

Install and verify Docker Container based NGINX, MODSECURITY WAF, OWASP Core Rule Sets and SSL

Tested Operating System and Architecture,

```
root@ip-172-31-3-97:~# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
root@ip-172-31-3-97:~# uname -m
x86_64
root@ip-172-31-3-97:~#
```

Steps to Install docker,

apt update && apt upgrade -y

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

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

sudo apt update

apt-cache policy docker-ce

sudo apt install docker-ce -y

sudo systemctl status docker

docker container ls -a (Ensure no container exists)

```

root@ip-172-31-3-97:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-01-17 06:46:52 UTC; 1h 1min ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 15218 (dockerd)
     Tasks: 25
    Memory: 24.4M
       CPU: 2.159s
    CGroup: /system.slice/docker.service
            └─15218 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
              └─15732 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8443 -container-ip 172.17.0.2 -container-port 443
                └─15738 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 8443 -container-ip 172.17.0.2 -container-port 443

root@ip-172-31-3-97:~# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
root@ip-172-31-3-97:~#
root@ip-172-31-3-97:~#

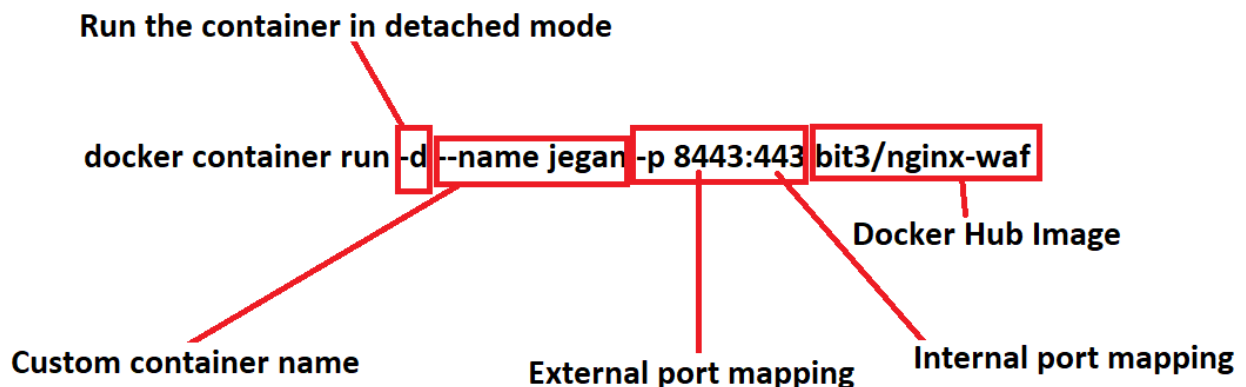
```

Find the below Docker Hub link to install NGINX OWASP MODSEC WAF CRS RULES container,

<https://hub.docker.com/r/bit3/nginx-waf>

Run the container image with the below command,

`docker container run -d --name jegan -p 8443:443 bit3/nginx-waf`



```

root@ip-172-31-3-97:~# docker container run -d --name jegan -p 8443:443 bit3/nginx-waf
Unable to find image 'bit3/nginx-waf:latest' locally
latest: Pulling from bit3/nginx-waf
7a6db449b51b: Pull complete
ca1981974b58: Pull complete
d4019c921e20: Pull complete
7cb804d746d4: Pull complete
e7a561826262: Pull complete
7247f6e5c182: Pull complete
a904d8a502b0: Pull complete
Digest: sha256:cfdb6b022cc3831b0afalf2d1e5feeced649105b759fe92be1dacf0ce7f5defdd
Status: Downloaded newer image for bit3/nginx-waf:latest
e9072970dae271d3505329a1f4220b65f42235b26149c39429bb81ad6b4290b3
root@ip-172-31-3-97:~# docker container ls -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9072970dae2	bit3/nginx-waf	"/docker-entrypoint..."	7 seconds ago	Up 6 seconds	80/tcp, 0.0.0.0:8443->443/tcp, :::8443->443/tcp	jegan

```

root@ip-172-31-3-97:~#

```

Note : By default the above command will install all needed NGINX OWASP MODSEC WAF CRS RULES components.

Log Into the container for SSL (Self Signed certificate example) configuration with the below command,

`docker exec -it e9072970dae2 bash`

