WAF Setup on Ubuntu

Yehia A. Mostafa

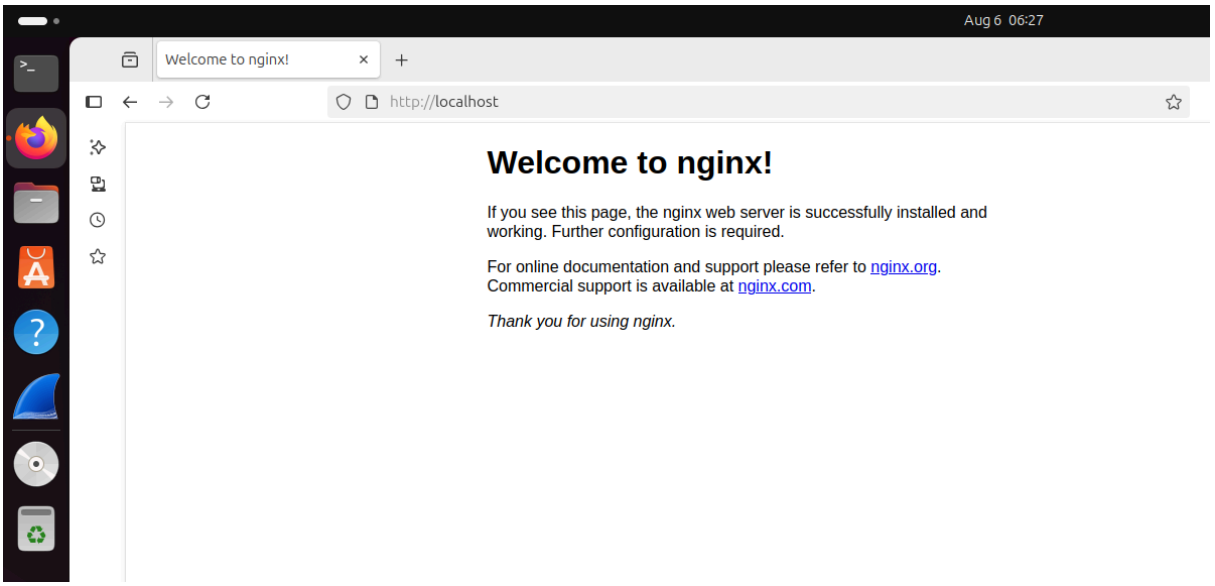
Configuring NGINX

1. Install NGINX

Update the package index and install NGINX:

```
sudo apt install nginx
```

Verified the installation by visiting browser.



---

2. Create a Static Web Page

We'll place our content in a custom location `/var/www/tutorial`:

```
cd /var/www
sudo mkdir tutorial
cd tutorial
sudo vi index.html
```

The HTML Code

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>Hello, Nginx!</title>
</head>
<body>
    <h1>Hello, Nginx!</h1>
    <p>We have just configured our Nginx web server on Ubuntu Server!</p
>
</body>
</html>
```

---

3. Configure a Virtual Host

We configure NGINX to listen on port 81:

```
cd /etc/nginx/sites-enabled
sudo vi tutorial
```

Configuration

```
server {
    listen 81;
    listen [::]:81;
    server_name example.ubuntu.com;
    root /var/www/tutorial;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Restart NGINX:

```
sudo service nginx restart
```

Success, by visiting `localhost:81`.

Installing Modsecurity

4. Install Dependencies for ModSecurity

```
sudo apt-get install bison build-essential ca-certificates curl dh-autoreconf doxygen \
flex gawk git iputils-ping libcurl4-gnutls-dev libexpat1-dev libgeoip-dev liblmdb-dev \
libpcre3-dev libpcre++-dev libssl-dev libtool libxml2 libxml2-dev libyajl-dev locales \
lua5.3-dev pkg-config wget zlib1g-dev zlibc libxslt libgd-dev
sudo apt install git
```

5. Build and Install ModSecurity (libmodsecurity)

Follow these commands:

```
cd /opt
sudo git clone https://github.com/SpiderLabs/ModSecurity
cd ModSecurity
sudo git submodule init
sudo git submodule update
sudo ./build.sh
sudo ./configure
sudo make
sudo make install
```

Download ModSecurity-nginx Connector

To enable Nginx to function as a WAF by integrating with the ModSecurity engine.

