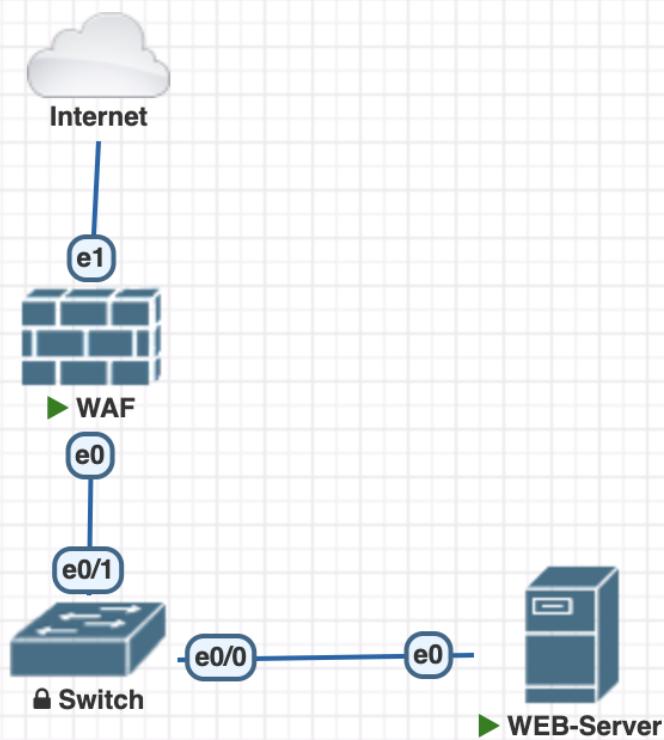


# Занятие 4. Web Application Firewall

## Задание практической работы:

- 1) Используя стенд первой практической работы - изменить инфраструктуру в соответствии со схемой. Главное условие - чтобы машинка с WAF была проксей для WEB-сервера, развернутого во время первой практической работы

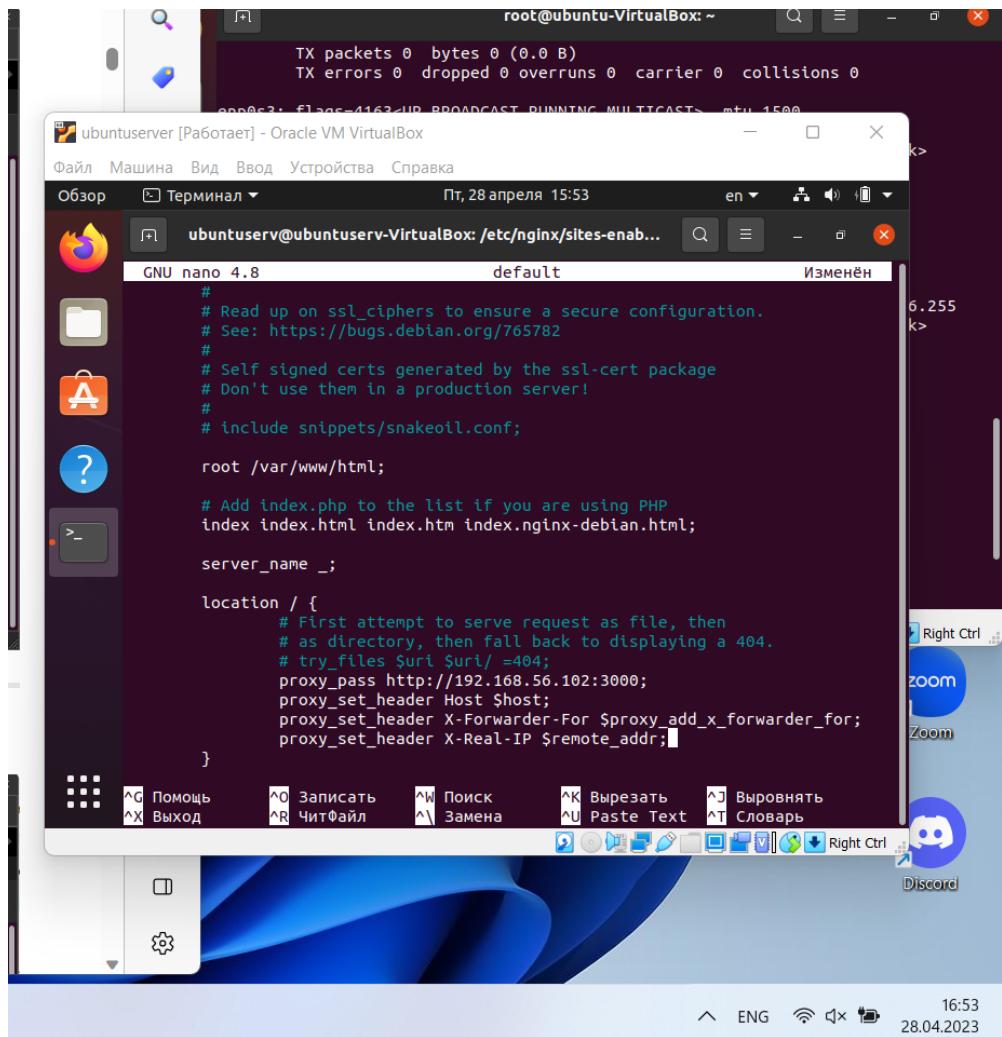


- 2) Установить WAF (ModSecurity 3) на машинку с названием WAF, продемонстрировать что ранее развернутый web-сайт открывается при коннекте на машинку с WAF
- 3) Настроить WAF с детектором OWASP top 10
- 4) Провести 3 атаки из списка OWASP top 10 и продемонстрировать работоспособность WAF (должны быть сработки правил WAF)

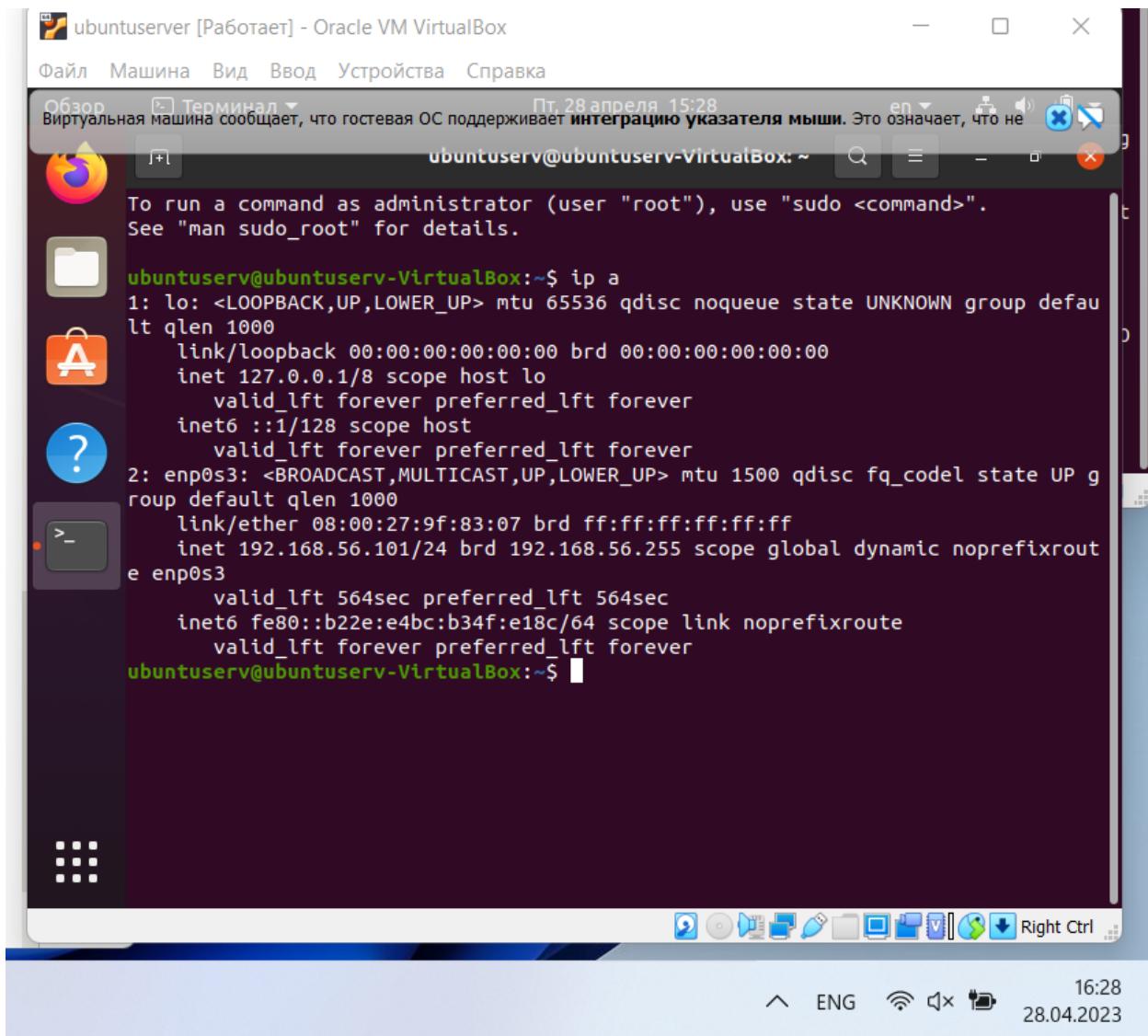
Настройка **WAF (Ubuntu Server)**, как прокси для **WEB-сервера**  
(в работе используется 2 машины на Ubuntu: одна - сервер, другая - waf)

Необходимо использовать следующие параметры:

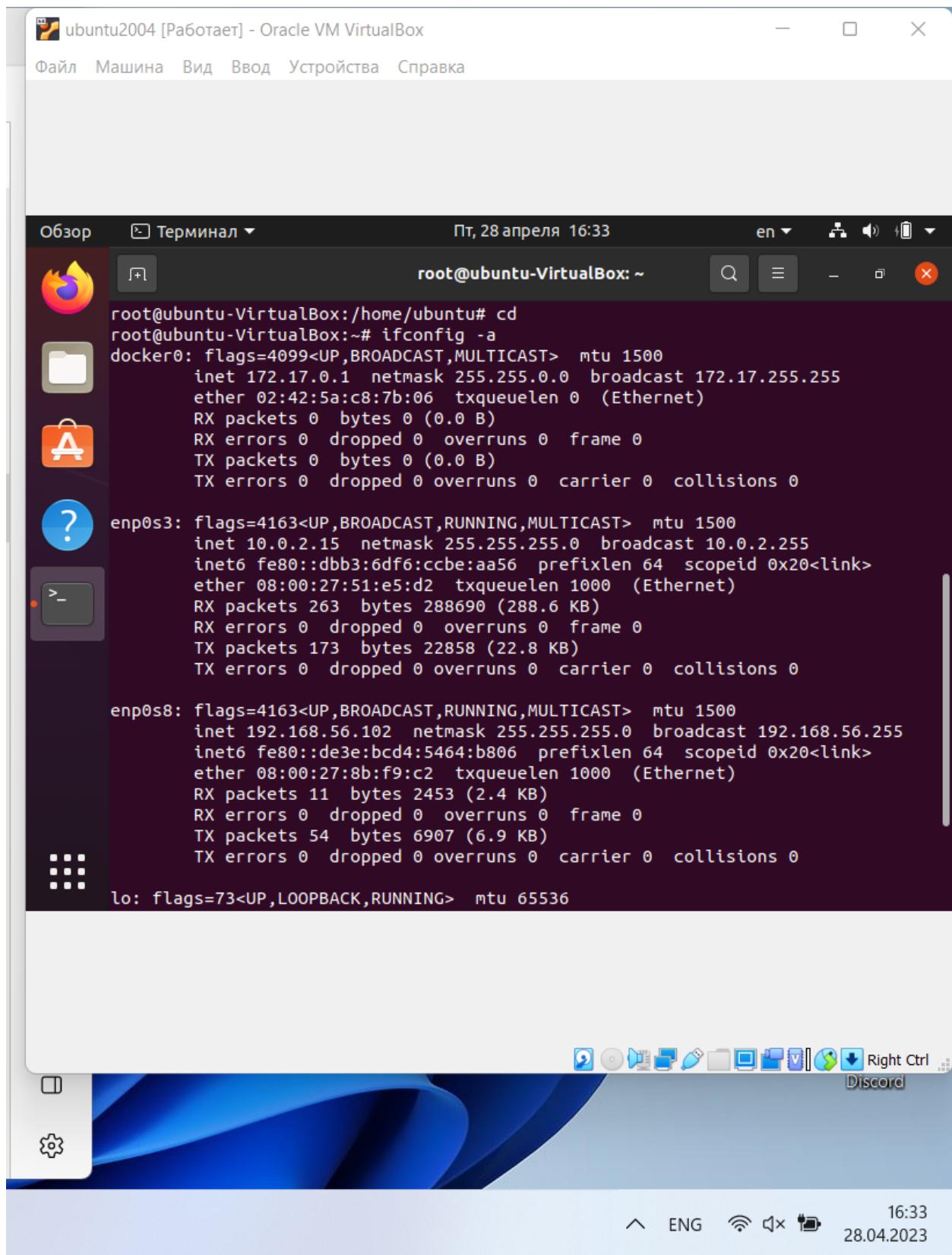
```
proxy_pass http://192.168.56.102:3000;
proxy_set_header Host $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Real-IP $remote_addr;
```



## Интерфейсы Ubuntu Server



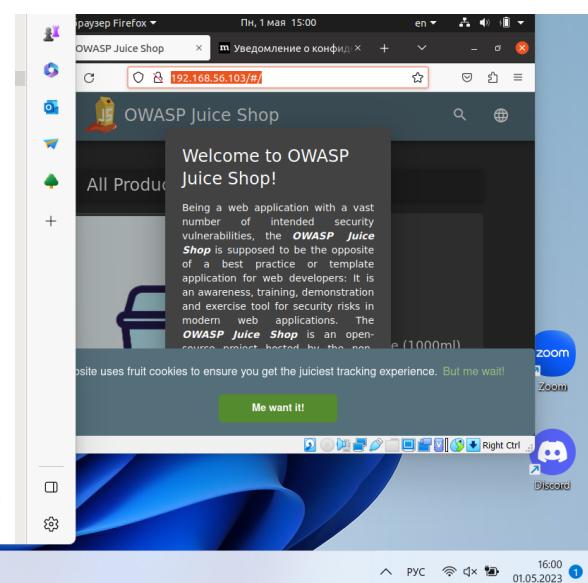
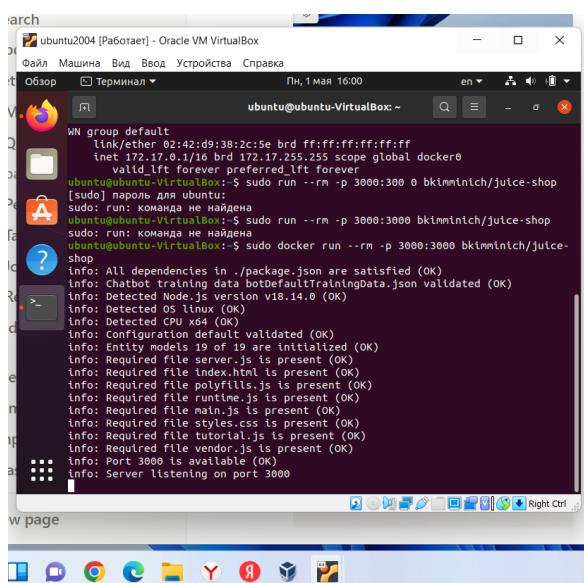
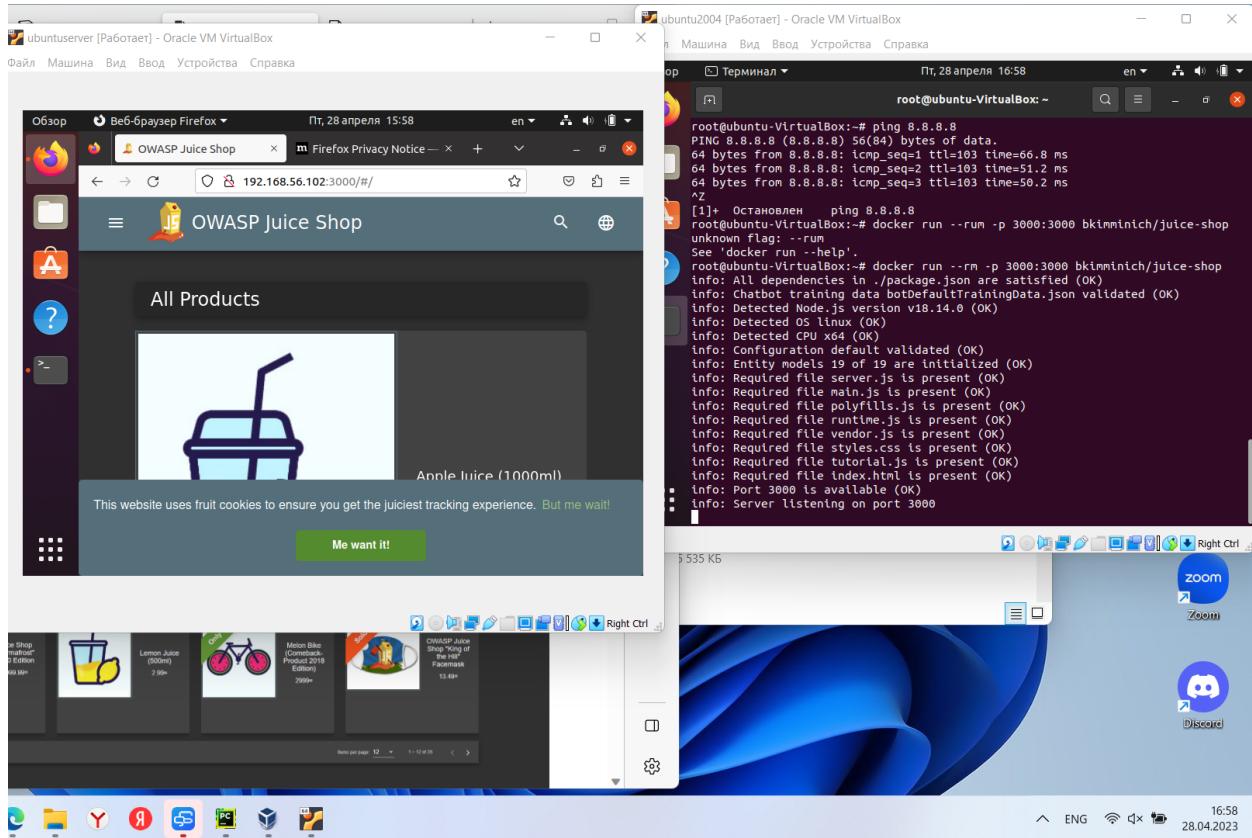
## Интерфейсы WEB Server



# Подключение к WEB-серверу

по

## IP WAF машины:



Обновление всех пакетов.

```
sudo apt update && sudo apt upgrade -y
```

## Установка последней версии Nginx на Ubuntu Server 22.04

Сначала необходимо удалить уже установленный **Nginx**.

Остановка процесса работы утилиты.

Используемая команда:

```
sudo systemctl stop nginx
```

После удаляется утилита

Используемая команда:

```
sudo apt-get purge nginx -y && sudo apt autoremove nginx -y
```

Удалив старую версию **Nginx**, можно начать процесс установки одного из двух **Nginx PPA (STABLE и MAINLINE)**.

В данной работе устанавливался **MAINLINE**.

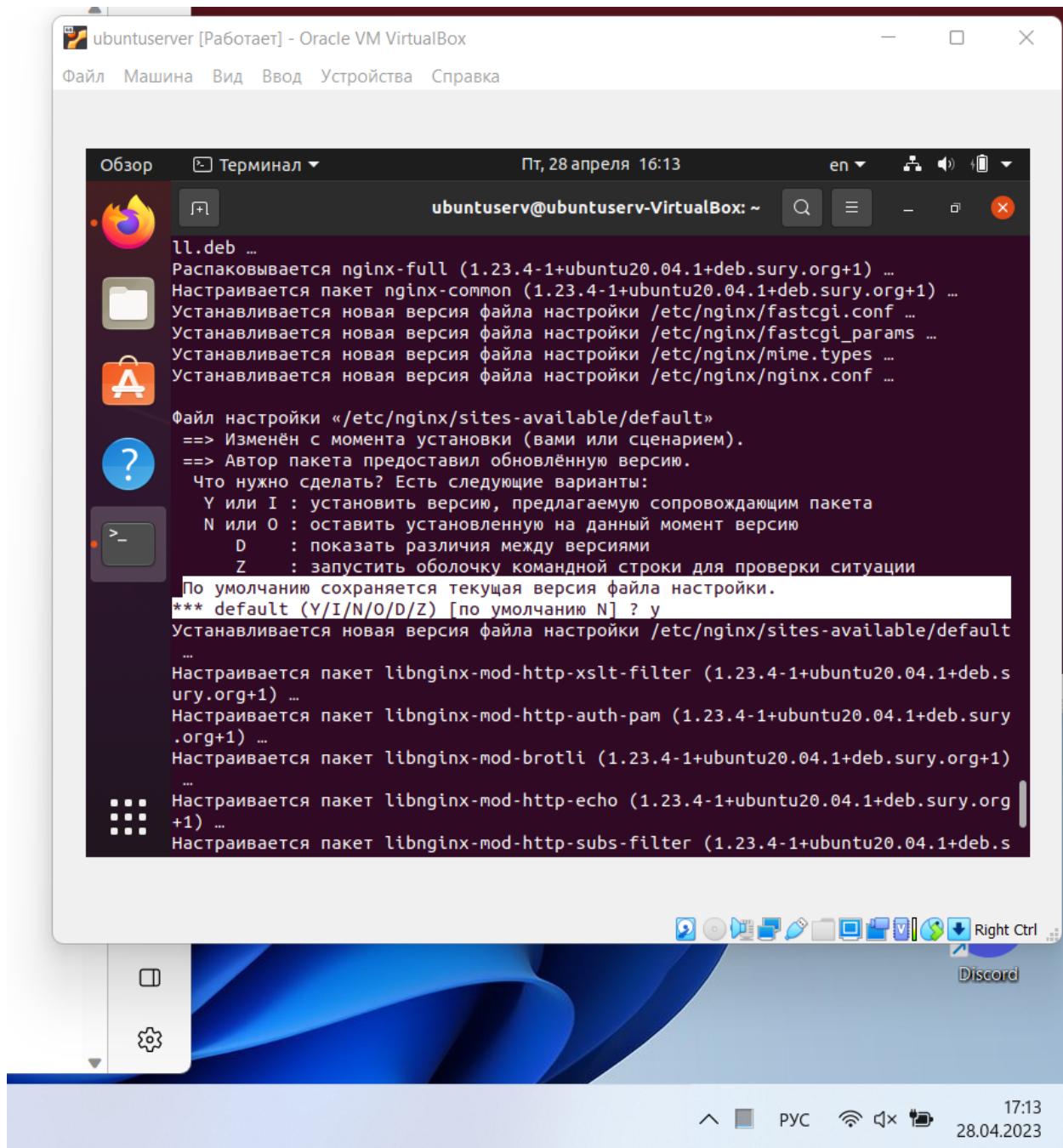
Используемая команда:

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

Теперь можно приступить к установке утилиты **Nginx**.

Используемая команда:

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



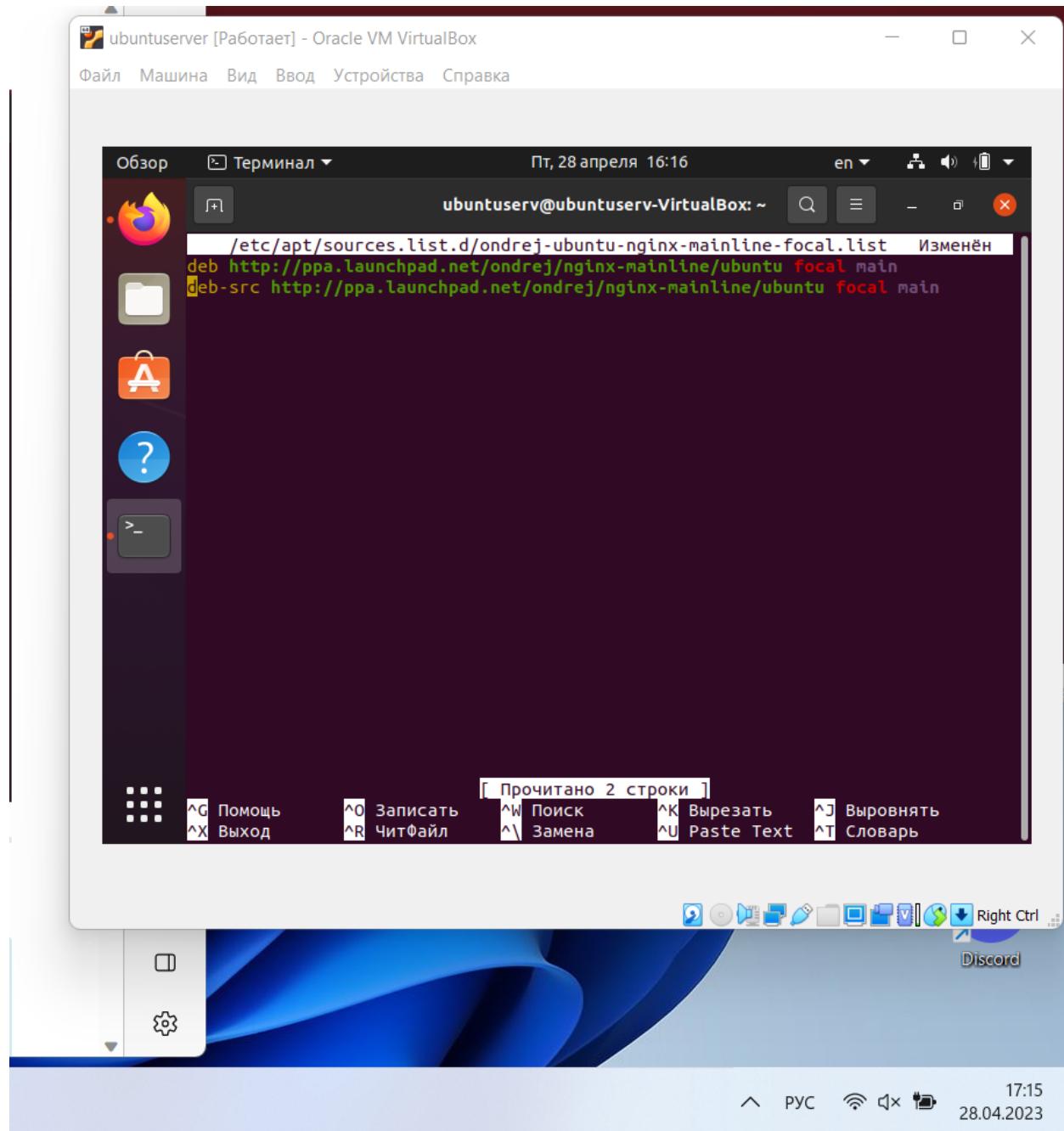
## Добавление исходного кода Nginx в репозиторий

Изначально исходный код не устанавливается при установке PPA.

Необходимо вручную включить это, чтобы загрузить исходный код Nginx для компиляции **Modsecurity**.

Используемая команда для открытия файла конфигурации:

```
sudo nano /etc/apt/sources.list.d/ondrej-ubuntu-nginx-mainline*.list
```



Обновление репозитория.

Используемая команда:

```
sudo apt update
```

## Загрузка Nginx Source

Для компилирования динамического модуля ModSecurity нужно скачать и становить исходный пакет в каталоге `/etc/local/src/nginx`.

Для этого создаётся и конфигурируется директория.

Используемая команда:

```
sudo mkdir /usr/local/src/nginx && cd /usr/local/src/nginx
```

Дополнительно - можно назначить разрешения для каталога.

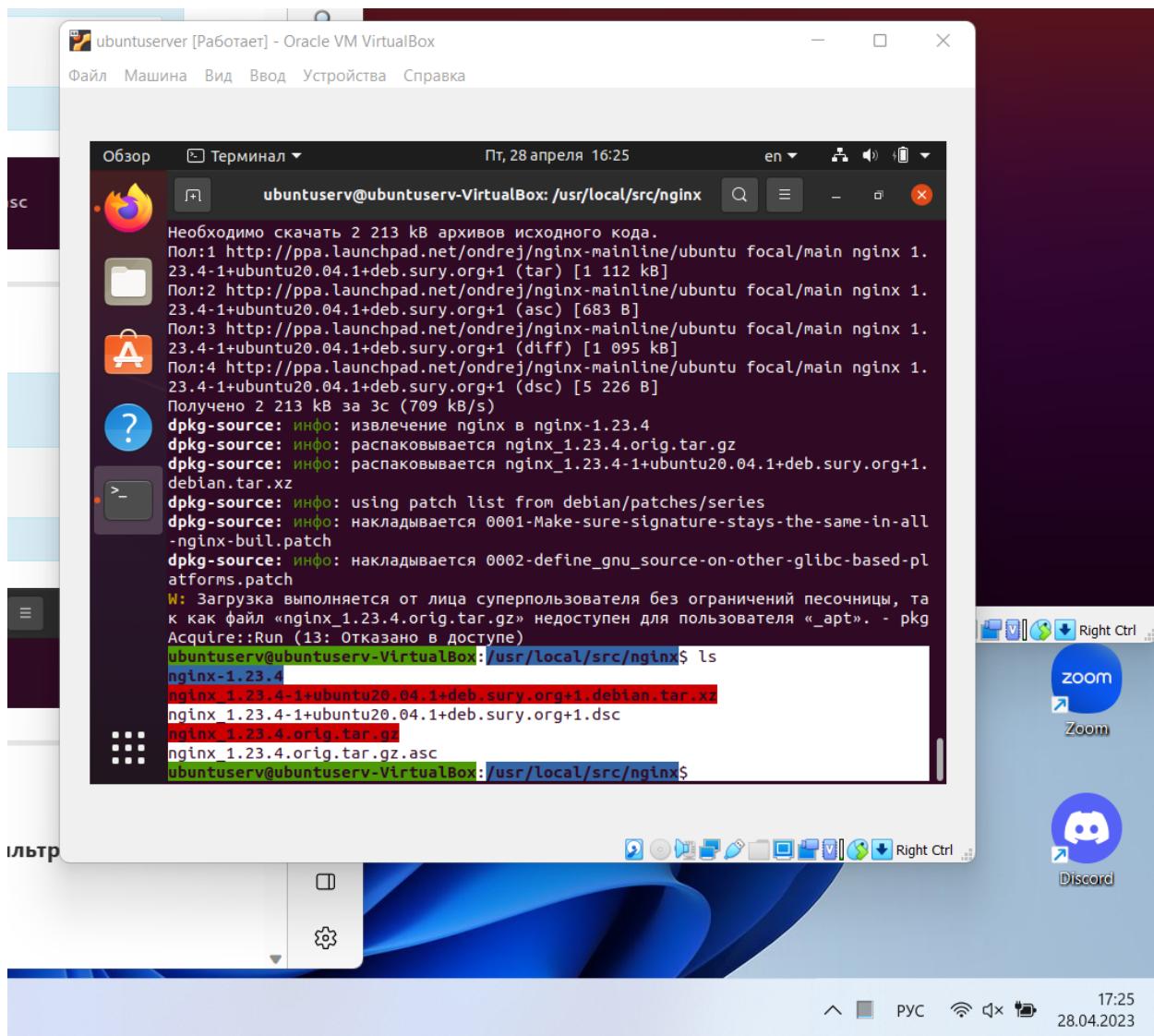
Используемая команда:

```
sudo chown root:ubuntuserv /usr/local/src/ -R
```

Загрузка исходного пакета.

Используемая команда:

```
sudo apt install dpkg-dev -y && sudo apt source nginx
```



## Установка libmodsecurity3 для ModSecurity

Пакет **libmodsecurity3** – это фактическая часть **WAF**, выполняющая **HTTP-фильтрацию** для веб-приложений.

До начала установки необходимо скачать и установить **Git**.

Используемая команда:

```
sudo apt install git -y
```

Далее клонируется **GIT** репозиторий **libmodsecurity3**.

Используемая команда:

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

После этого необходимо перейти в директорию `/usr/local/src/ModSecurity/`.

Используемая команда:

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

Для компиляции нужно установить следующие параметры.

Используемая команда:

```
sudo apt install gcc make build-essential autoconf automake libibtool libcurl4-openssl-dev liblua5.3-dev libfuzzy-dev ssdeep gettext pkg-config libpcre3 libpcre3-dev libxml2 libxml2-dev libcurl4 libgeoip-dev libyajl-dev doxygen -y
```

Установка следующих подмодулей **GIT**.

Используемая команда:

```
git submodule init
```

После обновление этих подмодулей.

Используемая команда:

```
git submodule update
```

## Создание среды ModSecurity

Создание среды.

Используемая команда:

```
./build.sh
```

Запуск команды конфигурирования.

Используемая команда:

```
./configure
```

Компиляция исходного кода **ModSecurity**.

Используемая команда:

```
make
```

После компиляции исходного кода запускается установку через терминал.

Используемая команда:

```
sudo make install
```

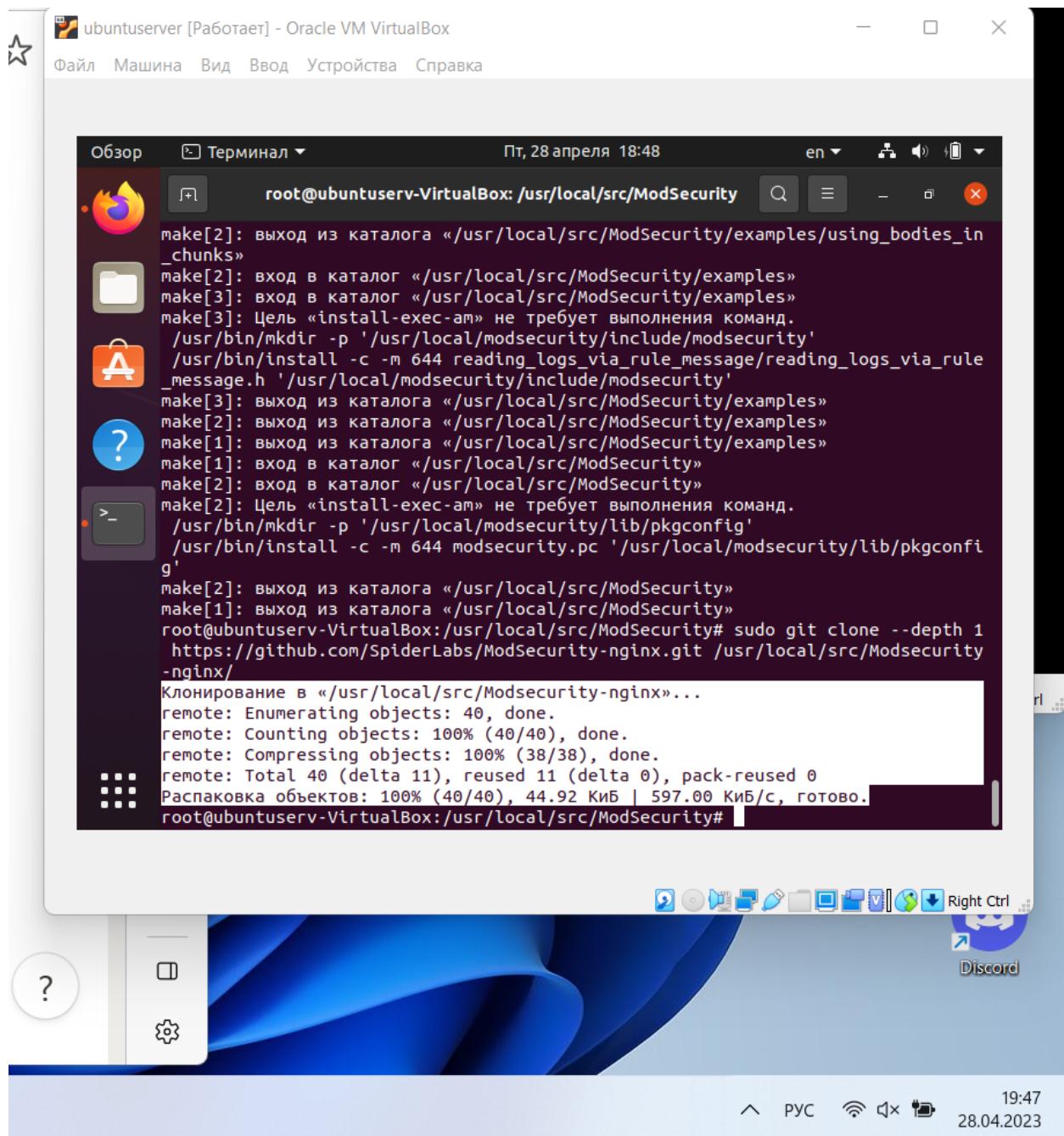
## Установка ModSecurity-nginx Connector

ModSecurity-nginx Connector — это точка соединения между **Nginx** и **libmodsecurity**, компонент, который взаимодействует между **Nginx** и **ModSecurity**.

Клонируется репозиторий с **Github**.

Используемая команда:

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



Далее переход в директорию `/usr/local/src/nginx/nginx-1.23.1`.

Используемая команда:

```
cd /usr/local/src/nginx/nginx-1.23.1
```

Установка необходимых параметров

Используемая команда:

```
sudo apt build-dep nginx && sudo apt install uuid-dev -y
```

Компиляция ModSecurity-nginx Connector

Используемая команда:

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

Создание динамические модули.

Используемая команда:

```
sudo make modules
```

Перенос и копирование динамического модуля в каталог `/usr/share/nginx/modules`.

Используемая команда:

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

## **Загружаем и настраиваем ModSecurity-nginx Connector с помощью Nginx.**

### **Включаем ModSecurity в nginx.conf**

С помощью текстового редактора открываем `/etc/nginx/nginx.conf`.

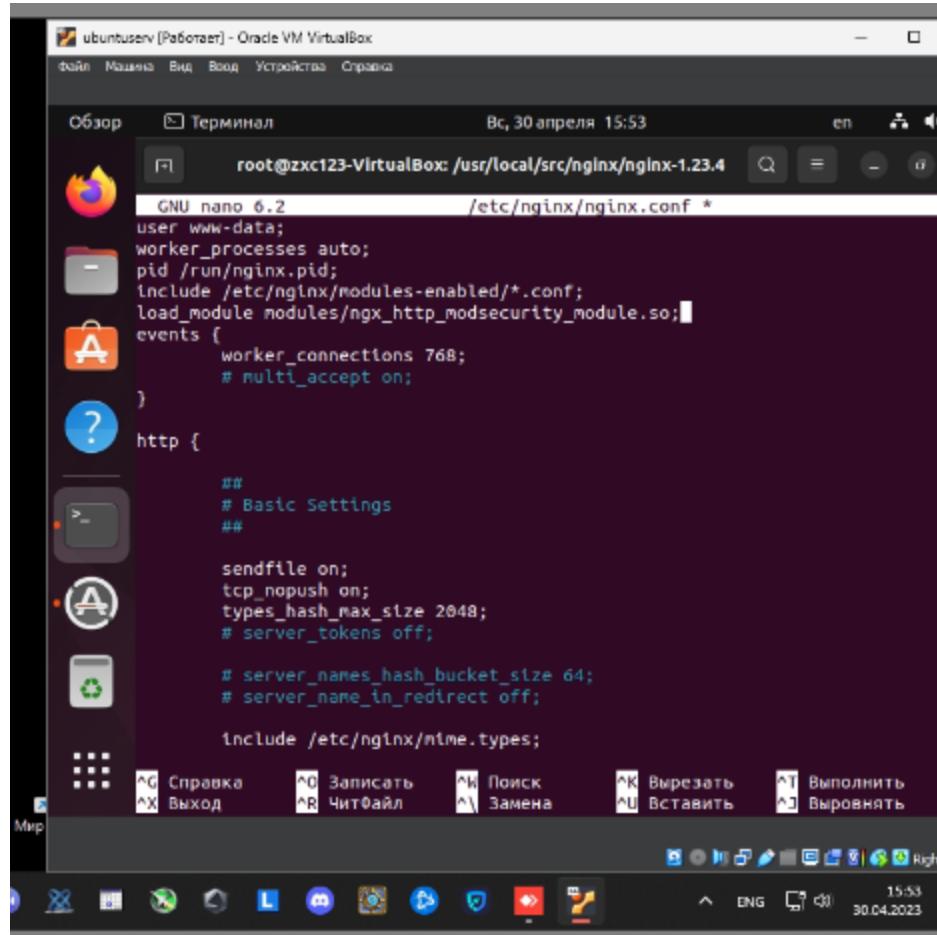
Используемая команда:

```
sudo nano /etc/nginx/nginx.conf
```

Добавляем строчку в верхушке файла.

Используемая строка:

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



```
GNU nano 6.2          /etc/nginx/nginx.conf *
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
load_module modules/ngx_http_modsecurity_module.so;■
events {
    worker_connections 768;
    # multi_accept on;
}
http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

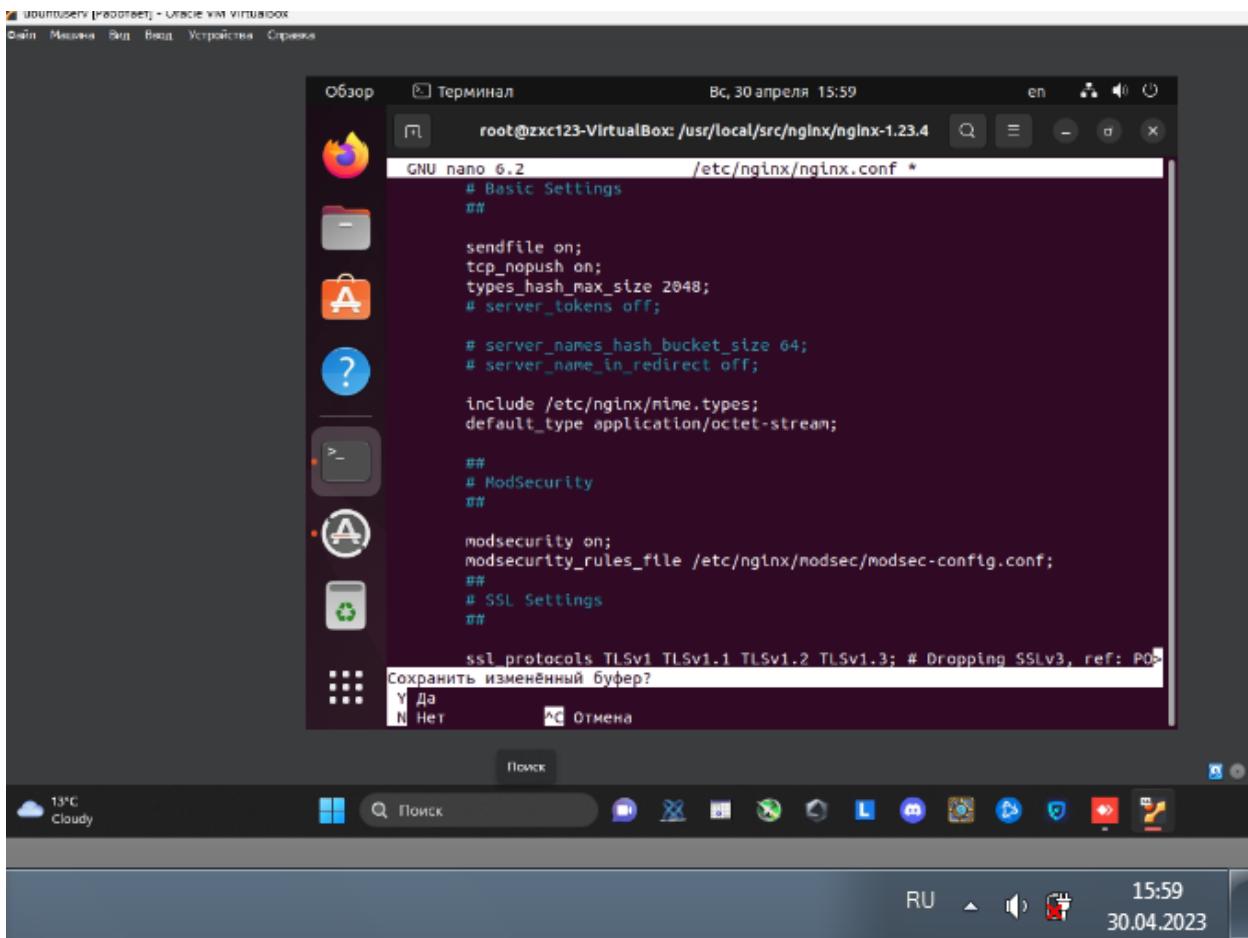
    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
}
```

Далее в раздел `HTTP{}` добавляем следующие строки:

Используемые строки:

```
modsecurity on;
modsecurity_rules_file /etc/nginx/modsec/modsec-config.conf;
```



## Создание и настройка директории и файлов для ModSecurity

Нам необходимо создать каталог для хранения файлов конфигурации и будущих правил OWASP CRS.

Создаем директорию и копируем из GIT директории файл конфигурации.

Используемые команды:

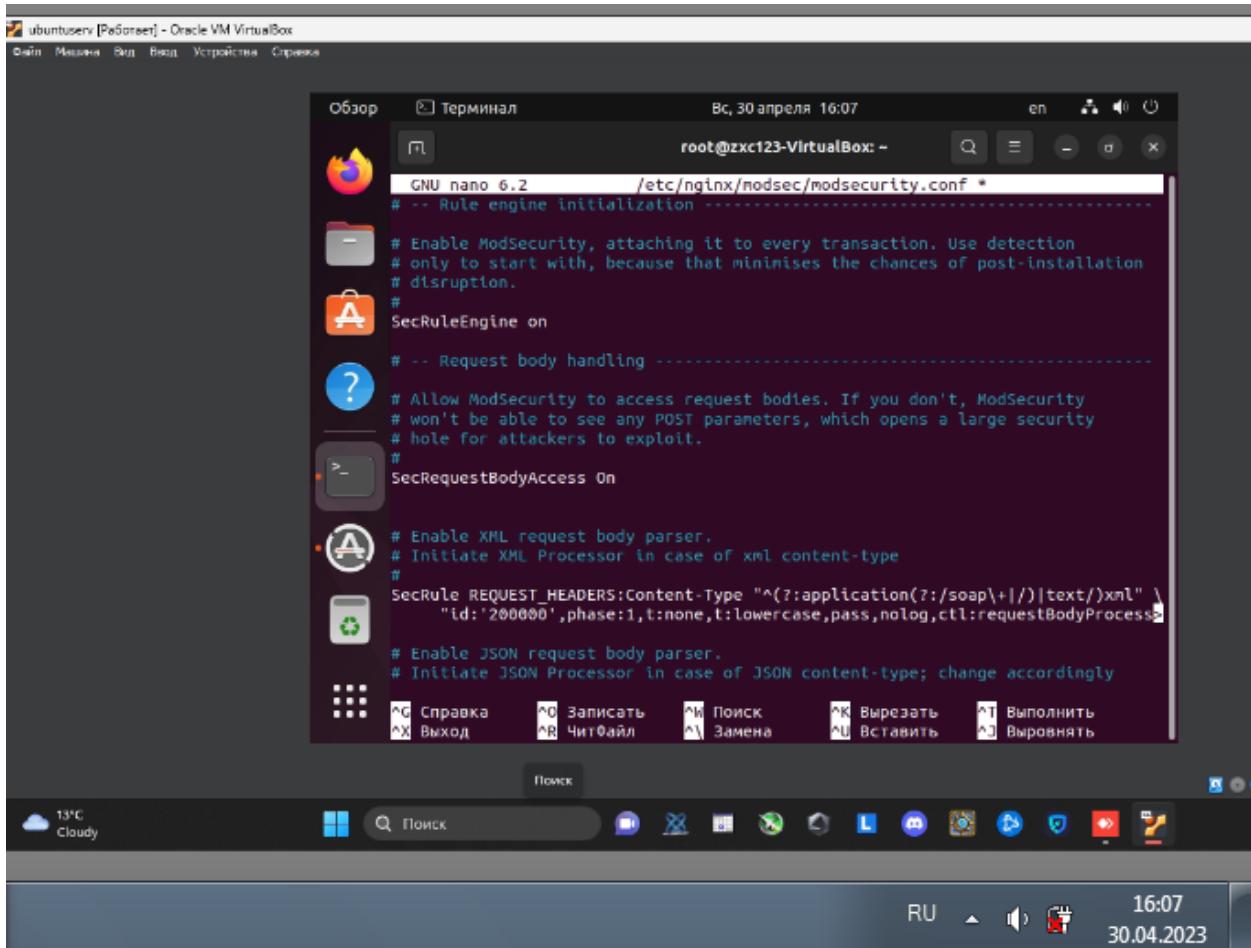
```
sudo mkdir /etc/nginx/modsec/
sudo cp /usr/local/src/ModSecurity/modsecurity.conf-recommended /etc/nginx/modsec/modsecurity.conf
```

Далее включаем работу правил в конфиге **modsecurity.conf**.

Используемая команда:

```
sudo nano /etc/nginx/modsec/modsecurity.conf
```

Изменяем параметр **SecRuleEngine DetectionOnly** на **SecRuleEngine On**.



Далее необходимо выполнить поиск по файлу параметра **SecAuditLogParts**.

Его мы заменяем.

Используемая строка:

```
SecAuditLogParts ABCEFHJKZ
```

Создадим файл с набором других файлов с правилами.

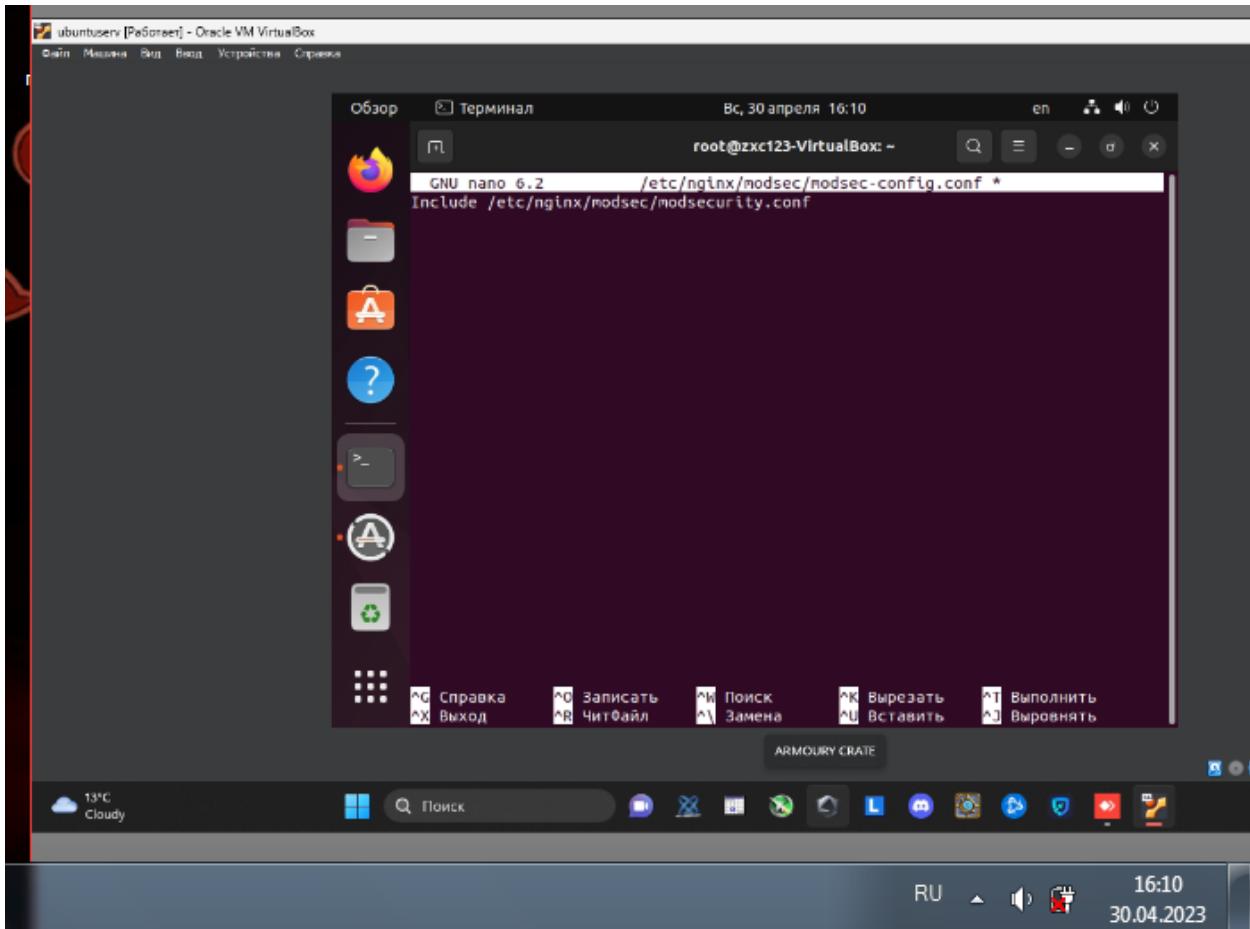
Используемая команда:

```
sudo nano /etc/nginx/modsec/modsec-config.conf
```

Добавляем туда файл с правилами.

Используемая команда:

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



Копируем файл unicode.mapping в другую директорию.

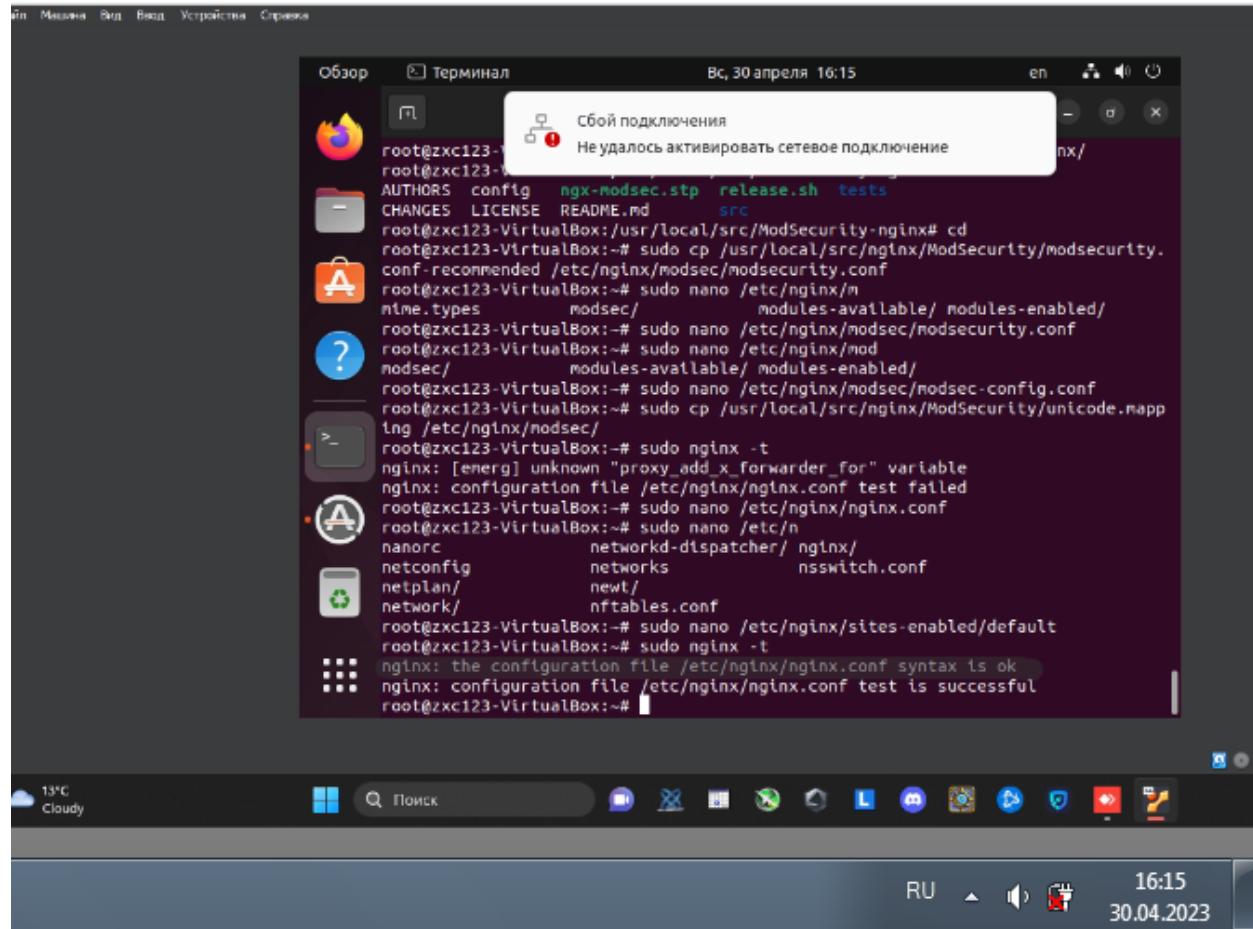
Используемая команда:

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

Протестируем службу **Nginx**.

Используемая команда:

```
sudo nginx -t
```



Всё работает корректно!

Перезапускаем **Nginx**.

Используемая команда:

```
sudo systemctl restart nginx
```

```
root@server:~# sudo systemctl restart nginx
root@server:~#
```

### 3) Настройка WAF с детектором OWASP top 10

Скачиваем архив **OWASP CRS 3.3.2**.

Используемая команда:

```
wget https://github.com/coreruleset/coreruleset/archive/refs/
tags/v3.3.2.zip
```

Скачиваем Unzip.

Используемая команда:

```
sudo apt install unzip -y
```

Используем Unzip, чтобы разархивировать скачанный **OWASP CRS 3.3.2**.

Используемая команда:

```
sudo unzip v3.3.2.zip -d /etc/nginx/modsec
```

Необходимо переименовать конфигурационный файл **OWASP CRS 3.3.2** и сделать резервную копию.

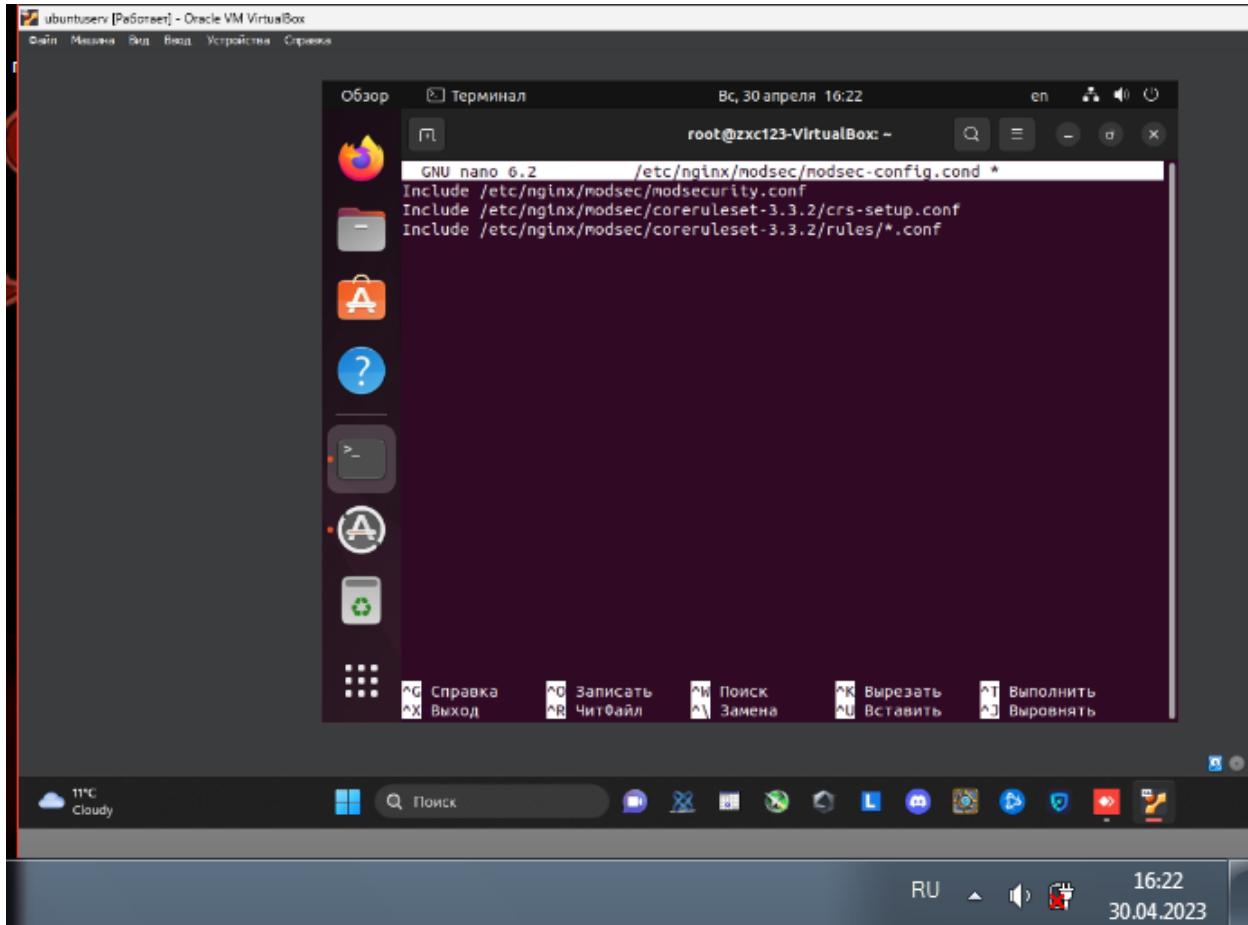
Используемая команда:

```
sudo cp /etc/nginx/modsec/coreruleset-3.3.2/crs-setup.conf.ex-
ample /etc/nginx/modsec/coreruleset-3.3.2/crs-setup.conf
```

Включаем файлы конфигурации в **modsec-config.conf**.

Используемые строки:

```
Include /etc/nginx/modsec/coreruleset-3.3.2/crs-setup.conf  
Include /etc/nginx/modsec/coreruleset-3.3.2/rules/*.conf
```



Протестируем службу **Nginx**.

Используемая команда:

```
sudo nginx -t
```

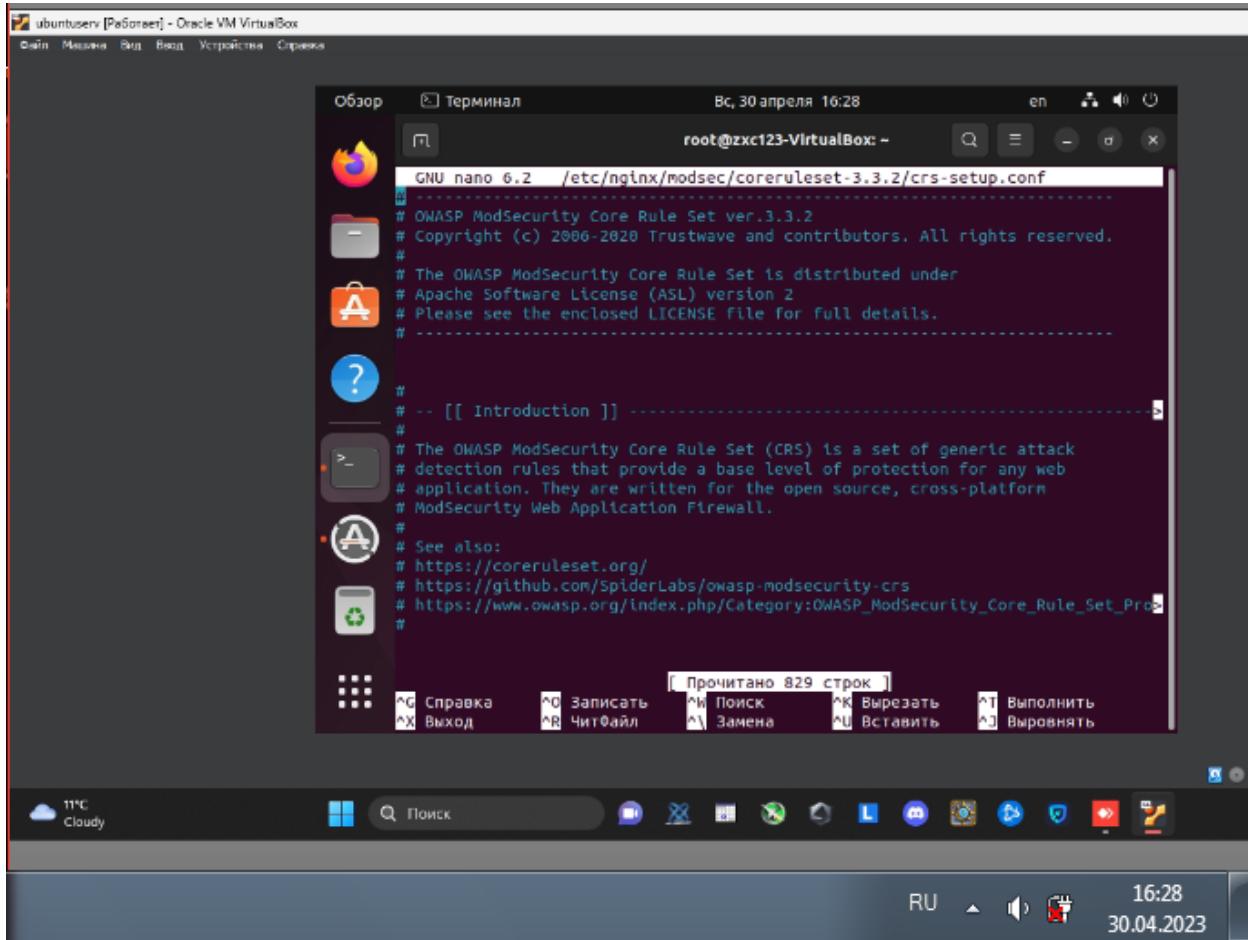
Перезапускаем **Nginx**.

Используемая команда:

```
sudo systemctl restart nginx
```

```
root@server:~# sudo systemctl restart nginx
root@server:~#
```

Содержание **crs-setup** файла.



```
GNU nano 6.2 /etc/nginx/modsec/coreruleset-3.3.2/crs-setup.conf
#
# OWASP ModSecurity Core Rule Set ver.3.3.2
#
# Copyright (c) 2006-2020 Trustwave and contributors. All rights reserved.
#
# The OWASP ModSecurity Core Rule Set is distributed under
# Apache Software License (ASL) version 2
# Please see the enclosed LICENSE file for full details.
#
#
# -- [[ Introduction ]]
#
# The OWASP ModSecurity Core Rule Set (CRS) is a set of generic attack
# detection rules that provide a base level of protection for any web
# application. They are written for the open source, cross-platform
# ModSecurity Web Application Firewall.
#
# See also:
# https://coreruleset.org/
# https://github.com/SpiderLabs/owasp-modsecurity-crs
# https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Pro
#
```

**4) Провести 3 атаки из списка OWASP top 10 и продемонстрировать работоспособность WAF (должны быть срабатывания правил WAF)**

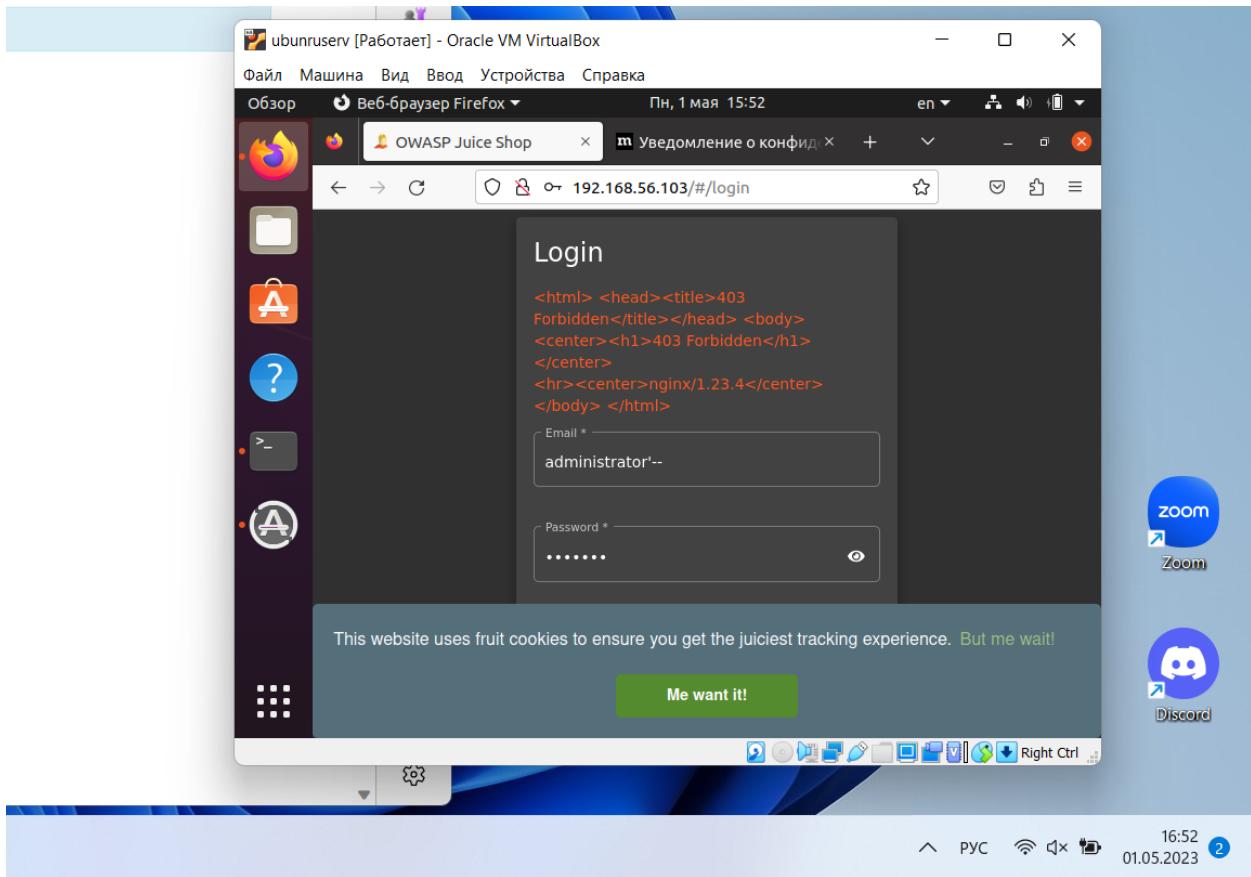
**SQL Injection**

Пытаемся обойти вход в систему.

Используемый логин:

```
administrator' --
```

Срабатывают правила.



Логи.

```
ubunrusev [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Обзор Терминал Пн, 1 мая 15:54 en
root@ubuntuserv-VirtualBox: /var/log/nginx
2023/05/01 15:39:57 [alert] 5049#5049: *6 open socket #14 left in connection 7
2023/05/01 15:39:57 [alert] 5049#5049: *259 open socket #3 left in connection 9
2023/05/01 15:39:57 [alert] 5049#5049: aborting
2023/05/01 15:39:57 [notice] 33374#33374: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/7/0)
2023/05/01 15:39:57 [notice] 33375#33375: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/7/0)
2023/05/01 15:47:56 [notice] 33457#33457: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/910/0)
2023/05/01 15:49:02 [alert] 33377#33377: *1 open socket #3 left in connection 3
2023/05/01 15:49:02 [alert] 33377#33377: *52 open socket #13 left in connection 4
2023/05/01 15:49:02 [alert] 33377#33377: aborting
2023/05/01 15:49:02 [notice] 33466#33466: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/910/0)
2023/05/01 15:49:03 [notice] 33467#33467: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/910/0)
2023/05/01 15:50:10 [error] 33469#33469: *16 [client 192.168.56.103] ModSecurity: Access denied with code 403 (phase 2). Matched "Operator `Ge` with parameter `5` against variable `TX:ANOMALY_SCORE` (Value: `8` ) [file "/etc/nginx/modsec/coreruleset-3.3.2/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "80"] [id "949110"] [rev ""] [msg "Inbound Anomaly Score Exceeded (Total Score: 8)"] [data ""] [severity "2"] [ver "OWASP_CRS/3.3.2"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "192.168.56.103"] [uri "/rest/user/login"] [unique_id "168294901024.694583"] [ref ""], client: 192.168.56.103, server: _, request: "POST /rest/user/login HTTP/1.1", host: "192.168.56.103", referer: "http://192.168.56.103/"
```

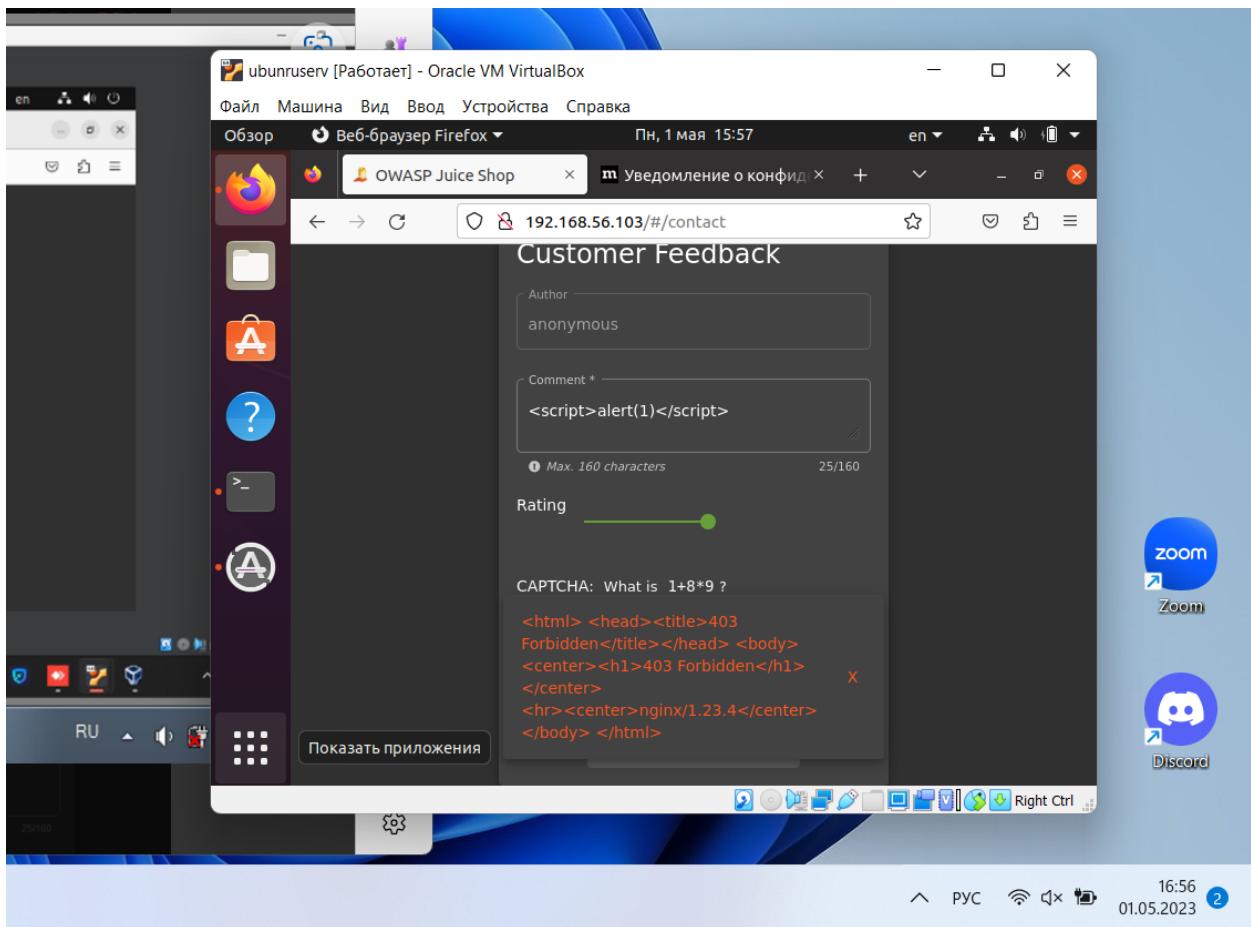
zoom  
Discord

16:53 01.05.2023 2

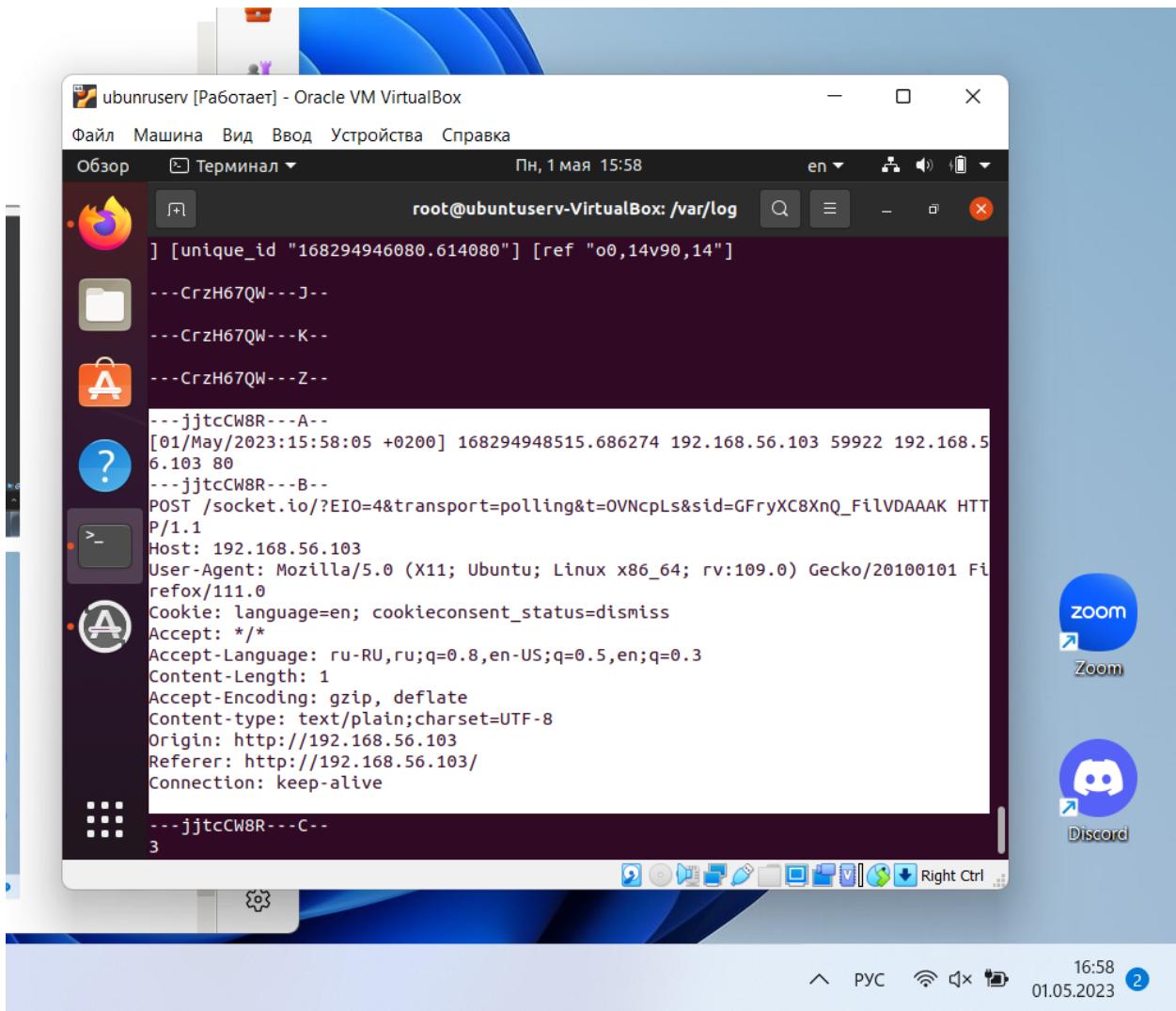
## XSS

Оставляем комментарий со скриптом.

Содержание комментария:



Логи.



RCE