```

root@ip-172-31-3-97:~# docker container ls -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9072970dae2	bit3/nginx-waf	"/docker-entrypoint..."	13 minutes ago	Up 13 minutes	80/tcp, 0.0.0.0:8443->443/tcp, :::8443->443/tcp	jegan

`docker exec -it e9072970dae2 bash`

Container ID

Important directories in this container are /etc/nginx, /etc/nginx/modsec, /etc/nginx/modules, /etc/nginx/conf.d

All the modsecurity owasp crs rules will reside inside /usr/local/owasp-modsecurity-crs-3.2.0/rules/ directory

```

root@e9072970dae2:/etc/nginx# ls
conf.d fastcgi_params mime.types modsec modules nginx.conf scgi_params uwsgi_params
root@e9072970dae2:/etc/nginx#

```

```
root@e9072970dae2:/etc/nginx# cd modsec
root@e9072970dae2:/etc/nginx/modsec# ls
main.conf  modsecurity.conf  unicode.mapping
root@e9072970dae2:/etc/nginx/modsec#
```

```
root@e9072970dae2:/etc/nginx/modules# ls
ngx_http_geoip_module-debug.so      ngx_http_image_filter_module.so  ngx_http_modsecurity_module.so  ngx_stream_geoip_module-debug.so  ngx_stream_js_module.so
ngx_http_geoip_module.so            ngx_http_js_module-debug.so      ngx_http_xslt_filter_module-debug.so  ngx_stream_geoip_module.so        ngx_stream_js_module-debug.so
ngx_http_image_filter_module-debug.so  ngx_http_js_module.so            ngx_http_xslt_filter_module.so      ngx_stream_js_module-debug.so
root@e9072970dae2:/etc/nginx/modules#
```

```
root@e9072970dae2:/etc/nginx/conf.d# ls
default.conf
root@e9072970dae2:/etc/nginx/conf.d#
```

```
root@e9072970dae2:/etc/nginx/modsec# cd /usr/local/owasp-modsecurity-crs-3.2.0/
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0# ls
CHANGES  CONTRIBUTING.md  CONTRIBUTORS.md  INSTALL  KNOWN_BUGS  LICENSE  README.md  crs-setup.conf  crs-setup.conf.example  documentation  rules  util
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0#
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0# cd rules/
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0/rules#
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0/rules#
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0/rules# pwd
/usr/local/owasp-modsecurity-crs-3.2.0/rules
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0/rules# ls
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf      REQUEST-933-APPLICATION-ATTACK-PHP.conf          java-code-leakages.data
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example  REQUEST-934-APPLICATION-ATTACK-NODEJS.conf      java-errors.data
REQUEST-901-INITIALIZATION.conf                  REQUEST-941-APPLICATION-ATTACK-XSS.conf          lfi-os-files.data
REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf      REQUEST-942-APPLICATION-ATTACK-SQLI.conf        php-config-directives.data
REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.conf   REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf  php-errors.data
REQUEST-903.9003-NEXTCLOUD-EXCLUSION-RULES.conf   REQUEST-944-APPLICATION-ATTACK-JAVA.conf        php-function-names-933150.data
REQUEST-903.9004-DOKUWIKI-EXCLUSION-RULES.conf    REQUEST-949-BLOCKING-EVALUATION.conf            php-function-names-933151.data
REQUEST-903.9005-CPANEL-EXCLUSION-RULES.conf      RESPONSE-950-DATA-LEAKAGES.conf                php-variables.data
REQUEST-903.9006-XENFORO-EXCLUSION-RULES.conf     RESPONSE-951-DATA-LEAKAGES-SQL.conf             restricted-files.data
REQUEST-905-COMMON-EXCEPTIONS.conf                RESPONSE-952-DATA-LEAKAGES-JAVA.conf            restricted-upload.data
REQUEST-910-IP-REPUTATION.conf                   RESPONSE-953-DATA-LEAKAGES-PHP.conf             scanners-headers.data
REQUEST-911-METHOD-ENFORCEMENT.conf              RESPONSE-954-DATA-LEAKAGES-IIS.conf             scanners-urls.data
REQUEST-912-DOS-PROTECTION.conf                  RESPONSE-959-BLOCKING-EVALUATION.conf           scanners-user-agents.data
REQUEST-913-SCANNER-DETECTION.conf                RESPONSE-980-CORRELATION.conf                   scripting-user-agents.data
REQUEST-920-PROTOCOL-ENFORCEMENT.conf             RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf      sql-errors.data
REQUEST-921-PROTOCOL-ATTACK.conf                  RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example  unix-shell.data
REQUEST-930-APPLICATION-ATTACK-LFI.conf            crawlers-user-agents.data                      windows-powershell-commands.data
REQUEST-931-APPLICATION-ATTACK-RFI.conf            iis-errors.data
REQUEST-932-APPLICATION-ATTACK-RCE.conf            java-classes.data
root@e9072970dae2:/usr/local/owasp-modsecurity-crs-3.2.0/rules#
```

