Configuring a Web Application Firewall with Nginx and ModSecurity

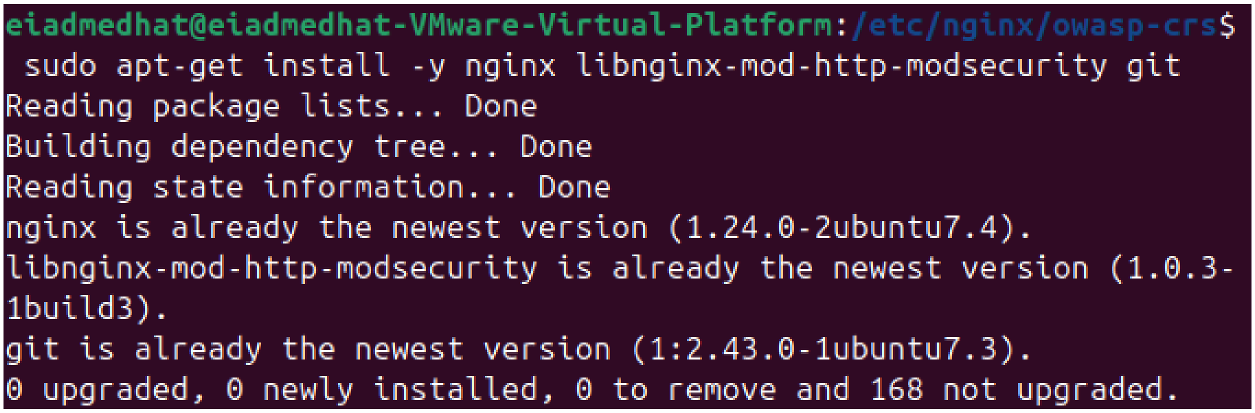
Name: Eiad Medhat Abd El Hady Ibrahim

Date: August 2025

Configuring Backend Web Application (Ubuntu Server)

Installing Nginx and ModSecurity:

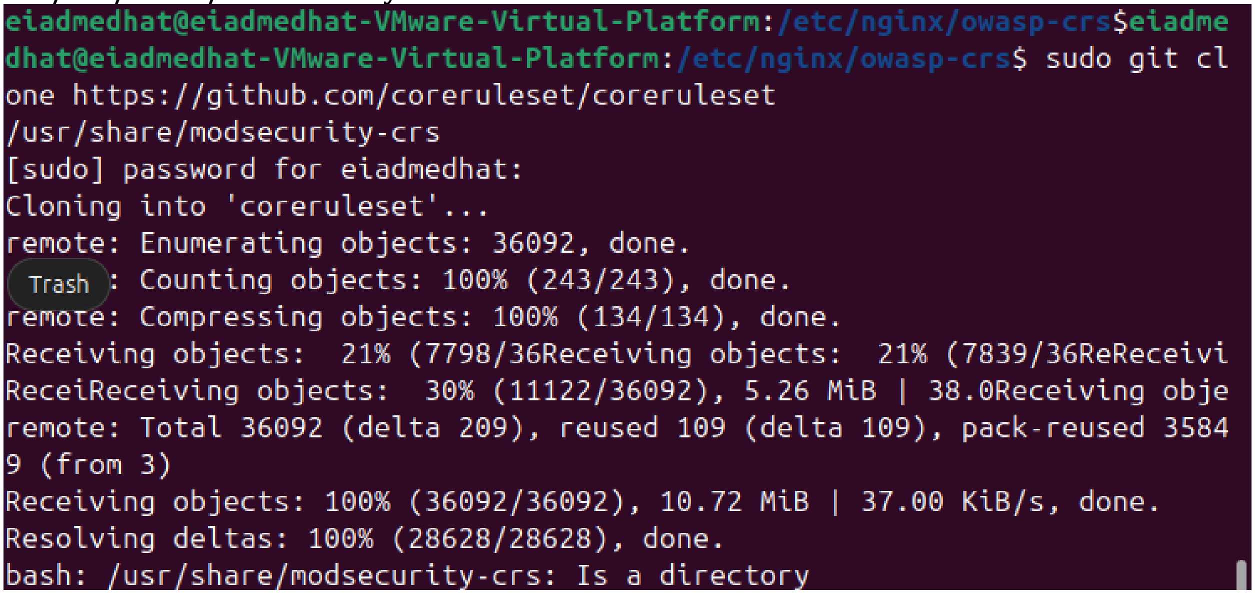
sudo apt-get install -y nginx libnginx-mod-http-modsecurity git



```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx/owasp-crs$
 sudo apt-get install -y nginx libnginx-mod-http-modsecurity git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.4).
libnginx-mod-http-modsecurity is already the newest version (1.0.3-
1build3).
git is already the newest version (1:2.43.0-1ubuntu7.3).
0 upgraded, 0 newly installed, 0 to remove and 168 not upgraded.
```

Acquiring the OWASP Core Rule Set

Command: **sudo git clone https://github.com/coreruleset/coreruleset /usr/share/modsecurity-crs**

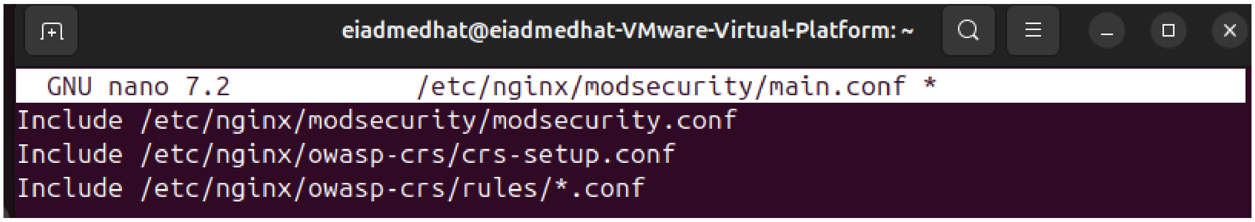


```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx/owasp-crs$eiadme
dhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx/owasp-crs$ sudo git cl
one https://github.com/coreruleset/coreruleset
/usr/share/modsecurity-crs
[sudo] password for eiadmedhat:
Cloning into 'coreruleset'...
remote: Enumerating objects: 36092, done.
Trash : Counting objects: 100% (243/243), done.
remote: Compressing objects: 100% (134/134), done.
Receiving objects:  21% (7798/36Receiving objects:  21% (7839/36ReReceivi
ReceiReceiving objects:  30% (11122/36092), 5.26 MiB | 38.0Receiving obje
remote: Total 36092 (delta 209), reused 109 (delta 109), pack-reused 3584
9 (from 3)
Receiving objects: 100% (36092/36092), 10.72 MiB | 37.00 KiB/s, done.
Resolving deltas: 100% (28628/28628), done.
bash: /usr/share/modsecurity-crs: Is a directory
```

Activating ModSecurity:

Open ModSecurity configuration:

sudo nano /etc/nginx/modsecurity/main.conf

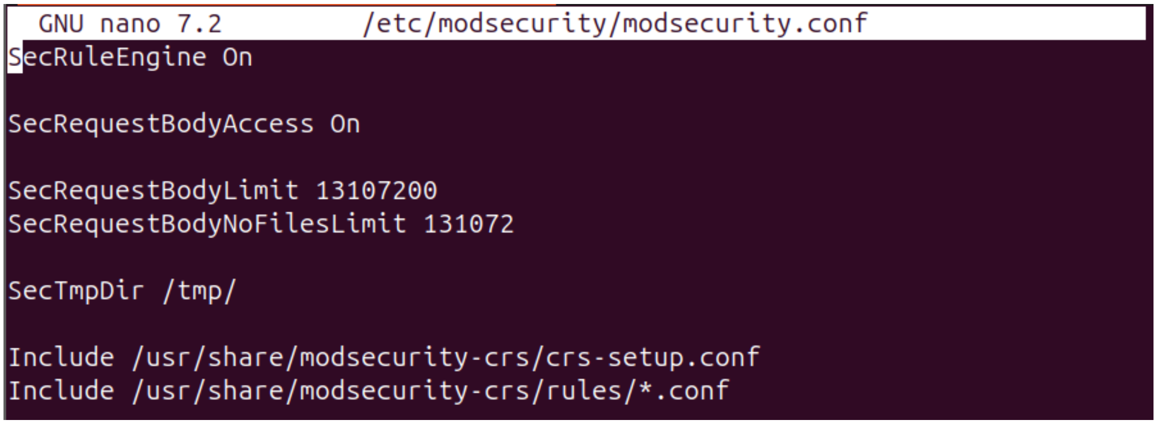


```
  GNU nano 7.2              /etc/nginx/modsecurity/main.conf *
Include /etc/nginx/modsecurity/modsecurity.conf
Include /etc/nginx/owasp-crs/crs-setup.conf
Include /etc/nginx/owasp-crs/rules/*.conf
```

sudo nano /etc/nginx/nginx.conf

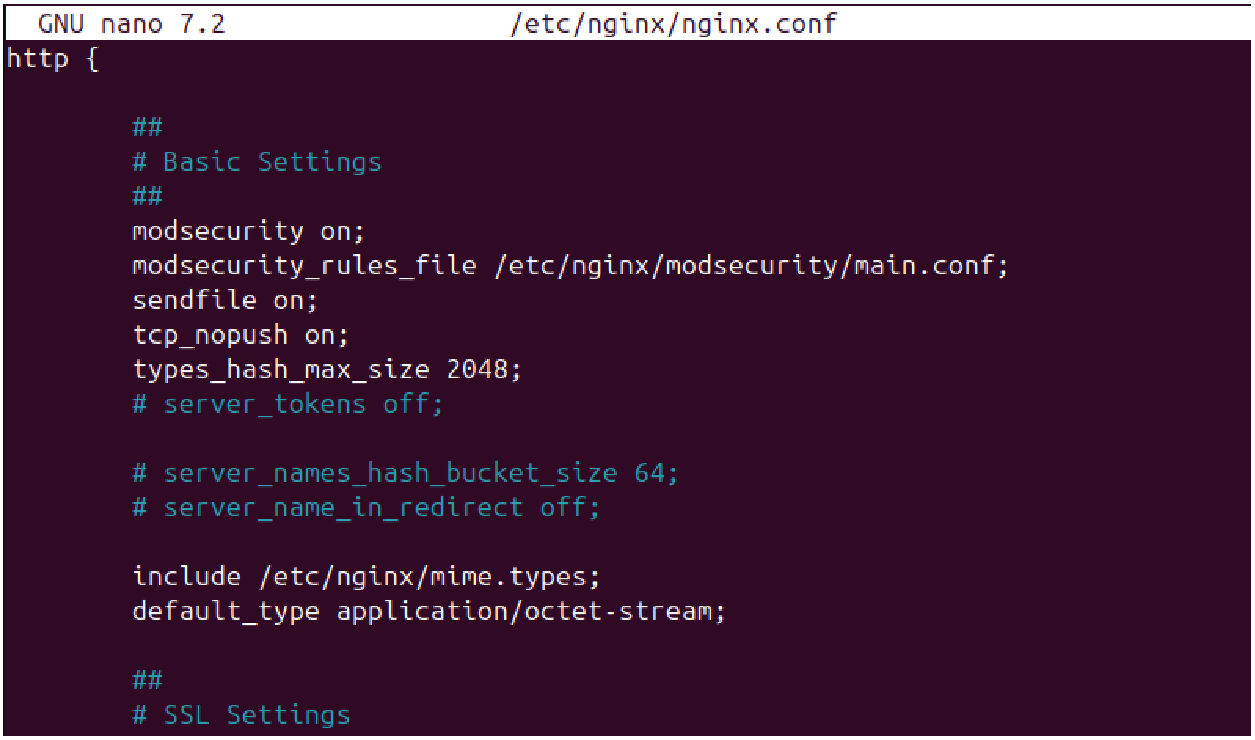
Find and change: SecRuleEngine DetectionOnly → SecRuleEngine On

Include /etc/nginx/modsecurity.conf
Include /etc/nginx/owasp-crs/crs-setup.conf Include /etc/nginx/owasp-crs/rules/*.conf



```
  GNU nano 7.2          /etc/modsecurity/modsecurity.conf
SecRuleEngine On

SecRequestBodyAccess On

SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072

SecTmpDir /tmp/

Include /usr/share/modsecurity-crs/crs-setup.conf
Include /usr/share/modsecurity-crs/rules/*.conf
```

Append the following lines to include OWASP rules:



```
  GNU nano 7.2                        /etc/nginx/nginx.conf
http {

        ##
        # Basic Settings
        ##
        modsecurity on;
        modsecurity_rules_file /etc/nginx/modsecurity/main.conf;
        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
```