6. Install The Module

```
cd /opt
sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git
```

7. Build ModSecurity Module for NGINX

Check NGINX version:

```
nginx -v
```

My version is: `1.24.0`

Download matching source:

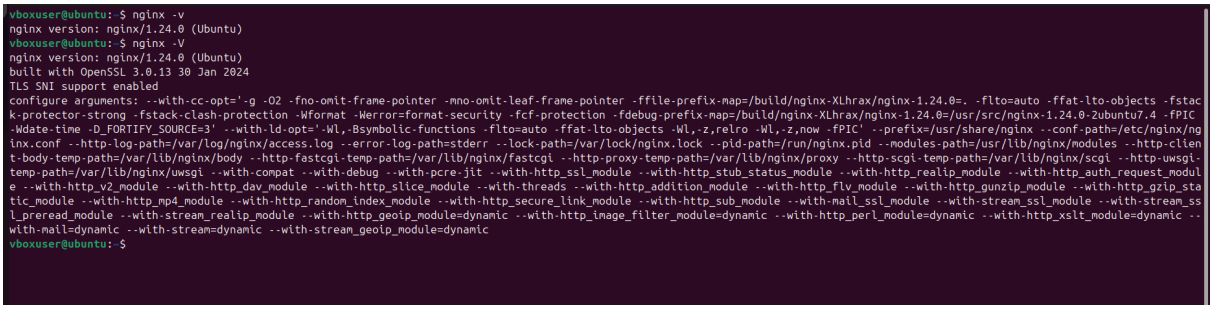
```
cd /opt
sudo wget http://nginx.org/download/nginx-1.24.0.tar.gz
sudo tar -xvzmf nginx-1.24.0.tar.gz
cd nginx-nginx/1.24.0
```

Get your NGINX configure arguments:

```
nginx -V
```

Copy everything after `configure` arguments: and paste it.

This is my `Config`



Compile with ModSecurity module:

```
sudo ./configure --add-dynamic-module=../ModSecurity-nginx --with-cc-opt='-g -O2 -fno-omit-frame-pointer -mno-omit-leaf-frame-pointer -ffile-prefix-map=/build/nginx-XLhrax/nginx-1.24.0=. -flto=auto -ffat-lto-objects -fstack-protector-strong -fstack-clash-protection -Wformat -Werror=format-security -fcf-protection -fdebug-prefix-map=/build/nginx-XLhrax/nginx-1.24.0=/usr/src/nginx-1.24.0-2ubuntu7.4 -fPIC -Wdate-time -D_FORTIFY_SOURCE=3' --with-ld-opt='-Wl,-Bsymbolic-functions -flto=auto -ffat-lto-objects -Wl,-z,relro -Wl,-z,now -fPIC' --prefix=/usr/share/nginx --conf-path=/etc/nginx/nginx.conf --http-log-path=/var/log/nginx/access.log --error-log-path=stderr --lock-path=/var/lock/nginx.lock --pid-path=/run/nginx.pid --modules-path=/usr/lib/nginx/modules --http-client-body-temp-path=/var/lib/nginx/body --http-fastcgi-temp-path=/var/lib/nginx/fastcgi --http-proxy-temp-path=/var/lib/nginx/proxy --http-scgi-temp-path=/var/lib/nginx/scgi --http-uwsgi-temp-path=/var/lib/nginx/uwsgi --with-compat --with-debug --with-pcre-jit --with-http_ssl_module --with-http_stub_status_module --with-http_realip_module --with-http_auth_request_module --with-http_v2_module --with-http_dav_module --with-http_slice_module --with-threads --with-http_addition_module --with-http_flv_module --with-http_gunzip_module --with-http_gzip_static_module --with-http_mp4_module --with-http_random_index_module --with-http_secure_link_module --with-http_sub_module --with-mail_ssl_module --with-stream_ssl_module --with-stream_ssl_preread_module --with-stream_realip_module --with-http_geoip_module=dynamic --with-http_image_filter_module=dynamic --with-http_perl_module=dynamic --with-http_xslt_module=dynamic --with-mail=dynamic --with-stream=dynamic --with-stream_geoip_module=dynamic

sudo make modules
```

Create modules folder and move built module:

```
sudo mkdir /etc/nginx/modules
sudo cp objs/ngx_http_modsecurity_module.so /etc/nginx/modules
```

8. Load the Module in NGINX

Edit `/etc/nginx/nginx.conf` and add:

```
load_module /etc/nginx/modules/ngx_http_modsecurity_module.so;
```

Place it near at the very top.