Module (load_module modules/ngx_http_modsecurity_module.so;) is the interface of NGINX and MODSEC WAF available in file /etc/nginx/nginx.conf

```
root@e9072970dae2:/etc/nginx# cat nginx.conf
load_module modules/ngx_http_modsecurity_module.so;
```

Install vim and nano (text editor) using the below commands inside the container,

```
apt update -y  
apt install vim nano -y
```

File /etc/nginx/modsec/modsecurity.conf represents the MODSEC WAF

File /etc/nginx/modsec/main.conf holds both rules files and MODSEC WAF file

File /etc/nginx/conf.d/default.conf binds SSL, MODSEC WAF and OWASP CRS RULES

From file /etc/nginx/conf.d/default.conf,

MODSEC WAF to 'ON'

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

This engages MODSEC WAF, OWASP CRS Rules with NGINX

Remove this line (`listen 80;`) and add the below lines in the file /etc/nginx/conf.d/default.conf,

```
listen 443 ssl default_server;  
listen [::]:443 ssl default_server;
```

```
server {  
    modsecurity on;  
    modsecurity_rules_file /etc/nginx/modsec/main.conf;  
    listen 80;
```

Remove this line

Create the certificate/key files `server.crt` and `server.key` inside /etc/nginx/ and add the below lines in the file /etc/nginx/conf.d/default.conf,

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt
```

```
ssl_certificate      /etc/nginx/server.crt;  
ssl_certificate_key  /etc/nginx/server.key;
```

```

root@e9072970dae2:/etc/nginx# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TN
Locality Name (eg, city) []:CBE
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ST21
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:15.207.55.167
Email Address []:opensourcejegan@gmail.com
root@e9072970dae2:/etc/nginx# ls
conf.d fastcgi_params mime.types modsec modules nginx.conf scgi_params server.crt server.key uwsgi_params
root@e9072970dae2:/etc/nginx#

```

```

server {
    modsecurity on;
    modsecurity rules file /etc/nginx/modsec/main.conf;
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;
    server_name localhost;

    ssl_certificate      /etc/nginx/server.crt;
    ssl_certificate_key  /etc/nginx/server.key;
}

```

Reload nginx with the below command,

`docker exec -it jegan nginx -s reload`

```

root@ip-172-31-3-97:~# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
e9072970dae2   bit3/nginx-waf "/docker-entrypoint..." About an hour ago Up About an hour 80/tcp, 0.0.0.0:8443->443/tcp, :::8443->443/tcp jegan
root@ip-172-31-3-97:~#
root@ip-172-31-3-97:~#
root@ip-172-31-3-97:~# docker exec -it jegan nginx -s reload
2023/01/17 09:06:53 [notice] 468#468: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/657/0)
2023/01/17 09:06:53 [notice] 468#468: signal process started
root@ip-172-31-3-97:~#

```

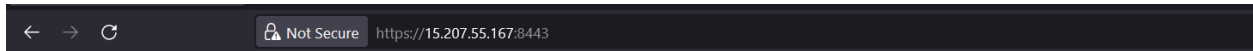
Container name

`docker exec -it jegan nginx -s reload`

Let's do the Testing of NGINX default page and with ATTACK URL like the below,

<https://15.207.55.167:8443>

Accept the exception, due to self signed server certificate, you will get the default NGINX page



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **15.207.55.167**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

Go Back (Recommended)

Advanced...

15.207.55.167:8443 uses an invalid security certificate.

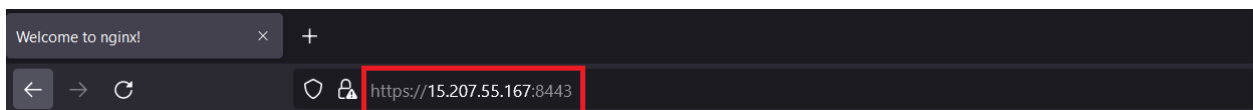
The certificate is not trusted because it is self-signed.

Error code: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[View Certificate](#)

Go Back (Recommended)

Accept the Risk and Continue



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

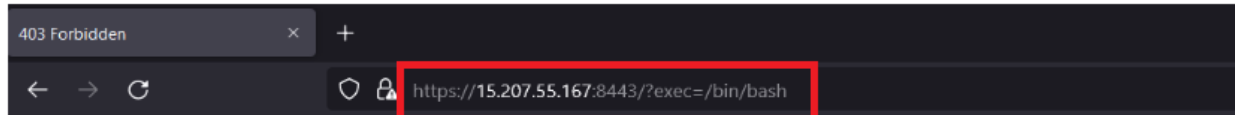
Thank you for using nginx.

Note : In the file `/etc/nginx/conf.d/default.conf`, if you replace the `server.crt` with Godaddy or any Public Certificate authority signed `server.crt`, you will not get the exception, you will get a green lock.

Execute ATTACK URL => `https://15.207.55.167:8443/?exec=/bin/bash` in browser and check the container log using `'tail -f /var/log/modsec_audit.log'` command

https://15.207.55.167:8443/?exec=/bin/bash

Attack



403 Forbidden

nginx/1.23.1

```
---Vhgjbxo5---H--
ModSecurity: Warning. Matched "Operator 'Rx' with parameter '^[\d.:]+$' against variable 'REQUEST_HEADERS:Host' (Value: '15.207.55.167:8443' ) [file "/usr/local/owasp-modsecurity-c
rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "678"] [id "920350"] [rev "" ] [msg "Host header is a numeric IP address"] [data "15.207.55.167:8443"] [severity "4"] [ver "OWASP
0"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-protocol"] [tag "OWASP_CRS"] [tag "OWASP_CRS/PROTOCOL_VIOLATI
T"] [tag "WASCTC/WASC-21"] [tag "OWASP_TOP_10/A7"] [tag "PCI/6.5.10"] [hostname "172.17.0.2"] [uri "/" ] [unique_id "1673947154"] [ref "o0,18v36,18"]
ModSecurity: Warning. Matched "Operator 'PmFromFile' with parameter 'unix-shell.data' against variable 'ARGS:exec' (Value: '/bin/bash' ) [file "/usr/local/owasp-modsecurity-crs-3.2
REQUEST-932-APPLICATION-ATTACK-RCE.conf"] [line "488"] [id "932160"] [rev "" ] [msg "Remote Command Execution: Unix Shell Code Found"] [data "Matched Data: bin/bash found within ARG
bin/bash"] [severity "2"] [ver "OWASP_CRS/3.2.0"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-shell"] [tag "platform-unix"] [tag "attack-rce"] [tag "OWAS
tag "OWASP_CRS/WEB_ATTACK/CMD_INJECTION"] [tag "WASCTC/WASC-31"] [tag "OWASP_TOP_10/A1"] [tag "PCI/6.5.2"] [hostname "172.17.0.2"] [uri "/" ] [unique_id "1673947154"] [ref "o1,8
lDecodeUni,t:cmdLine,t:normalizePath,t:lowercase"]
ModSecurity: Access denied with code 403 (phase 2). Matched "Operator 'Ge' with parameter '5' against variable 'TX:ANOMALY_SCORE' (Value: '8' ) [file "/usr/local/owasp-modsecurity-
/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "79"] [id "949110"] [rev "" ] [msg "Inbound Anomaly Score Exceeded (Total Score: 8)"] [data "" ] [severity "2"] [ver "" ] [maturity
curacy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "172.17.0.2"] [uri "/" ] [unique_id "1673947154"] [ref "" ]
```

You will get 403 forbidden page in the browser and logs for the same at container level

Done!!!