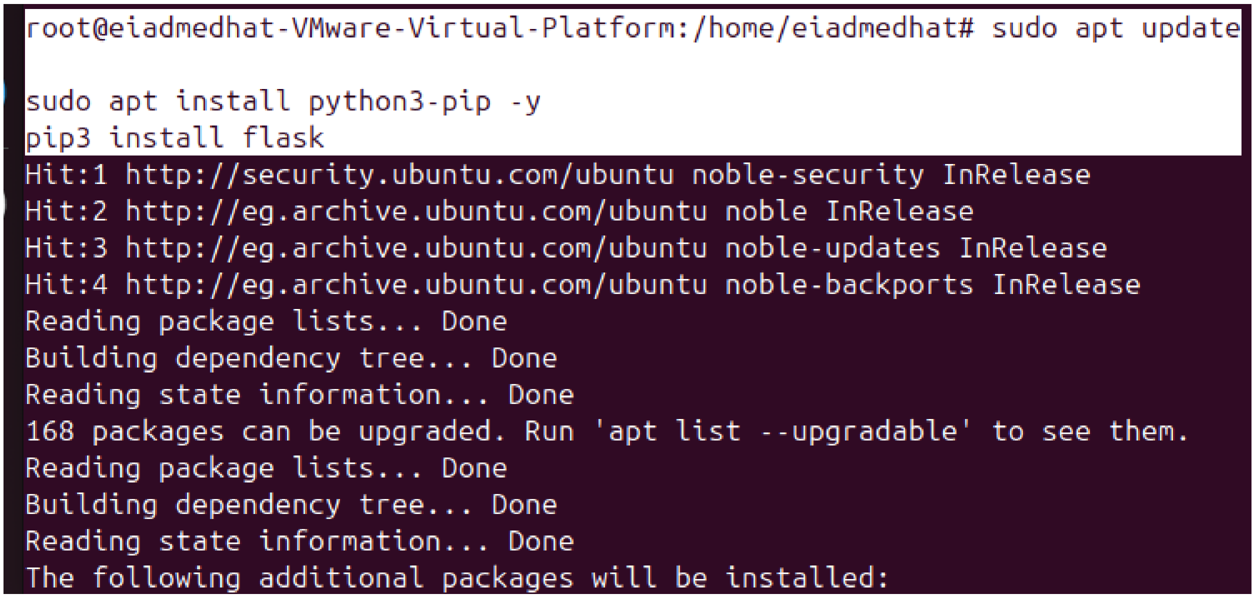
modsecurity on;
modsecurity_rules_file /etc/nginx/modsecurity/main.conf;

Initial Setup of WAF on Ubuntu

Installing Flask

sudo apt install python3-pip

Pip3 install flask

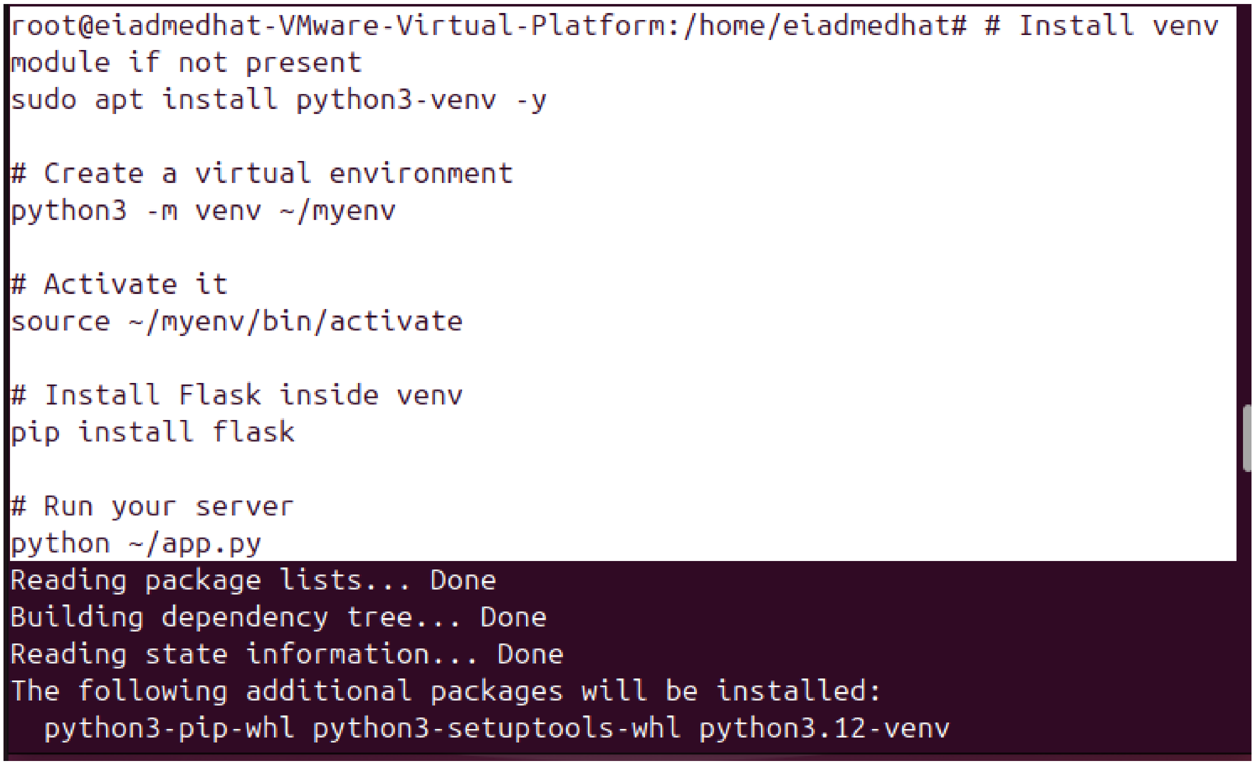


```
root@eiadmedhat-VMware-Virtual-Platform:/home/eiadmedhat# sudo apt update

sudo apt install python3-pip -y
pip3 install flask
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://eg.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://eg.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://eg.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
168 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

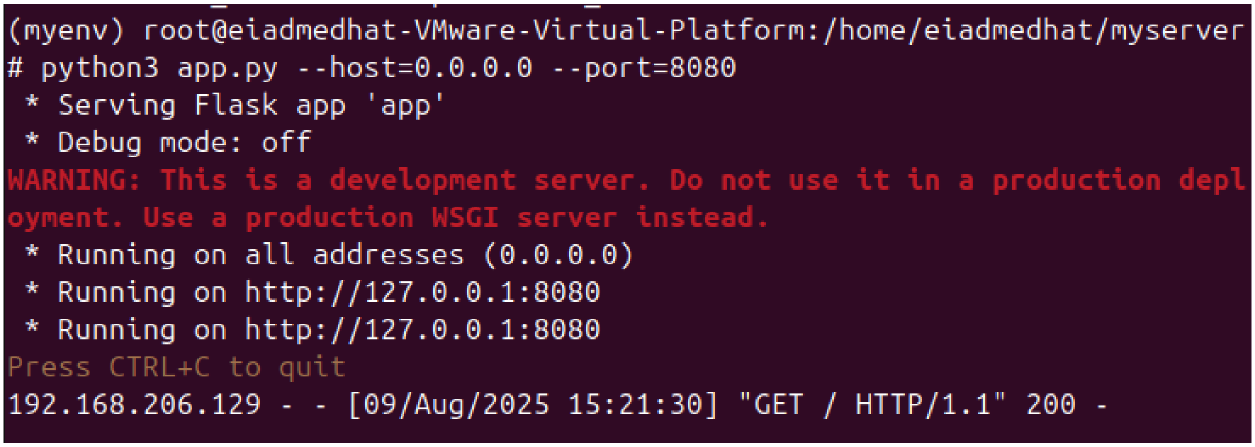
Application Folder Setup

mkdir ~/python-app && cd ~/python-app

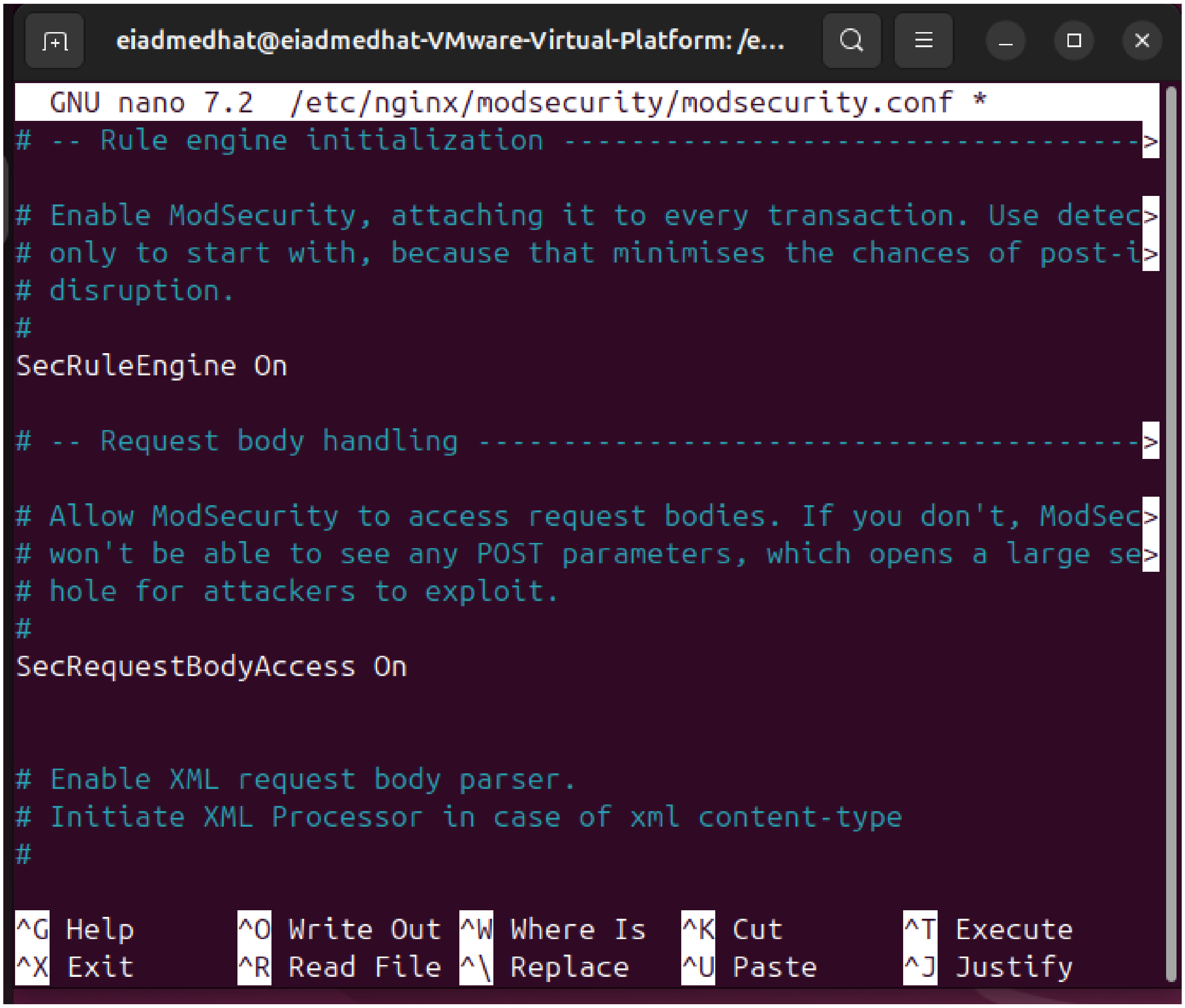
nano app.py



```
root@eiadmedhat-VMware-Virtual-Platform:/home/eiadmedhat# # Install venv
module if not present
sudo apt install python3-venv -y

# Create a virtual environment
python3 -m venv ~/myenv

# Activate it
source ~/myenv/bin/activate

# Install Flask inside venv
pip install flask

# Run your server
python ~/app.py
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pip-whl python3-setuptools-whl python3.12-venv
```



```
(myenv) root@eiadmedhat-VMware-Virtual-Platform:/home/eiadmedhat/myserver
# python3 app.py --host=0.0.0.0 --port=8080
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production depl
oyment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://127.0.0.1:8080
Press CTRL+C to quit
192.168.206.129 - - [09/Aug/2025 15:21:30] "GET / HTTP/1.1" 200 -
```



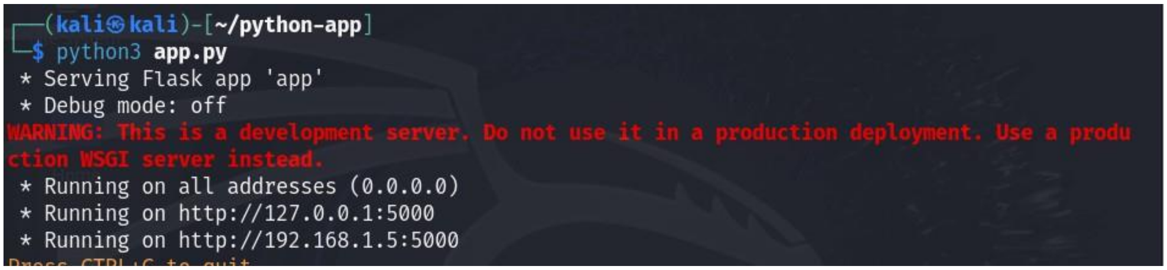
```
  GNU nano 7.2  /etc/nginx/modsecurity/modsecurity.conf *
# -- Rule engine initialization ---------------------------------->

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

# -- Request body handling --------------------------------------->

# Allow ModSecurity to access request bodies. If you don't, ModSec>
# won't be able to see any POST parameters, which opens a large se>
# hole for attackers to exploit.
#
SecRequestBodyAccess On


# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Writing the Flask App

Create a basic application that listens on all interfaces (0.0.0.0:5000) and responds with a message.

Running the Application

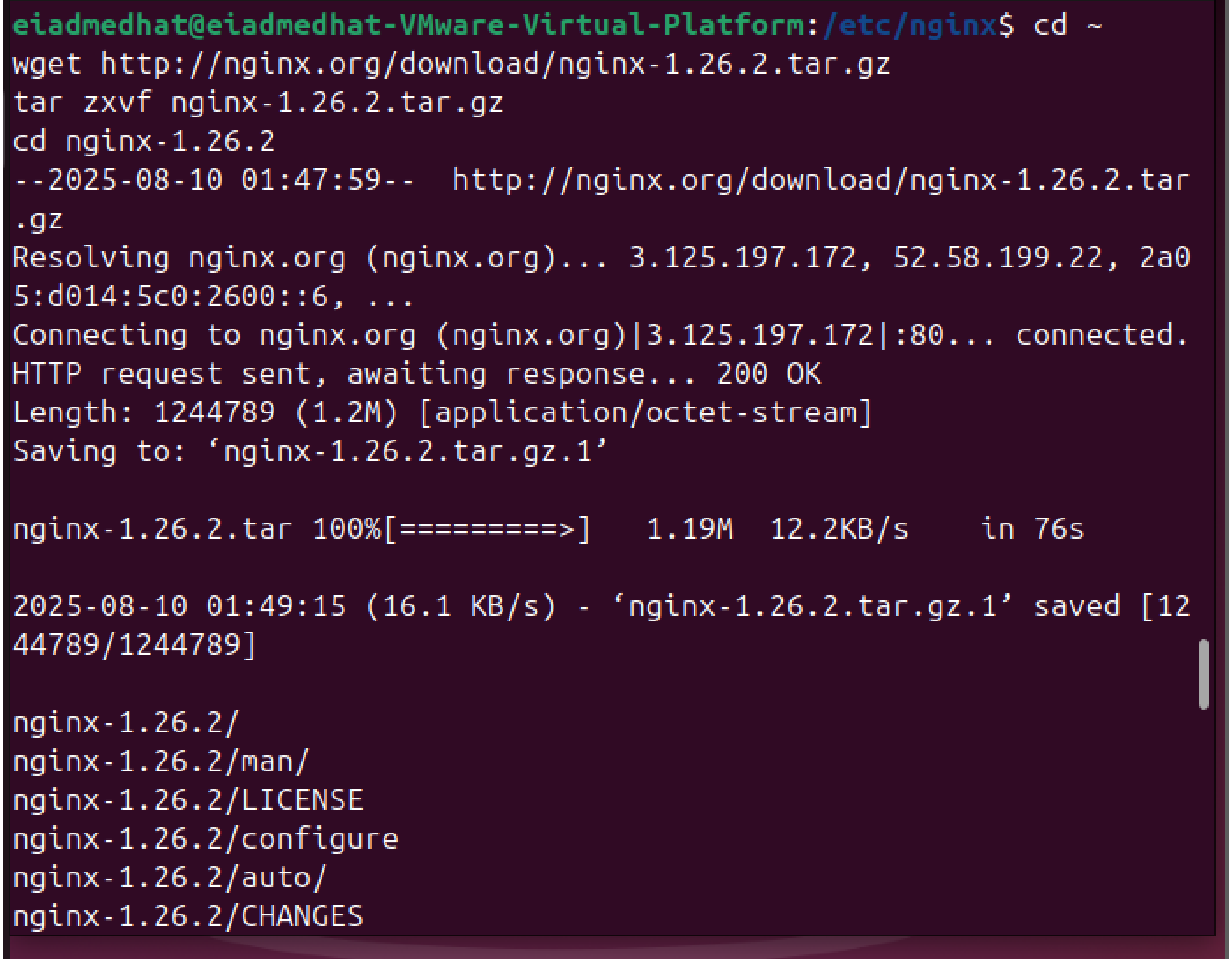
python3 app.py



```
┌──(kali㉿kali)-[~/python-app]
└─$ python3 app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a produ
ction WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.5:5000
Press CTRL+C to quit
```

WAF Proxy Setup on Ubuntu Linux

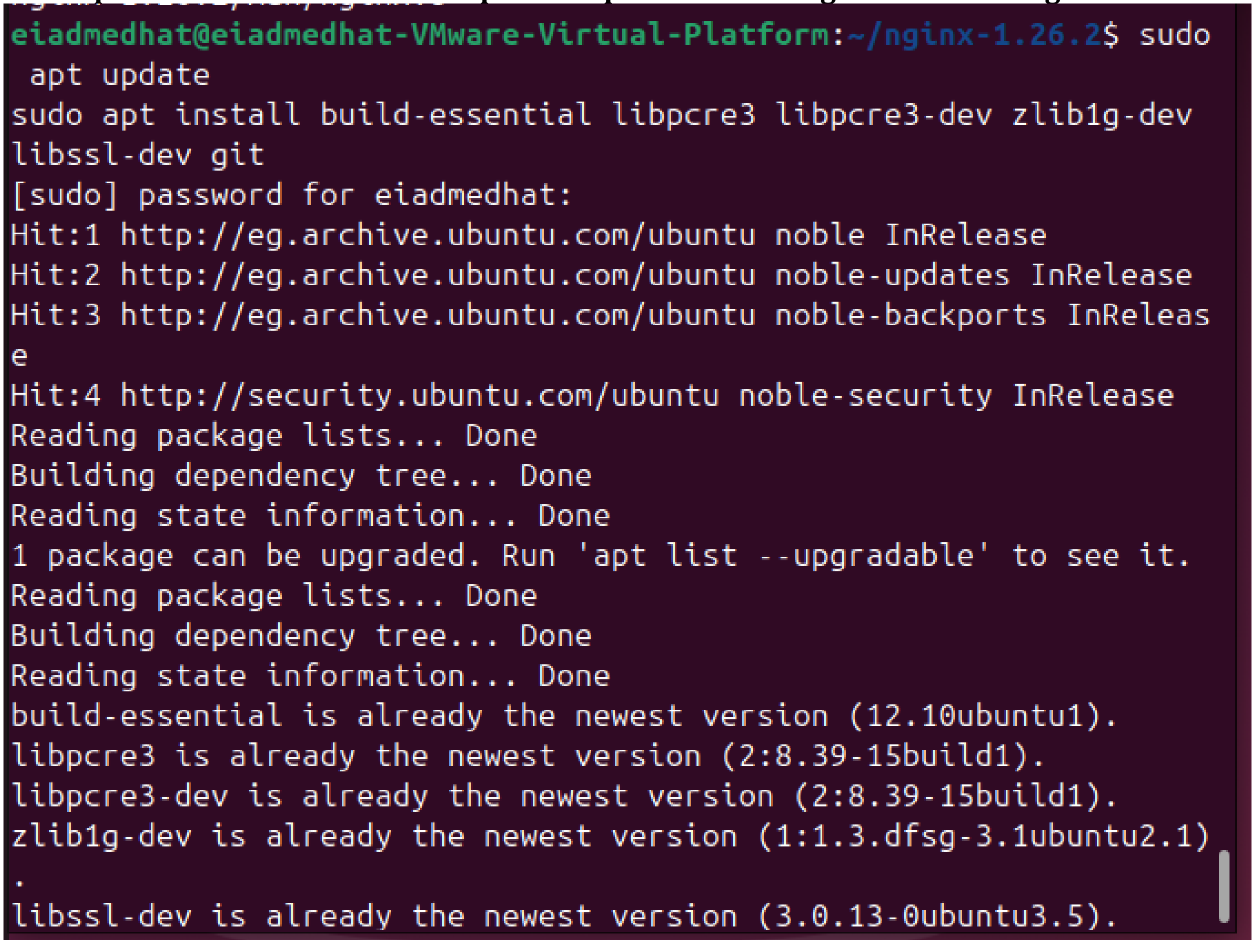
Download and extract in your home directory (or another writable directory):
cd ~wget http://nginx.org/download/nginx-1.26.2.tar.gz
tar zxvf nginx-1.26.2.tar.gz
cd nginx-1.26.2



```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx$ cd ~
wget http://nginx.org/download/nginx-1.26.2.tar.gz
tar zxvf nginx-1.26.2.tar.gz
cd nginx-1.26.2
--2025-08-10 01:47:59--  http://nginx.org/download/nginx-1.26.2.tar
.gz
Resolving nginx.org (nginx.org)... 3.125.197.172, 52.58.199.22, 2a0
5:d014:5c0:2600::6, ...
Connecting to nginx.org (nginx.org)|3.125.197.172|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1244789 (1.2M) [application/octet-stream]
Saving to: ‘nginx-1.26.2.tar.gz.1’

nginx-1.26.2.tar 100%[=========>]   1.19M  12.2KB/s    in 76s

2025-08-10 01:49:15 (16.1 KB/s) - ‘nginx-1.26.2.tar.gz.1’ saved [12
44789/1244789]

nginx-1.26.2/
nginx-1.26.2/man/
nginx-1.26.2/LICENSE
nginx-1.26.2/configure
nginx-1.26.2/auto/
nginx-1.26.2/CHANGES
```

sudo apt update

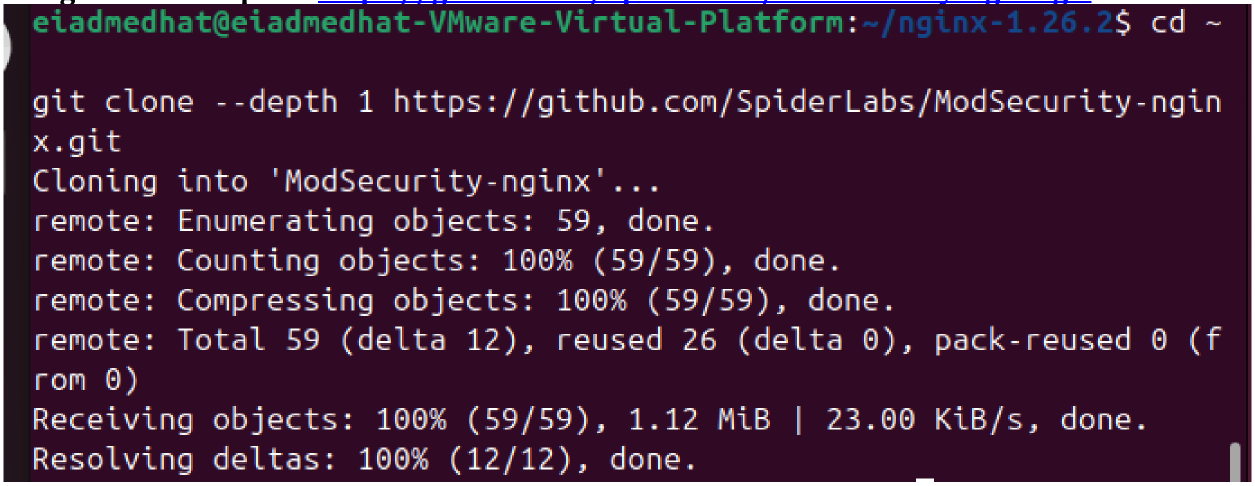
sudo apt install build-essential libpcre3 libpcre3-dev zlib1g-dev libssl-dev git



```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:~/nginx-1.26.2$ sudo
 apt update
sudo apt install build-essential libpcre3 libpcre3-dev zlib1g-dev
libssl-dev git
[sudo] password for eiadmedhat:
Hit:1 http://eg.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://eg.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://eg.archive.ubuntu.com/ubuntu noble-backports InReleas
e
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10ubuntu1).
libpcre3 is already the newest version (2:8.39-15build1).
libpcre3-dev is already the newest version (2:8.39-15build1).
zlib1g-dev is already the newest version (1:1.3.dfsg-3.1ubuntu2.1)
.
libssl-dev is already the newest version (3.0.13-0ubuntu3.5).
```

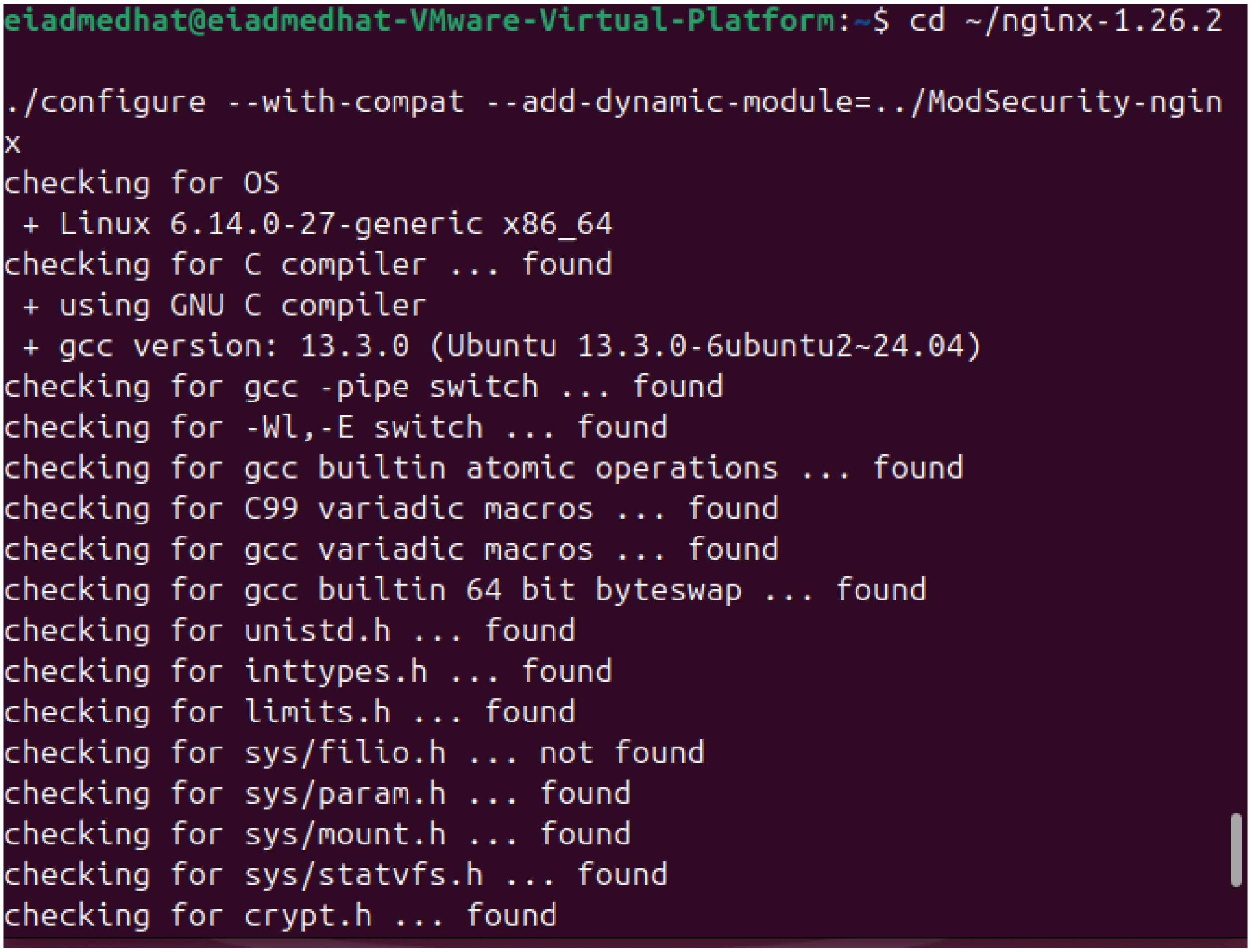
Clone ModSecurity-nginx connector:

cd ~git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git



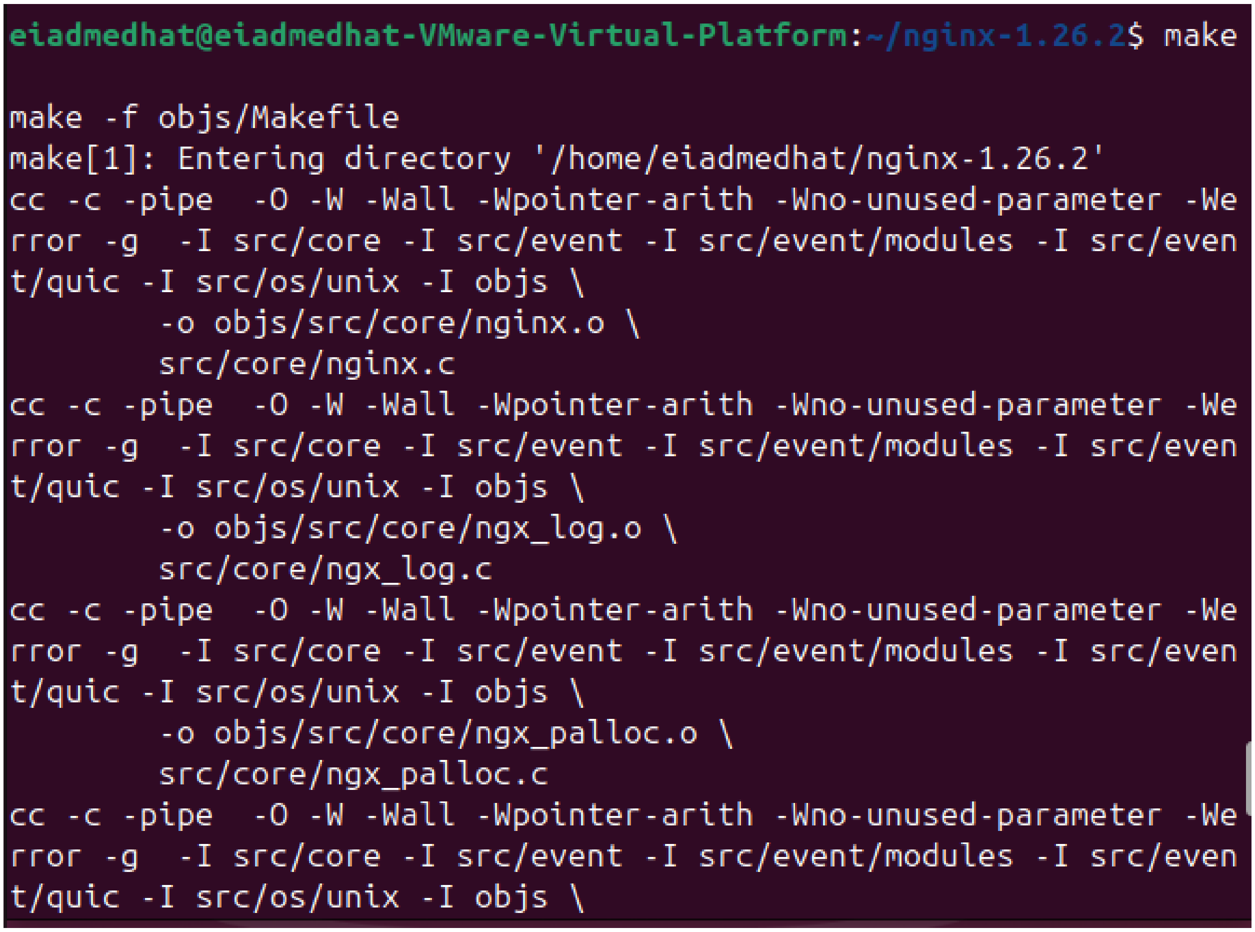
```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:~/nginx-1.26.2$ cd ~

git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-ngin
x.git
Cloning into 'ModSecurity-nginx'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 59 (delta 12), reused 26 (delta 0), pack-reused 0 (f
rom 0)
Receiving objects: 100% (59/59), 1.12 MiB | 23.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
```

Configure NGINX build with ModSecurity module:
cd ~/nginx-1.26.2
./configure --with-compat --add-dynamic-module=../ModSecurity-nginx



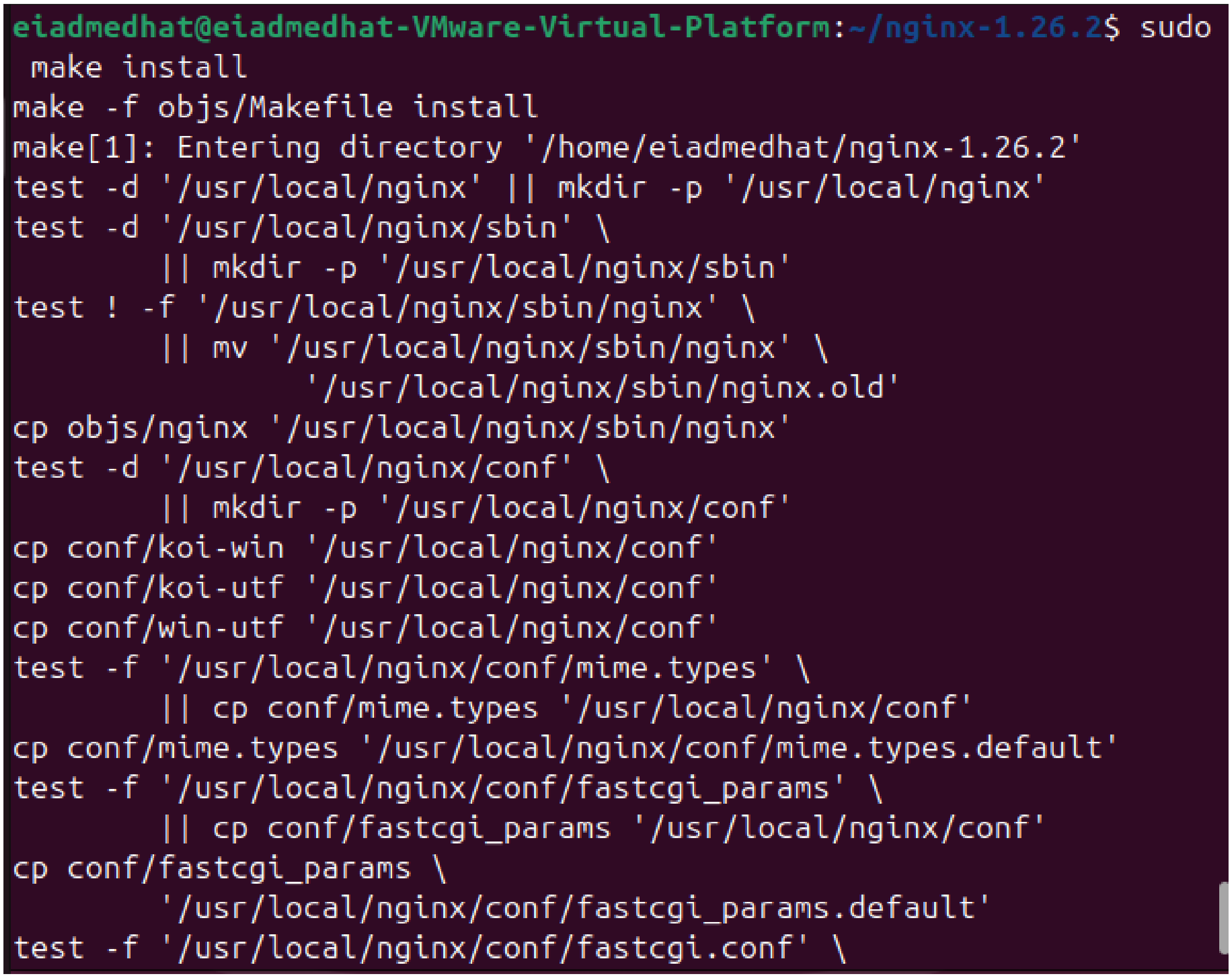
```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:~$ cd ~/nginx-1.26.2

./configure --with-compat --add-dynamic-module=../ModSecurity-ngin
x
checking for OS
 + Linux 6.14.0-27-generic x86_64
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 13.3.0 (Ubuntu 13.3.0-6ubuntu2~24.04)
checking for gcc -pipe switch ... found
checking for -Wl,-E switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
checking for gcc builtin 64 bit byteswap ... found
checking for unistd.h ... found
checking for inttypes.h ... found
checking for limits.h ... found
checking for sys/filio.h ... not found
checking for sys/param.h ... found
checking for sys/mount.h ... found
checking for sys/statvfs.h ... found
checking for crypt.h ... found
```

Build Nginx
make

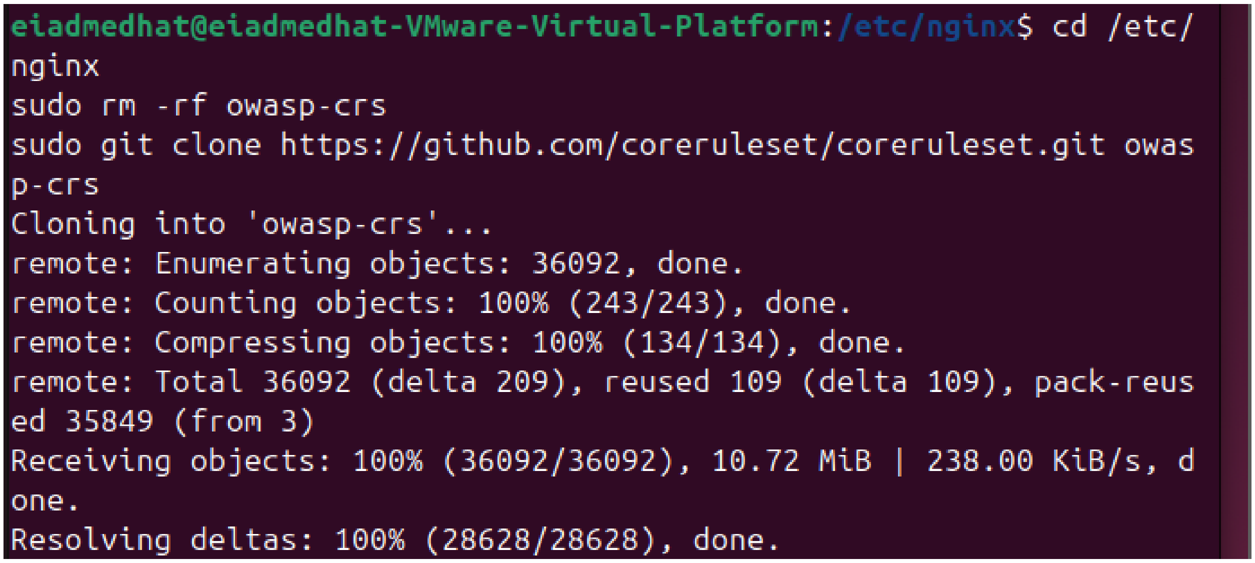


```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:~/nginx-1.26.2$ make

make -f objs/Makefile
make[1]: Entering directory '/home/eiadmedhat/nginx-1.26.2'
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -We
rror -g  -I src/core -I src/event -I src/event/modules -I src/even
t/quic -I src/os/unix -I objs \
        -o objs/src/core/nginx.o \
        src/core/nginx.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -We
rror -g  -I src/core -I src/event -I src/event/modules -I src/even
t/quic -I src/os/unix -I objs \
        -o objs/src/core/ngx_log.o \
        src/core/ngx_log.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -We
rror -g  -I src/core -I src/event -I src/event/modules -I src/even
t/quic -I src/os/unix -I objs \
        -o objs/src/core/ngx_palloc.o \
        src/core/ngx_palloc.c
cc -c -pipe  -O -W -Wall -Wpointer-arith -Wno-unused-parameter -We
rror -g  -I src/core -I src/event -I src/event/modules -I src/even
t/quic -I src/os/unix -I objs \
```

Install the built NGINX
sudo make install

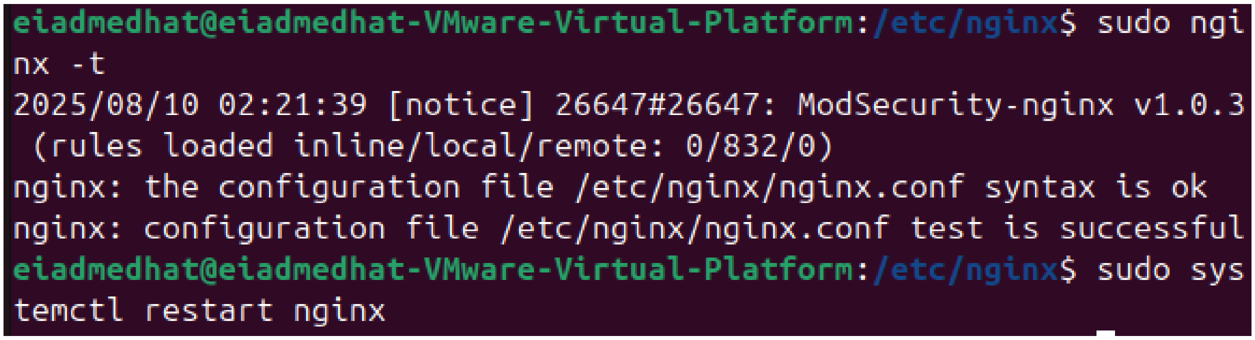


```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:~/nginx-1.26.2$ sudo
 make install
make -f objs/Makefile install
make[1]: Entering directory '/home/eiadmedhat/nginx-1.26.2'
test -d '/usr/local/nginx' || mkdir -p '/usr/local/nginx'
test -d '/usr/local/nginx/sbin' \
        || mkdir -p '/usr/local/nginx/sbin'
test ! -f '/usr/local/nginx/sbin/nginx' \
        || mv '/usr/local/nginx/sbin/nginx' \
                '/usr/local/nginx/sbin/nginx.old'
cp objs/nginx '/usr/local/nginx/sbin/nginx'
test -d '/usr/local/nginx/conf' \
        || mkdir -p '/usr/local/nginx/conf'
cp conf/koi-win '/usr/local/nginx/conf'
cp conf/koi-utf '/usr/local/nginx/conf'
cp conf/win-utf '/usr/local/nginx/conf'
test -f '/usr/local/nginx/conf/mime.types' \
        || cp conf/mime.types '/usr/local/nginx/conf'
cp conf/mime.types '/usr/local/nginx/conf/mime.types.default'
test -f '/usr/local/nginx/conf/fastcgi_params' \
        || cp conf/fastcgi_params '/usr/local/nginx/conf'
cp conf/fastcgi_params \
        '/usr/local/nginx/conf/fastcgi_params.default'
test -f '/usr/local/nginx/conf/fastcgi.conf' \
```



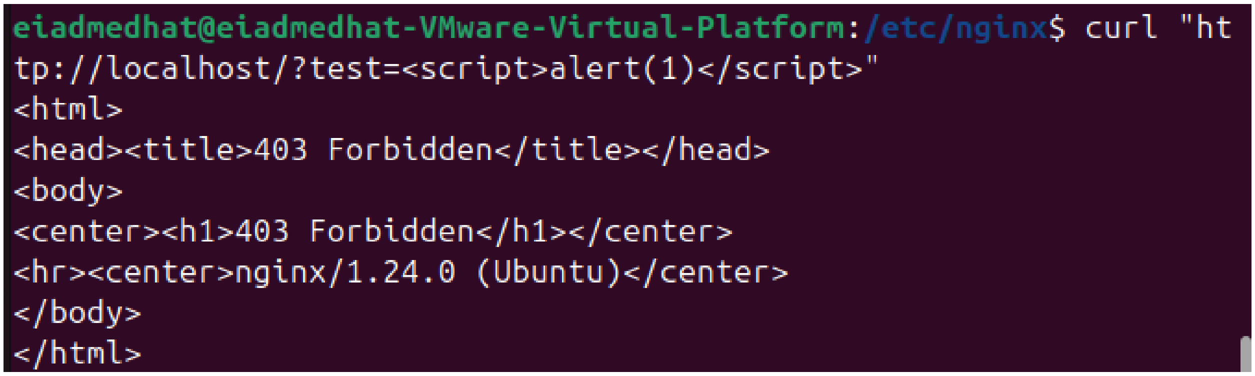
```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx$ cd /etc/
nginx
sudo rm -rf owasp-crs
sudo git clone https://github.com/coreruleset/coreruleset.git owas
p-crs
Cloning into 'owasp-crs'...
remote: Enumerating objects: 36092, done.
remote: Counting objects: 100% (243/243), done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 36092 (delta 209), reused 109 (delta 109), pack-reus
ed 35849 (from 3)
Receiving objects: 100% (36092/36092), 10.72 MiB | 238.00 KiB/s, d
one.
Resolving deltas: 100% (28628/28628), done.
```

sudo nginx -t
sudo systemctl restart nginx

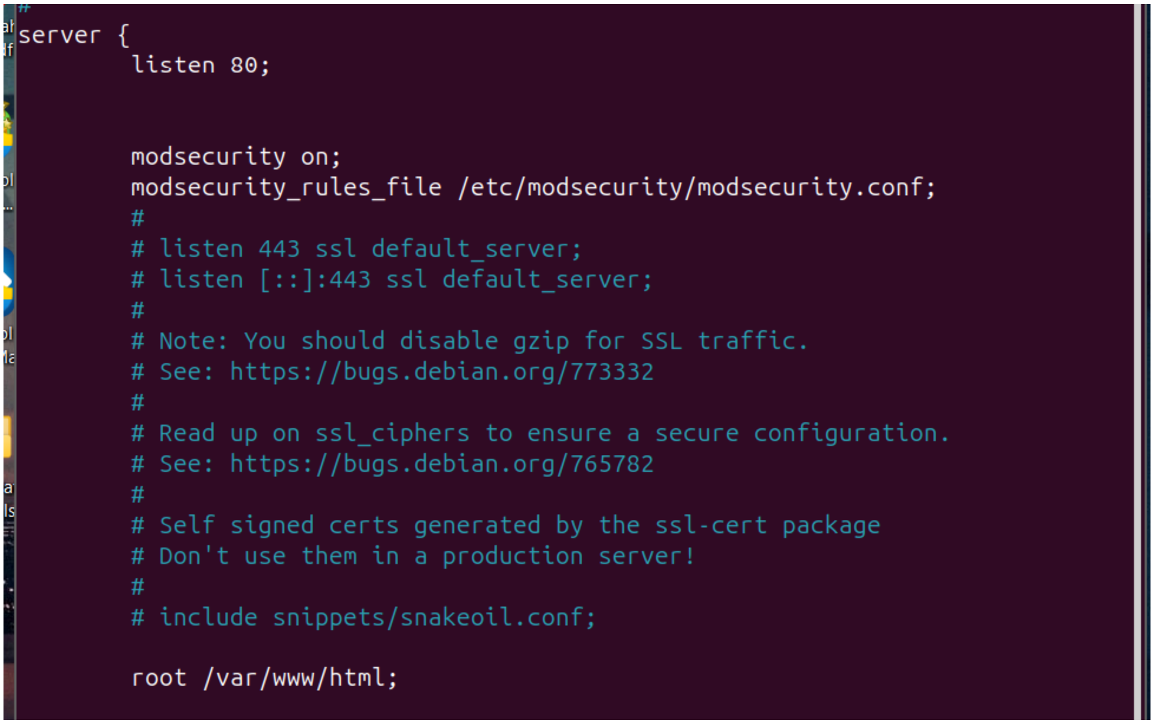


```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx$ sudo ngi
nx -t
2025/08/10 02:21:39 [notice] 26647#26647: ModSecurity-nginx v1.0.3
 (rules loaded inline/local/remote: 0/832/0)
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx$ sudo sys
temctl restart nginx
```

Send a simple attack payload with curl
curl "http://localhost/?test=<script>alert(1)</script>"



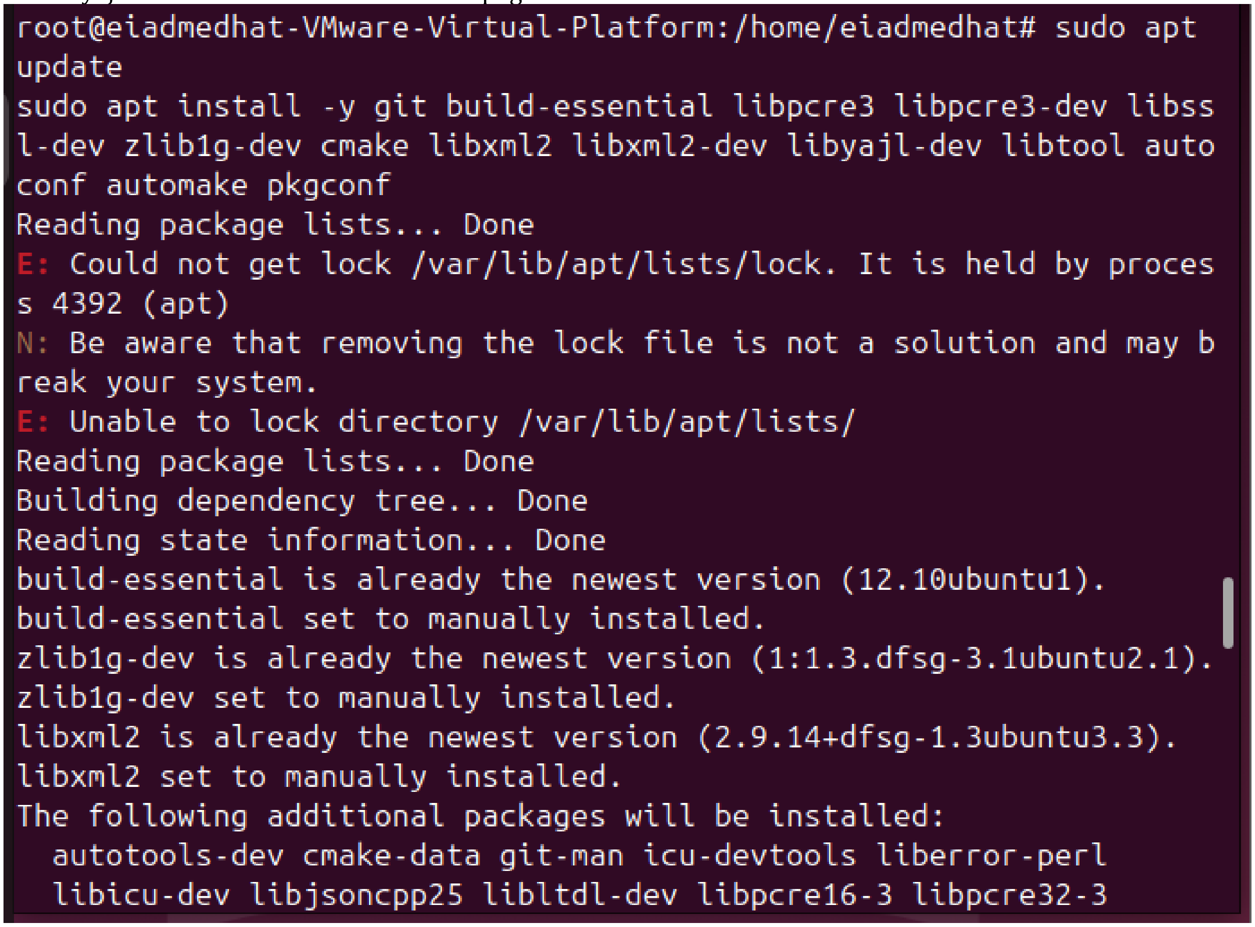
```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/etc/nginx$ curl "ht
tp://localhost/?test=<script>alert(1)</script>"
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.24.0 (Ubuntu)</center>
</body>
</html>
```



```
server {
        listen 80;


        modsecurity on;
        modsecurity_rules_file /etc/modsecurity/modsecurity.conf;
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;
```

Install prerequisites:

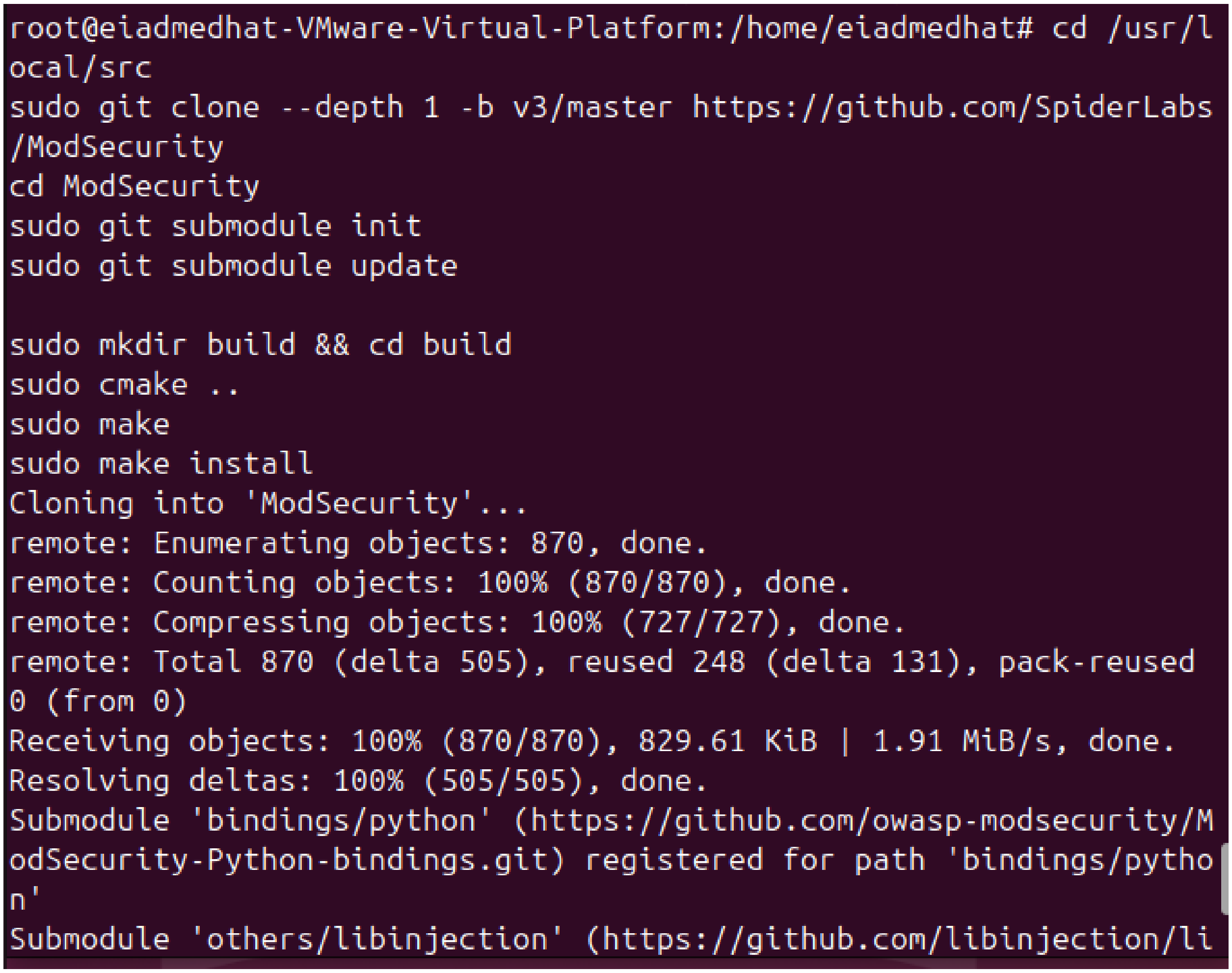
sudo apt update
sudo apt install -y git build-essential libpcre3 libpcre3-dev libssl-dev zlib1g-dev cmake libxml2 libxml2-dev libyajl-dev libtool autoconf automake pkgconf



```
root@eiadmedhat-VMware-Virtual-Platform:/home/eiadmedhat# sudo apt
update
sudo apt install -y git build-essential libpcre3 libpcre3-dev libss
l-dev zlib1g-dev cmake libxml2 libxml2-dev libyajl-dev libtool auto
conf automake pkgconf
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by proces
s 4392 (apt)
N: Be aware that removing the lock file is not a solution and may b
reak your system.
E: Unable to lock directory /var/lib/apt/lists/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10ubuntu1).
build-essential set to manually installed.
zlib1g-dev is already the newest version (1:1.3.dfsg-3.1ubuntu2.1).
zlib1g-dev set to manually installed.
libxml2 is already the newest version (2.9.14+dfsg-1.3ubuntu3.3).
libxml2 set to manually installed.
The following additional packages will be installed:
  autotools-dev cmake-data git-man icu-devtools liberror-perl
  libicu-dev libjsoncpp25 libltdl-dev libpcre16-3 libpcre32-3
```

Download and build ModSecurity:

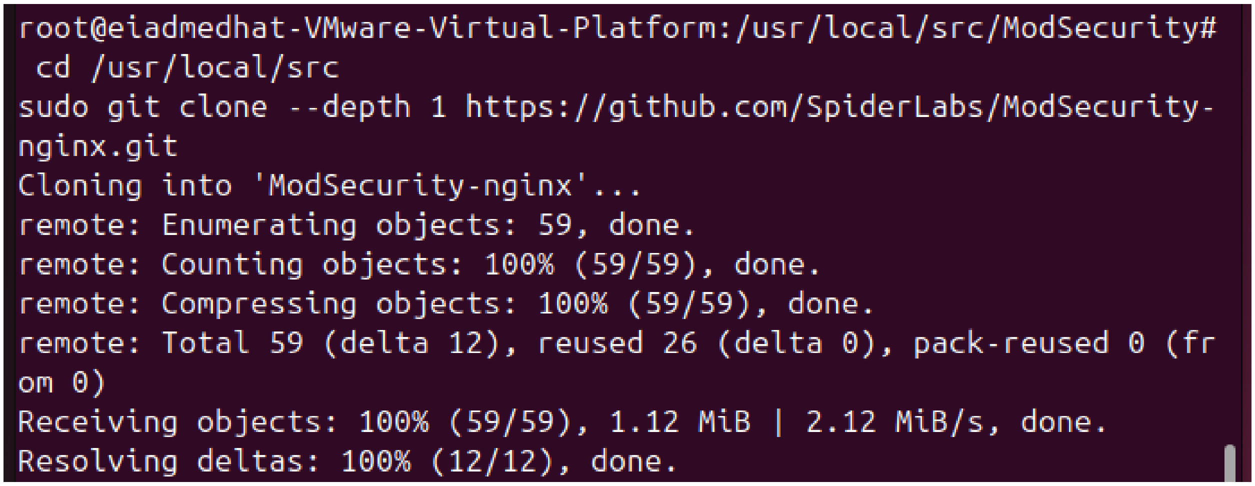
```
    cd /usr/local/src
    sudo git clone --depth 1 -b v3/master https://github.com/SpiderLabs/ModSecurity
    cd ModSecurity
    sudo git submodule init
    sudo git submodule update

    sudo mkdir build && cd build
    sudo cmake ..
    sudo make
    sudo make install
```



```
root@eiadmedhat-VMware-Virtual-Platform:/home/eiadmedhat# cd /usr/l
ocal/src
sudo git clone --depth 1 -b v3/master https://github.com/SpiderLabs
/ModSecurity
cd ModSecurity
sudo git submodule init
sudo git submodule update

sudo mkdir build && cd build
sudo cmake ..
sudo make
sudo make install
Cloning into 'ModSecurity'...
remote: Enumerating objects: 870, done.
remote: Counting objects: 100% (870/870), done.
remote: Compressing objects: 100% (727/727), done.
remote: Total 870 (delta 505), reused 248 (delta 131), pack-reused
0 (from 0)
Receiving objects: 100% (870/870), 829.61 KiB | 1.91 MiB/s, done.
Resolving deltas: 100% (505/505), done.
Submodule 'bindings/python' (https://github.com/owasp-modsecurity/M
odSecurity-Python-bindings.git) registered for path 'bindings/pytho
n'
Submodule 'others/libinjection' (https://github.com/libinjection/li
```

Download modsecurity-nginx connector:

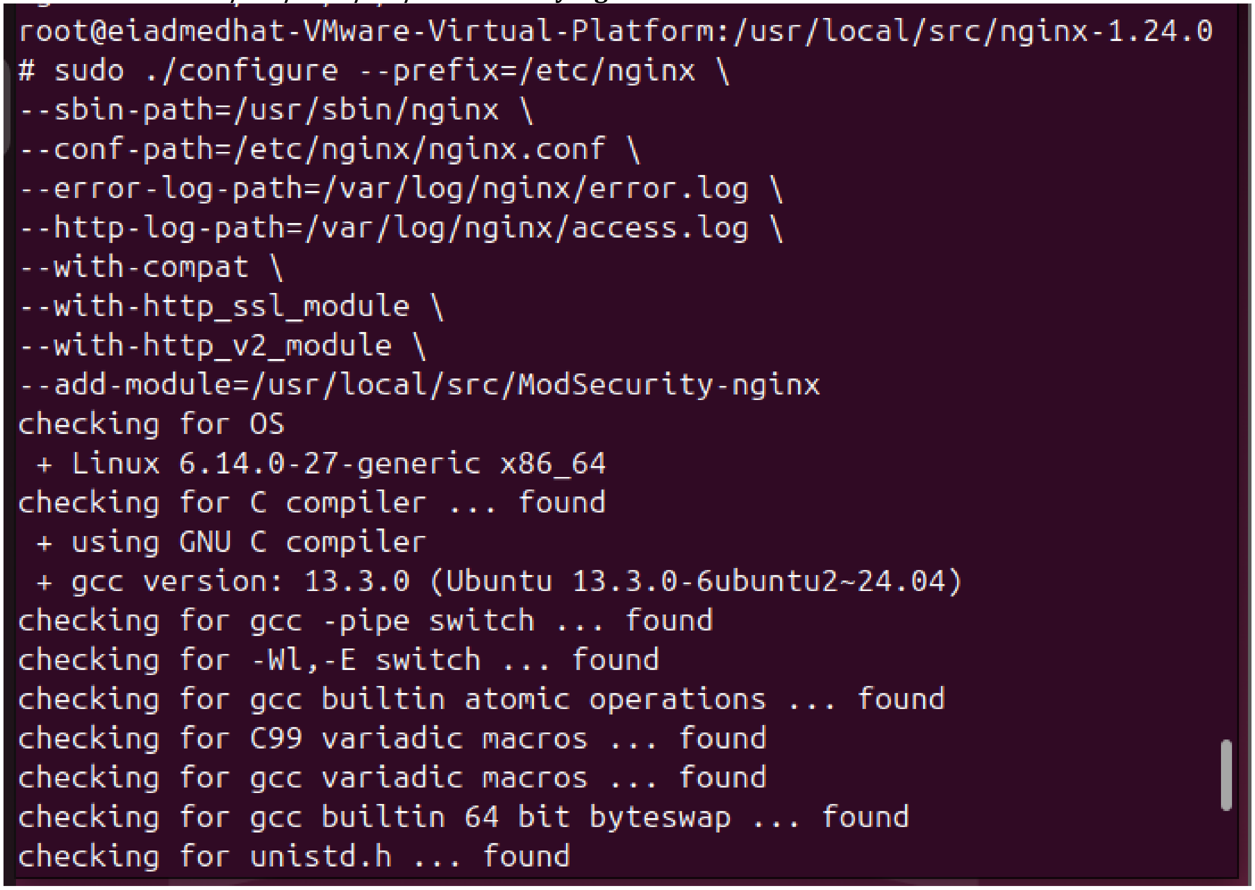
```
cd /usr/local/src
sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git
```



```
root@eiadmedhat-VMware-Virtual-Platform:/usr/local/src/ModSecurity#
 cd /usr/local/src
sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-
nginx.git
Cloning into 'ModSecurity-nginx'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (59/59), done.
remote: Total 59 (delta 12), reused 26 (delta 0), pack-reused 0 (fr
om 0)
Receiving objects: 100% (59/59), 1.12 MiB | 2.12 MiB/s, done.
Resolving deltas: 100% (12/12), done.
```

Configure NGINX build with ModSecurity module:

```
sudo ./configure --prefix=/etc/nginx \
--sbin-path=/usr/sbin/nginx \
--conf-path=/etc/nginx/nginx.conf  \
--error-log-path=/var/log/nginx/error.log  \
--http-log-path=/var/log/nginx/access.log  \
--with-compat  \
--with-http_ssl_module  \
--with-http_v2_module  \
--add-module=/usr/local/src/ModSecurity-nginx
```

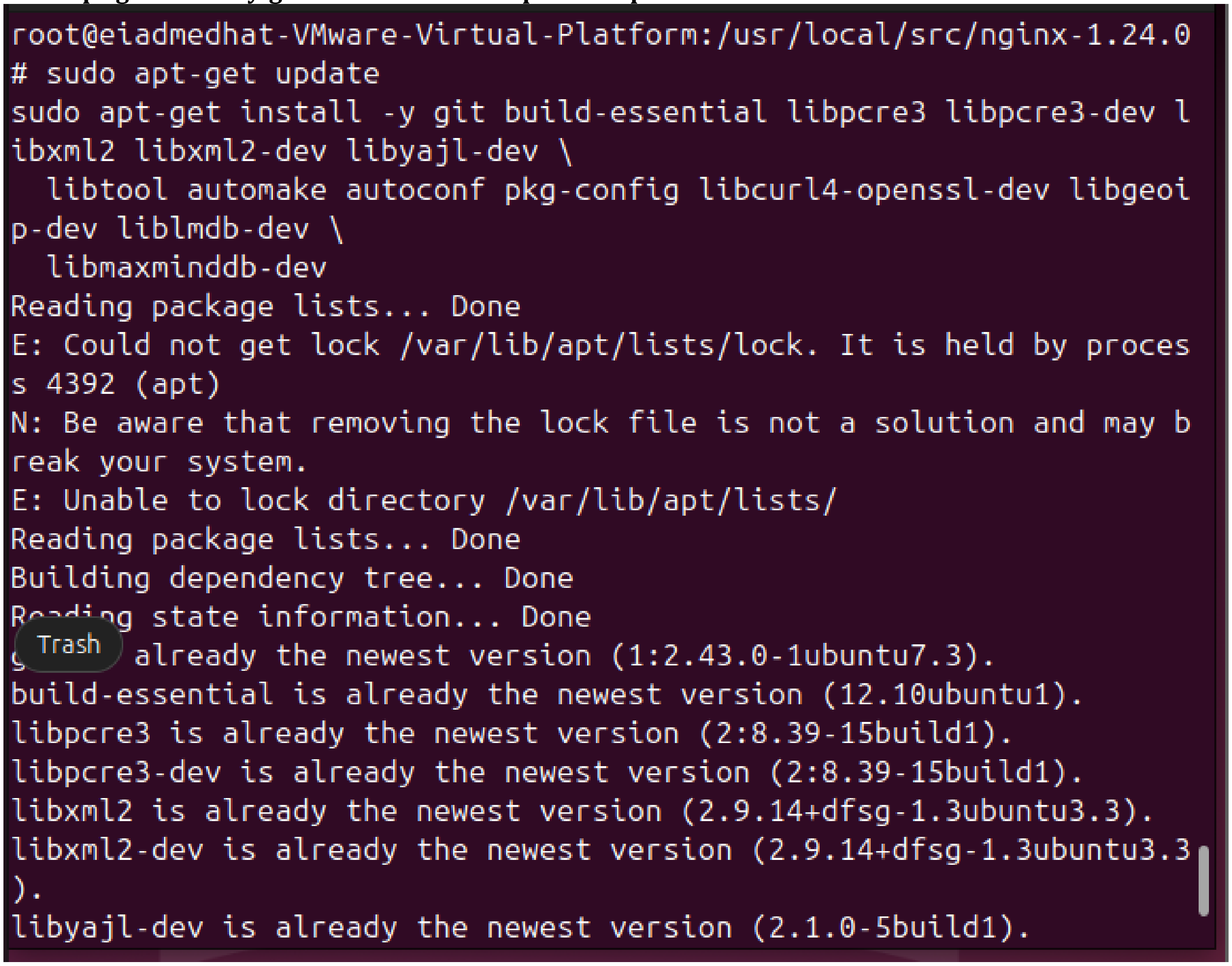


```
root@eiadmedhat-VMware-Virtual-Platform:/usr/local/src/nginx-1.24.0
# sudo ./configure --prefix=/etc/nginx \
--sbin-path=/usr/sbin/nginx \
--conf-path=/etc/nginx/nginx.conf \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-compat \
--with-http_ssl_module \
--with-http_v2_module \
--add-module=/usr/local/src/ModSecurity-nginx
checking for OS
 + Linux 6.14.0-27-generic x86_64
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 13.3.0 (Ubuntu 13.3.0-6ubuntu2~24.04)
checking for gcc -pipe switch ... found
checking for -Wl,-E switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
checking for gcc builtin 64 bit byteswap ... found
checking for unistd.h ... found
```

Install prerequisites:

sudo apt-get update

**sudo apt-get install -y git build-essential libpcre3 libpcre3-dev libxml2 libxml2-dev libyajl-dev **
**libtool automake autoconf pkg-config libcurl4-openssl-dev libgeoip-dev liblmdb-dev **
libmaxminddb-dev

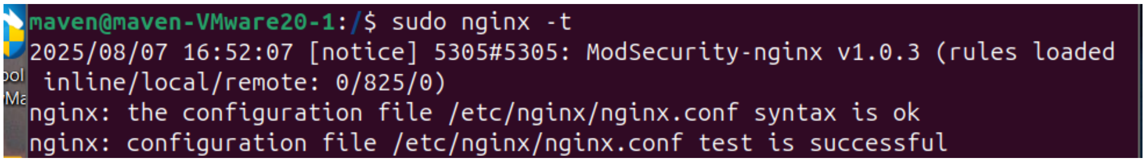


```
root@eiadmedhat-VMware-Virtual-Platform:/usr/local/src/nginx-1.24.0
# sudo apt-get update
sudo apt-get install -y git build-essential libpcre3 libpcre3-dev l
ibxml2 libxml2-dev libyajl-dev \
  libtool automake autoconf pkg-config libcurl4-openssl-dev libgeoi
p-dev liblmdb-dev \
  libmaxminddb-dev
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by proces
s 4392 (apt)
N: Be aware that removing the lock file is not a solution and may b
reak your system.
E: Unable to lock directory /var/lib/apt/lists/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
g        already the newest version (1:2.43.0-1ubuntu7.3).
build-essential is already the newest version (12.10ubuntu1).
libpcre3 is already the newest version (2:8.39-15build1).
libpcre3-dev is already the newest version (2:8.39-15build1).
libxml2 is already the newest version (2.9.14+dfsg-1.3ubuntu3.3).
libxml2-dev is already the newest version (2.9.14+dfsg-1.3ubuntu3.3
).
libyajl-dev is already the newest version (2.1.0-5build1).
```

Validate and Restart

sudo nginx -t

sudo systemctl restart nginx



```
maven@maven-VMware20-1:/$ sudo nginx -t
2025/08/07 16:52:07 [notice] 5305#5305: ModSecurity-nginx v1.0.3 (rules loaded
 inline/local/remote: 0/825/0)
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Functionality Testing

1. Prepare your environment

```
sudo apt update
sudo apt install -y git build-essential libpcre3 libpcre3-dev libssl-dev zlib1g-dev cmake libtool automake autoconf pkgconf
```

2. Download NGINX source (same version as your installed one)

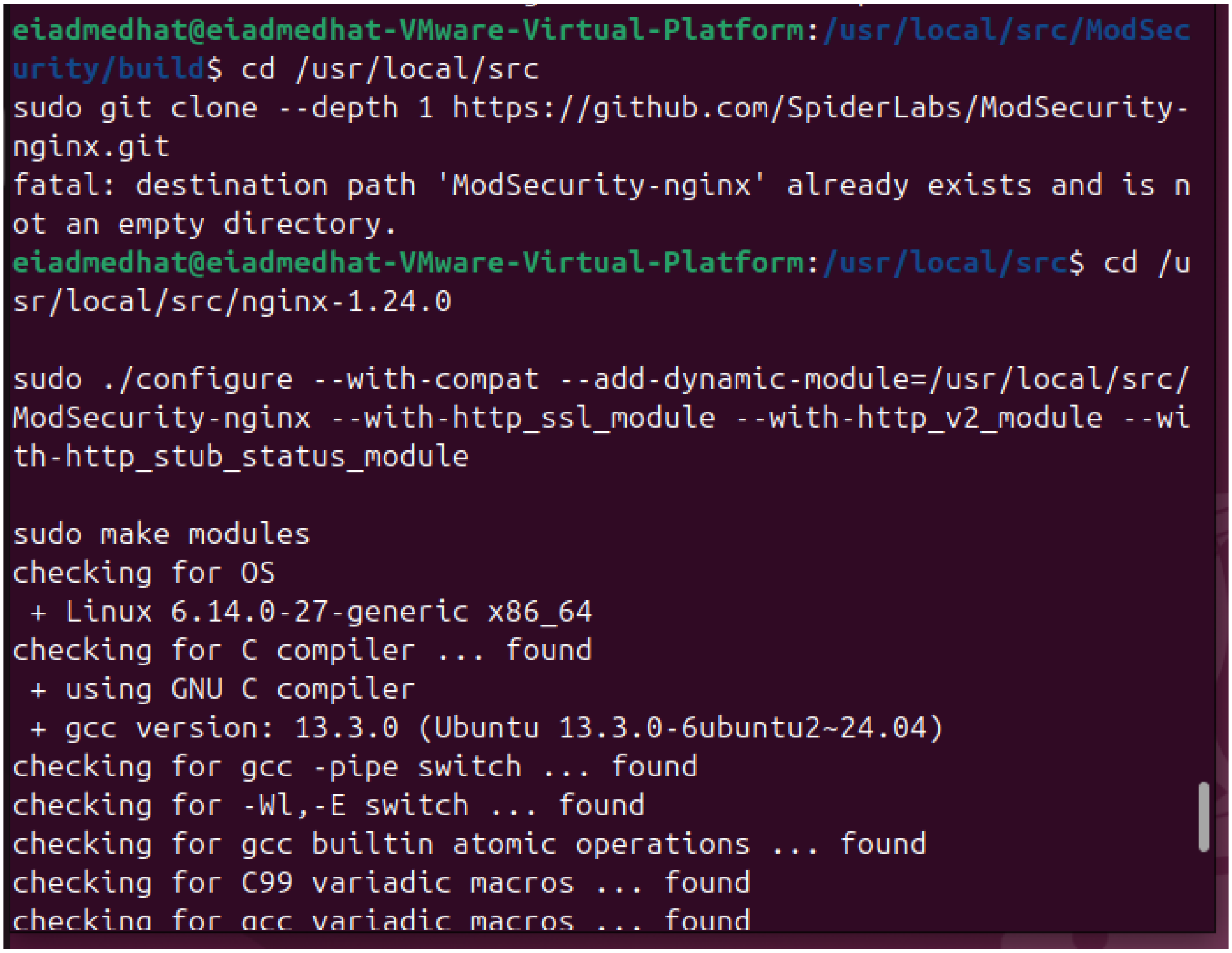
```
cd /usr/local/src
sudo wget http://nginx.org/download/nginx-1.24.0.tar.gz
sudo tar zxvf nginx-1.24.0.tar.gz
```

3. Download ModSecurity v3 (libmodsecurity) source and build it

```
cd /usr/local/src
sudo git clone --depth 1 -b v3/master https://github.com/SpiderLabs/ModSecurity
cd ModSecurity
sudo git submodule init
sudo git submodule update
sudo mkdir build
cd build
sudo cmake ..
sudo make
sudo make install
```

4. Download the ModSecurity-nginx connector

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



```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/usr/local/src/ModSec
urity/build$ cd /usr/local/src
sudo git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-
nginx.git
fatal: destination path 'ModSecurity-nginx' already exists and is n
ot an empty directory.
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/usr/local/src$ cd /u
sr/local/src/nginx-1.24.0

sudo ./configure --with-compat --add-dynamic-module=/usr/local/src/
ModSecurity-nginx --with-http_ssl_module --with-http_v2_module --wi
th-http_stub_status_module

sudo make modules
checking for OS
 + Linux 6.14.0-27-generic x86_64
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 13.3.0 (Ubuntu 13.3.0-6ubuntu2~24.04)
checking for gcc -pipe switch ... found
checking for -Wl,-E switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
```

5. Build NGINX with the ModSecurity module

cd /usr/local/src/nginx-1.24.0

sudo ./configure --with-compat --add-dynamic-module=/usr/local/src/ModSecurity-nginx --with-http_ssl_module --with-http_v2_module --with-http_stub_status_module

sudo make modules

6. Install the compiled module

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

7. Edit your NGINX config to load the module dynamically

At the very top of /etc/nginx/nginx.conf, add:

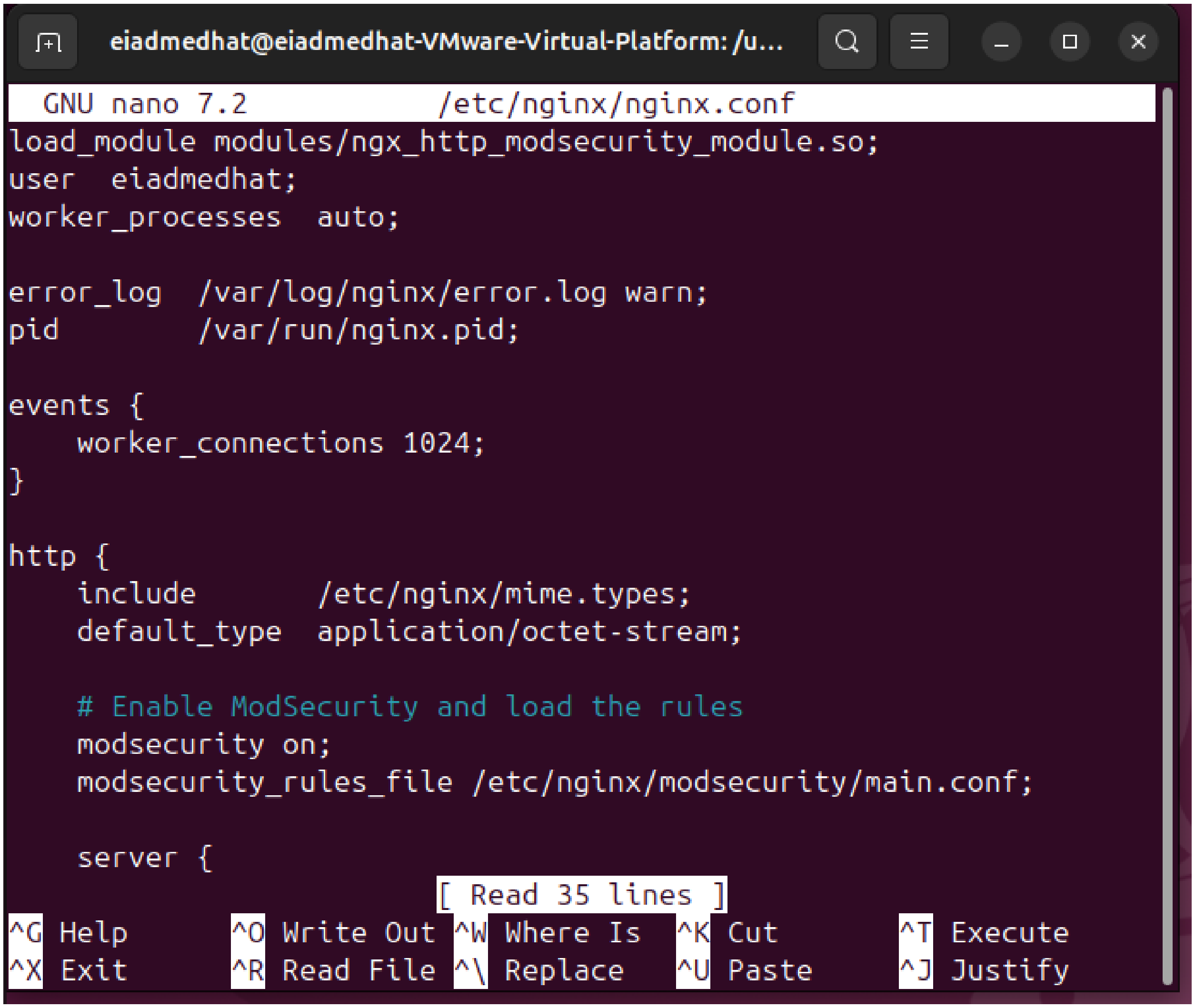
load_module modules/ngx_http_modsecurity_module.so;

8. Configure ModSecurity

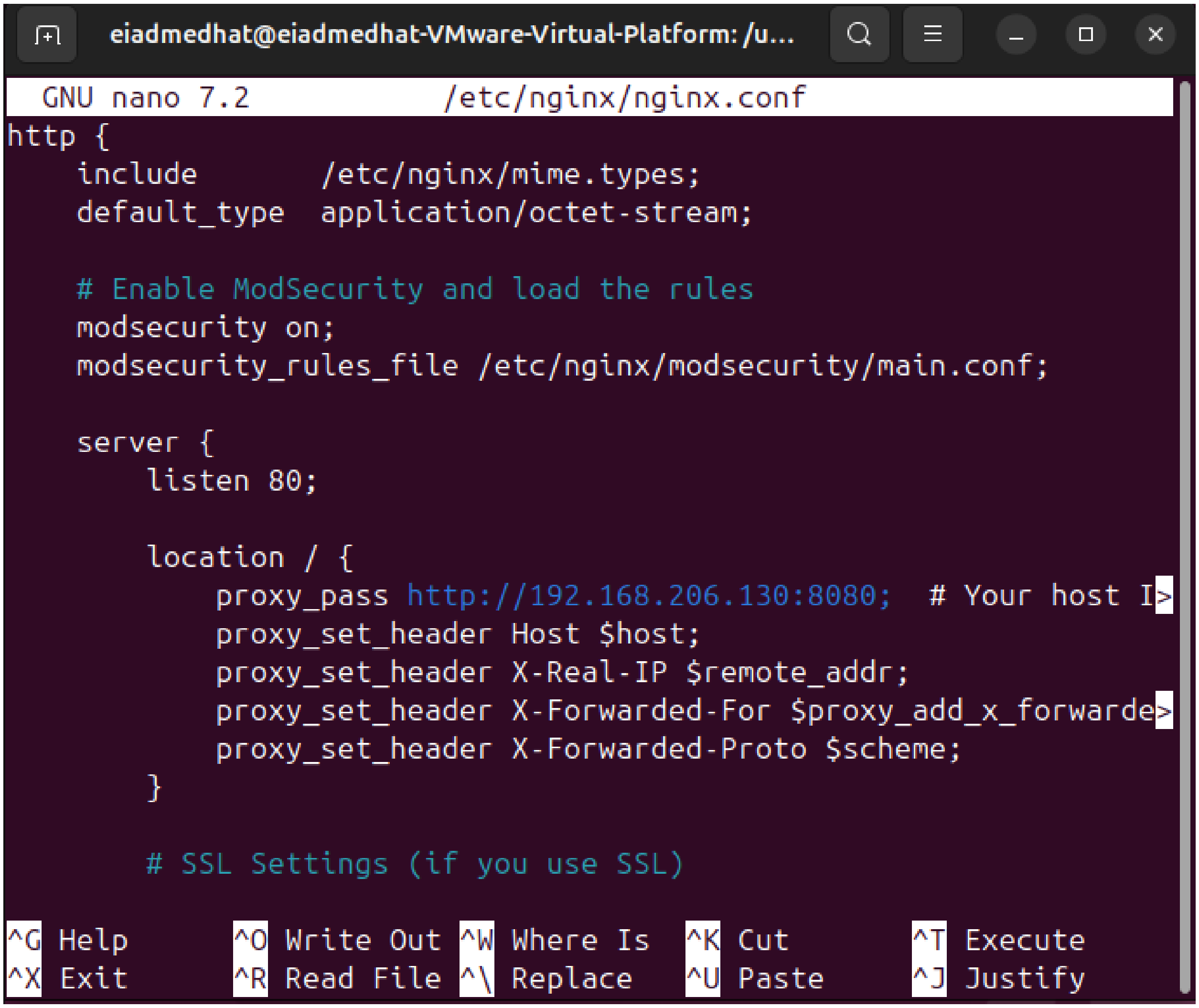
/etc/nginx/nginx.conf include:

```
http {
   modsecurity on;
```

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

    server {
        listen 80;
        location / {
            proxy_pass http://192.168.206.130:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```



```
  GNU nano 7.2              /etc/nginx/nginx.conf
load_module modules/ngx_http_modsecurity_module.so;
user  eiadmedhat;
worker_processes  auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    # Enable ModSecurity and load the rules
    modsecurity on;
    modsecurity_rules_file /etc/nginx/modsecurity/main.conf;

    server {
                        [ Read 35 lines ]
```

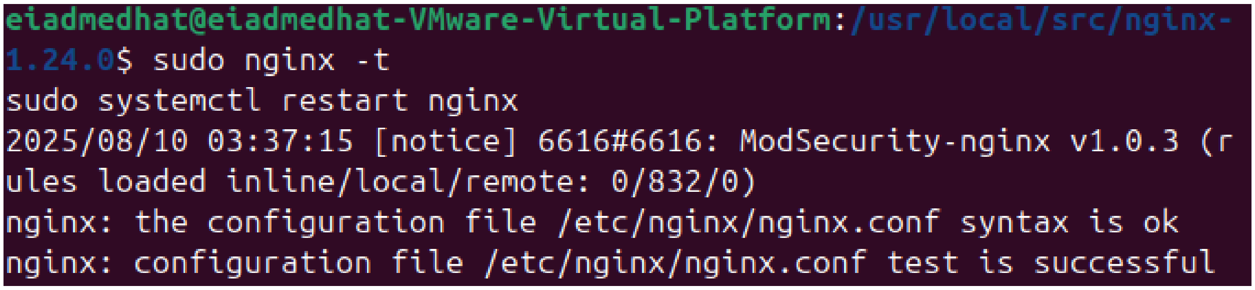


```
 GNU nano 7.2              /etc/nginx/nginx.conf
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    # Enable ModSecurity and load the rules
    modsecurity on;
    modsecurity_rules_file /etc/nginx/modsecurity/main.conf;

    server {
        listen 80;

        location / {
            proxy_pass http://192.168.206.130:8080;  # Your host I>
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarde>
            proxy_set_header X-Forwarded-Proto $scheme;
        }

        # SSL Settings (if you use SSL)
```

Test your configuration and reload NGINX

sudo nginx -t

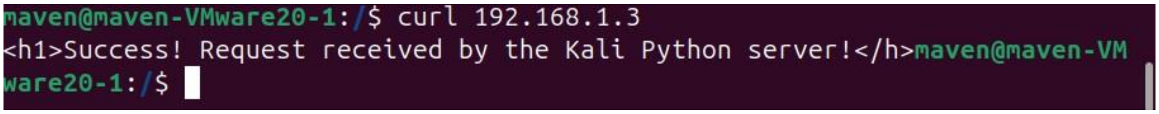
sudo systemctl restart nginx



```
eiadmedhat@eiadmedhat-VMware-Virtual-Platform:/usr/local/src/nginx-
1.24.0$ sudo nginx -t
sudo systemctl restart nginx
2025/08/10 03:37:15 [notice] 6616#6616: ModSecurity-nginx v1.0.3 (r
ules loaded inline/local/remote: 0/832/0)
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Valid Request Check

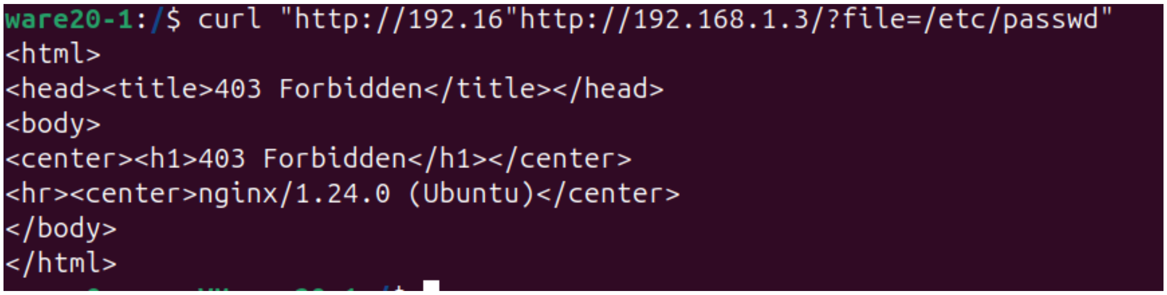
curl 192.168.1.3



```
maven@maven-VMware20-1:/$ curl 192.168.1.3
<h1>Success! Request received by the Kali Python server!</h>maven@maven-VM
ware20-1:/$ 
```

Attempt to Bypass WAF

curl http://192.168.1.3/?file=/etc/passwd



```
ware20-1:/$ curl "http://192.16"http://192.168.1.3/?file=/etc/passwd"
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.24.0 (Ubuntu)</center>
</body>
</html>
```

Conclusion

This guide demonstrated the setup of a secure WAF using Nginx and ModSecurity.

The system effectively protects backend applications and blocks malicious requests as designed.