9. Install OWASP CRS

Remove any pre-existing CRS:

```
sudo rm -rf /usr/share/modsecurity-crs
```

Clone the CRS:

```
sudo git clone https://github.com/coreruleset/coreruleset /usr/local/modsecurity-crs
```

Rename configuration files:

```
sudo mv /usr/local/modsecurity-crs/crs-setup.conf.example /usr/local/modsecurity-crs/crs-setup.conf
sudo mv /usr/local/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example \
    /usr/local/modsecurity-crs/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf
```

10. ModSecurity Configuration

Create config folder:

```
sudo mkdir -p /etc/nginx/modsec
```

Copy necessary files:

```
sudo cp /opt/ModSecurity/unicode.mapping /etc/nginx/modsec
sudo cp /opt/ModSecurity/modsecurity.conf-recommended /etc/nginx/mo
dsec
sudo cp /etc/nginx/modsec/modsecurity.conf-recommended /etc/nginx/m
odsec/modsecurity.conf
```

Edit `/etc/nginx/modsec/modsecurity.conf`:

```
SecRuleEngine On
```

Create main.conf:

```
sudo vi /etc/nginx/modsec/main.conf
```

Insert:

```
Include /etc/nginx/modsec/modsecurity.conf
Include /usr/local/modsecurity-crs/crs-setup.conf
Include /usr/local/modsecurity-crs/rules/*.conf
```

11. Enable ModSecurity in NGINX Site Configuration

Edit your site config (e.g., `/etc/nginx/sites-available/default`):

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

Example block:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;

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

```
    index index.html index.htm index.nginx-debian.html;
    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

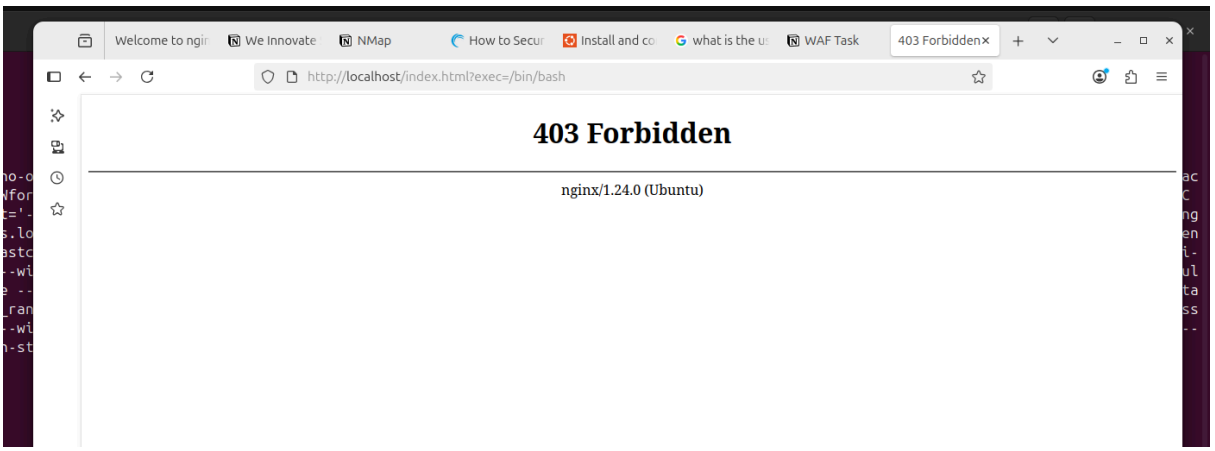
Restart NGINX:

```
sudo systemctl restart nginx
```

12. Test ModSecurity

Test the WAF using **Command Injection:**

```
http://localhost/index.html?exec=/bin/bash
```



Secure.