 100% (7687/7687), done.
```

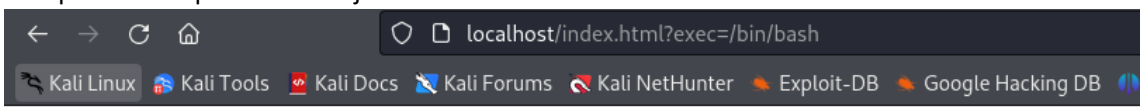
Ahora editamos el fichero security2.conf con el fin de que nos proteja ante intentos de hackeo. Nos debería quedar algo asi:

```
 GNU nano 7.2                          /etc/apache2/mods-enabled/security2.conf *
<IfModule security2_module>
        # Default Debian dir for modsecurity's persistent data
        SecDataDir /var/cache/modsecurity

        # Include all the *.conf files in /etc/modsecurity.
        # Keeping your local configuration in that directory
        # will allow for an easy upgrade of THIS file and
        # make your life easier
        IncludeOptional /etc/modsecurity/*.conf

        # Include OWASP ModSecurity CRS rules if installed
        IncludeOptional /usr/share/modsecurity-crs/*.load
</IfModule>
        IncludeOptional /usr/share/modsecurity-crs/*.conf
        IncludeOptional /usr/share/modsecurity-crs/rules/*.conf
```

Comprobamos que no nos deja entrar:

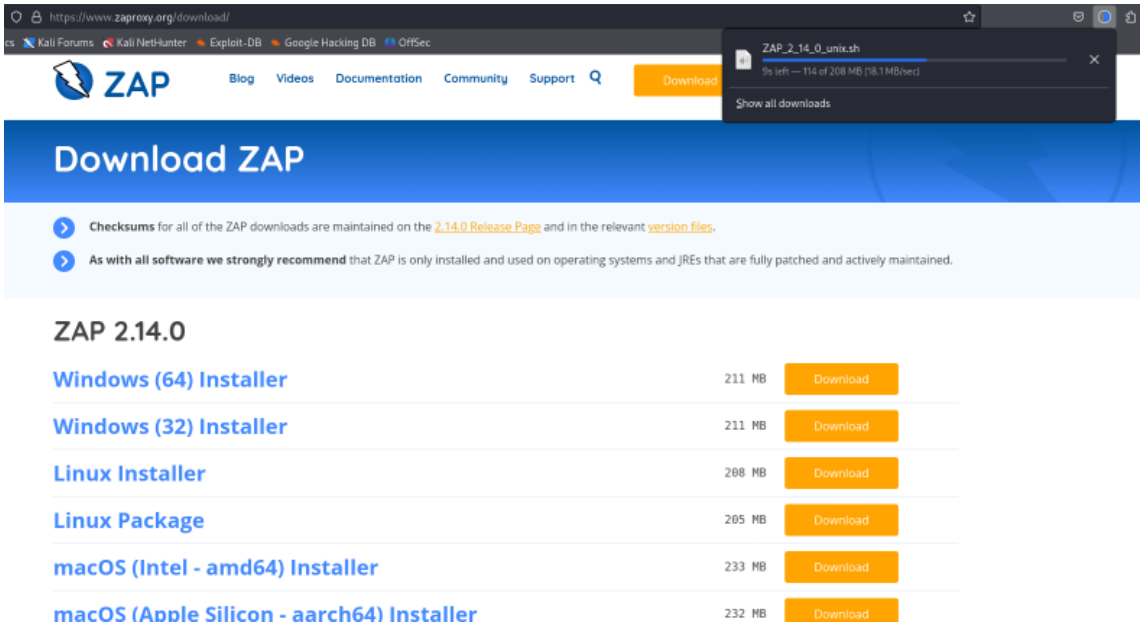


Forbidden

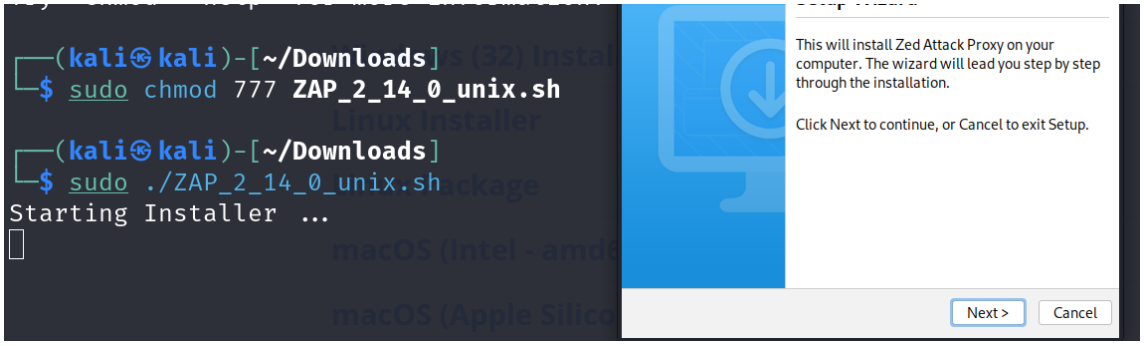
You don't have permission to access this resource.

Apache/2.4.58 (Debian) Server at localhost Port 80

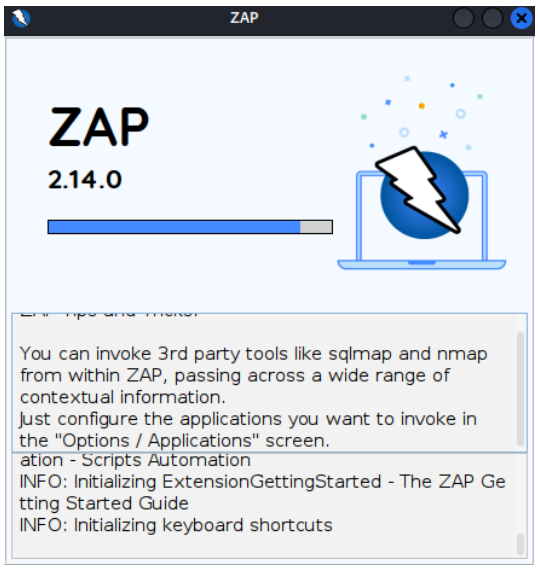
Instalamos OWASP:



Ahora simplemente ejecutas el sh descargado:



Ahora lo ejecutamos:



Comprobamos si funciona sobre nuestro servidor apache: