

Домашнее задание №3

Contents

Первая часть	2
Установить Apache и убедиться, что он работает через команду	2
Просмотреть страницу по URL-адресу для проверки рабочего состояния	2
Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Apache.....	3
Ставим mod_ssl и openssl.....	3
Генерим сертификат	4
Настраиваем права	4
Создаем отдельный файл конфигурации	4
Настраиваем файервол.....	5
Финальные проверки.....	5
Часть 2	8
Установить Nginx и убедиться, что он работает:	8
Просмотреть страницу по URL-адресу для проверки рабочего состояния	9
Настройте прямой и обратный прокси в Nginx для перенаправления запросов	9
Обратный (reverse) прокси	9
Прямой (forward) прокси	10
Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Nginx	11
Установить и настроить ModSecurity, в том числе настроить фильтрацию запросов для обеспечения безопасности от SQL-запросов	17
Установка ModSecurity.....	17
Настройка ModSecurity	17
Создание файла main.conf.....	18
Продемонстрировать успешную блокировку SQL-инъекций.....	19
Демонстрация работы Modsecurity (в примере показана XSS, вам необходимо показать на SQLi)	19
Закомментировать правило в файле конфигурации Modsecurity и показать, что Nginx возвращает код 200	20
Дополнительная проверка различных паттернов.....	20

Первая часть

Установить Apache и убедиться, что он работает через команду

```
sudo systemctl status apache2.service
```

У меня Fedora, там вместо apache есть сервис httpd.

```
vladimirkuryndin@fedora:~$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/service.d
            └─10-timeout-abort.conf
   Active: active (running) since Mon 2025-09-29 20:36:13 MSK; 26s ago
 Invocation: ed2a06aa72204b998bed7eae27afb037
    Docs: man:httpd.service(8)
   Main PID: 3974 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
   Tasks: 177 (limit: 9408)
  Memory: 18.4M (peak: 18.7M)
     CPU: 70ms
   CGroup: /system.slice/httpd.service
           └─3974 /usr/sbin/httpd -DFOREGROUND
             └─3975 /usr/sbin/httpd -DFOREGROUND
               └─3977 /usr/sbin/httpd -DFOREGROUND
                 └─3978 /usr/sbin/httpd -DFOREGROUND
                   └─3980 /usr/sbin/httpd -DFOREGROUND

Sep 29 20:36:13 fedora systemd[1]: Starting httpd.service - The Apache HTTP Server...
Sep 29 20:36:13 fedora httpd[3974]: httpd.service: Referenced but unset environment variable evaluates to an empty st
Sep 29 20:36:13 fedora httpd[3974]: AH00558: httpd: Could not reliably determine the server's fully qualified domain na
Sep 29 20:36:13 fedora httpd[3974]: Server configured, listening on: port 80
Sep 29 20:36:13 fedora systemd[1]: Started httpd.service - The Apache HTTP Server.
lines 1-24/24 (END)...skipping...
```

Просмотреть страницу по URL-адресу для проверки рабочего состояния

Для начала разрешим в файерволле

```
vladimirkuryndin@fedora:~$ rpm -q firewalld
firewalld-2.3.1-1.fc42.noarch
vladimirkuryndin@fedora:~$ sudo firewall-cmd --add-service=http --permanent
[sudo] password for vladimirkuryndin:
success
vladimirkuryndin@fedora:~$ sudo firewall-cmd --reload
success
```

Создадим пустую страницу, чтобы не возвращался 403

```
vladimirkuryndin@fedora:~$ echo 'Hello from fedora Apache' | sudo tee /var/www/html/index.html
Hello from fedora Apache
vladimirkuryndin@fedora:~$ curl -I http://localhost
HTTP/1.1 200 OK
Date: Mon, 29 Sep 2025 17:46:05 GMT
Server: Apache/2.4.64 (Fedora Linux)
Last-Modified: Mon, 29 Sep 2025 17:46:00 GMT
ETag: "19-63ff43563ad76"
Accept-Ranges: bytes
Content-Length: 25
Content-Type: text/html; charset=UTF-8
```

Теперь проверим магическую страницу

```
vladmirkuryndin@fedora:~$ curl -I http://localhost
Content-Length: 8484
Content-Type: text/html; charset=UTF-8

vladmirkuryndin@fedora:~$ echo 'Hello from fedora Apache' | sudo tee /var/www/html/index.html
Hello from fedora Apache
vladmirkuryndin@fedora:~$ curl -I http://localhost
HTTP/1.1 200 OK
Date: Mon, 29 Sep 2025 17:46:05 GMT
Server: Apache/2.4.64 (Fedora Linux)
Last-Modified: Mon, 29 Sep 2025 17:46:00 GMT
ETag: "19-63ff43563ad76"
Accept-Ranges: bytes
Content-Length: 25
Content-Type: text/html; charset=UTF-8

vladmirkuryndin@fedora:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:e6:d8:76 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname enx000c29e6d876
    inet 192.168.179.130/24 brd 192.168.179.255 scope global dynamic noprefixroute ens160
        valid_lft 1223sec preferred_lft 1223sec
    inet6 fe80::218f:8a57:503d:63e9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
vladmirkuryndin@fedora:~$
```

Browser view: Not secure 192.168.179.130
Bug 227139 Паден... CSP Win ViPNet CS.
Hello from fedora Apache

Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Apache

Ставим mod_ssl и openssl

```
vladmirkuryndin@fedora:~$ sudo dnf install -y mod_ssl openssl
Updating and loading repositories:
Repositories loaded.
Package "openssl-1:3.2.4-4.fc42.x86_64" is already installed.

Package Arch Version Repository Size
Installing:
mod_ssl x86_64 1:2.4.64-2.fc42 updates 249.2 KiB

Transaction Summary:
Installing: 1 package

Total size of inbound packages is 105 KiB. Need to download 105 KiB.
After this operation, 249 KiB extra will be used (install 249 KiB, remove 0 B).
[1/1] mod_ssl-1:2.4.64-2.fc42.x86_64 100% | 146.1 KiB/s | 105.4 KiB | 00m01s
-----
[1/1] Total 100% | 85.0 KiB/s | 105.4 KiB | 00m01s
Running transaction
[1/3] Verify package files 100% | 333.0 B/s | 1.0 B | 00m00s
[2/3] Prepare transaction 100% | 5.0 B/s | 1.0 B | 00m00s
[3/3] Installing mod_ssl-1:2.4.64-2.fc42.x86_64 100% | 149.6 KiB/s | 250.9 KiB | 00m02s
Complete!
```

Генерим сертификат

[illegible]

Проверяем

```
vladimirkuryndin@fedora:~$ openssl x509 -in /etc/pki/tls/certs/selfsigned.crt -noout -subject -issuer
subject=C=RU, ST=Moscow, L=Moscow, O=VladimirKuryndin, CN=192.168.179.130
issuer=C=RU, ST=Moscow, L=Moscow, O=VladimirKuryndin, CN=192.168.179.130
```

Настраиваем права

```
vladmirkuryndin@fedora:~$ sudo chown root:root /etc/pki/tls/private/selfsigned.key
vladmirkuryndin@fedora:~$ sudo chmod 600 /etc/pki/tls/private/selfsigned.key
vladmirkuryndin@fedora:~$ sudo restorecon -Rv /etc/pki/tls
```

Создаем отдельный файл конфигурации

После множества попыток все отлаить он получился таким

GNU nano 8.3

```
/etc/httpd/conf.d/ssl-localhost.conf
```

Listen 443 https

<VirtualHost *:443>

ServerName 192.168.179.130

DocumentRoot "/var/www/html"

SSL Engine on

```
SSLCertificateFile /etc/pki/tls/certs/selfsigned.crt
```

SSLCertificateKeyFile /etc/pki/tls/private/selfsigned.key

using Fedora system cryptopolicies

SSLCipherSuite PROFILE=SYSTEM

SSLProxyCipherSuite PROFILE=SYSTEM

SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1

<Directory "/var/www/html">

AllowOverride None

Require all granted

</Directory>

ErrorLog logs/ssl_error_log

CustomLog logs/ssl_access_log combined

</VirtualHost>

Настраиваем фаерволл

```
vladmirkuryndin@fedora:~$ sudo firewall-cmd --add-service=https ==permanent
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: ==permanent
vladmirkuryndin@fedora:~$ sudo firewall-cmd --add-service=https --permanent
success
vladmirkuryndin@fedora:~$ sudo firewall-cmd --reload
success
```

Финальные проверки

Проверяем, работает ли порт 443

vladmirkuryndin@fedora:~\$ sudo ss -tlnp | grep httpd

LISTEN 0 511 *:443 *.*

users:(("httpd",pid=5469,fd=6),("httpd",pid=5468,fd=6),("httpd",pid=5467,fd=6),("httpd",pid=5466,fd=6),("httpd",pid=5464,fd=6))

LISTEN 0 511 *:80 *.*

users:(("httpd",pid=5469,fd=4),("httpd",pid=5468,fd=4),("httpd",pid=5467,fd=4),("httpd",pid=5466,fd=4),("httpd",pid=5464,fd=4))

Собственно проверяем, что работает https

vladimirkuryndin@fedora:~\$ curl -vkl https://192.168.179.130

- * Trying 192.168.179.130:443...
- * ALPN: curl offers h2,http/1.1
- * TLSv1.3 (OUT), TLS handshake, Client hello (1):
- * TLSv1.3 (IN), TLS handshake, Server hello (2):
- * TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
- * TLSv1.3 (IN), TLS handshake, Certificate (11):
- * TLSv1.3 (IN), TLS handshake, CERT verify (15):
- * TLSv1.3 (IN), TLS handshake, Finished (20):
- * TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
- * TLSv1.3 (OUT), TLS handshake, Finished (20):
- * SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / x25519 / RSASSA-PSS
- * ALPN: server accepted http/1.1
- * Server certificate:
- * subject: C=RU; ST=Moscow; L=Moscow; O=VladimirKuryndin; CN=192.168.179.130
- * start date: Sep 29 18:03:16 2025 GMT
- * expire date: Sep 29 18:03:16 2026 GMT
- * issuer: C=RU; ST=Moscow; L=Moscow; O=VladimirKuryndin; CN=192.168.179.130
- * SSL certificate verify result: self-signed certificate (18), continuing anyway.
- * Certificate level 0: Public key type RSA (4096/152 Bits/secBits), signed using sha256WithRSAEncryption
- * Connected to 192.168.179.130 (192.168.179.130) port 443
- * using HTTP/1.x
- > HEAD / HTTP/1.1
- > Host: 192.168.179.130
- > User-Agent: curl/8.11.1
- > Accept: */*
- >
- * Request completely sent off

* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):

* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):

< HTTP/1.1 200 OK

HTTP/1.1 200 OK

< Date: Mon, 29 Sep 2025 18:54:19 GMT

Date: Mon, 29 Sep 2025 18:54:19 GMT

< Server: Apache/2.4.64 (Fedora Linux) OpenSSL/3.2.4

Server: Apache/2.4.64 (Fedora Linux) OpenSSL/3.2.4

< Last-Modified: Mon, 29 Sep 2025 17:46:00 GMT

Last-Modified: Mon, 29 Sep 2025 17:46:00 GMT

< ETag: "19-63ff43563ad76"

ETag: "19-63ff43563ad76"

< Accept-Ranges: bytes

Accept-Ranges: bytes

< Content-Length: 25

Content-Length: 25

< Content-Type: text/html; charset=UTF-8

Content-Type: text/html; charset=UTF-8

<

* Connection #0 to host 192.168.179.130 left intact

Проверяем в Windows

← ↻ Not secure https://192.168.179.130

📄 🌐 Bug 227139 Паден... 🌐 CSP Win ViPNet CS

Hello from fedora Apache

Certificate Viewer: 192.168.179.130

General Details

Issued To

Common Name (CN)	192.168.179.130
Organization (O)	VladimirKuryndin
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	192.168.179.130
Organization (O)	VladimirKuryndin
Organizational Unit (OU)	<Not Part Of Certificate>

Validity Period

Issued On	Monday, September 29, 2025 at 9:03:16 PM
Expires On	Tuesday, September 29, 2026 at 9:03:16 PM

SHA-256 Fingerprints

Certificate	3635ac9fe4f991523bb208a828b0fc1a66b95949db48fea6f31c7b4cd87fdb32
Public Key	097b1dceb95f52fa3d43a03188f0ccc11ef7c88a6fd9343f0733cacd93218e73

Часть 2

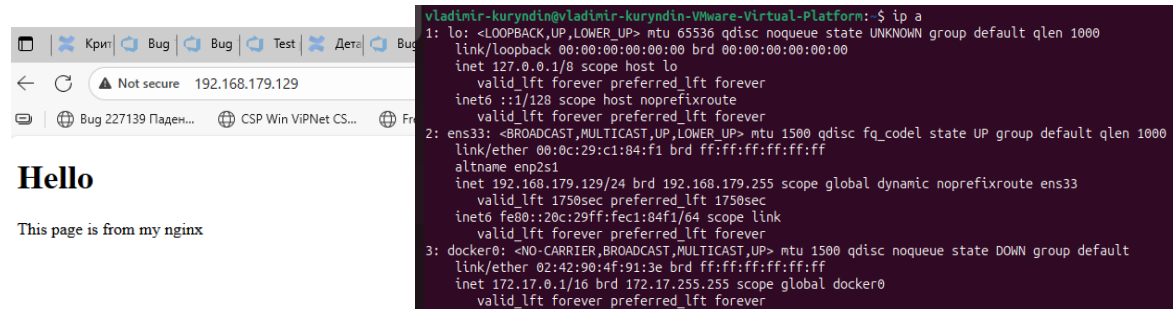
Установить Nginx и убедиться, что он работает:

`sudo systemctl status nginx`

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: $ sudo systemctl enable --now nginx && systemctl status nginx --no-pager
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-09-30 16:01:57 +04; 20min ago
     Docs: man:nginx(8)
    Main PID: 9052 (nginx)
      Tasks: 3 (limit: 4540)
    Memory: 2.4M (peak: 5.1M)
       CPU: 17ms
    CGroup: /system.slice/nginx.service
            └─9052 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
               └─9053 "nginx: worker process"
                  └─9054 "nginx: worker process"

Sep 30 16:01:57 vladimir-kuryndin-VMware-Virtual-Platform systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Sep 30 16:01:57 vladimir-kuryndin-VMware-Virtual-Platform systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -I http://localhost
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Tue, 30 Sep 2025 12:22:51 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Tue, 30 Sep 2025 12:01:56 GMT
Connection: keep-alive
ETag: "68dbc6b4-267"
Accept-Ranges: bytes
```


Просмотреть страницу по URL-адресу для проверки рабочего состояния



Настройте прямой и обратный прокси в Nginx для перенаправления запросов

Обратный (reverse) прокси

Поднимем простой сервер на Python

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ nohup python3 -m http.server 8080 --directory /srv/app >/tmp/app.log 2>&1 &
[1] 4541
```

Проверим, что он работает

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -I http://127.0.0.1:8080
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.12.3
Date: Tue, 30 Sep 2025 12:59:07 GMT
Content-type: text/html
Content-Length: 32
Last-Modified: Tue, 30 Sep 2025 12:56:39 GMT
```

Проверим сам прокси

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -I http://192.168.179.129/app/
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Tue, 30 Sep 2025 14:27:00 GMT
Content-Type: text/html
Content-Length: 484
Connection: keep-alive
Last-Modified: Tue, 30 Sep 2025 14:20:42 GMT

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -I http://192.168.179.129/app/test.html
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Tue, 30 Sep 2025 14:27:10 GMT
Content-Type: text/html
Content-Length: 228
Connection: keep-alive
Last-Modified: Tue, 30 Sep 2025 14:20:59 GMT
```

Клиент → http://192.168.179.129/...

└─ если путь начинается НЕ с /app → Nginx читает файл с диска /var/www/html (без прокси)

└─ если путь начинается с /app → Nginx проксирует на 127.0.0.1:8080

└─ из-за конечного "/" в проху_pass префикс /app/ отрезается:

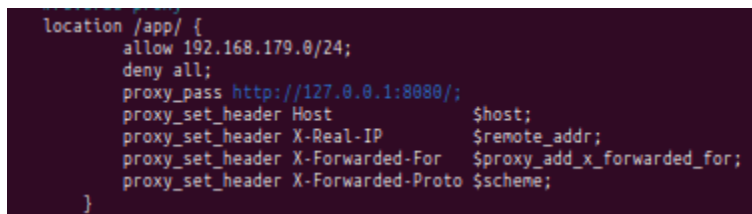
/app/foo → backend получает /foo

/app/docs/ → backend получает /docs/

В конфиг nginx добавили:

#reverse proxy

```
location /app/ {  
    proxy_pass http://127.0.0.1:8080/  
    allow 192.168.179.0/24  
  
    deny all;  
  
    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;  
}
```



```
location /app/ {  
    allow 192.168.179.0/24;  
    deny all;  
    proxy_pass http://127.0.0.1:8080/  
    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;  
}
```

Прямой (forward) прокси

В конфиг nginx добавляем

```
server {  
  
    listen 8888;  
  
    # DNS-resolvers for proxy_pass with variables  
    resolver 127.0.0.53 valid=30s ipv6=off;  
    resolver_timeout 5s;  
  
    # optional: disable access to the proxy for all hosts outside my network  
    allow 192.168.179.0/24;  
  
    deny all;
```

```

# all requests will be proxied to the URL, which came from the client
location / {

    proxy_pass $scheme://$http_host$request_uri;

    proxy_set_header Host      $http_host;

    proxy_set_header X-Real-IP $remote_addr;

}
}

```

```

#forward proxy
server {
    listen 8888;

    # DNS-resolvers for proxy_pass with variables
    resolver 127.0.0.53 valid=30s ipv6=off;
    resolver_timeout 5s;

    # optional: disable access to the proxy for all hosts outside my network
    allow 192.168.179.0/24;
    deny all;

    # all requests will be proxied to the URL, which came from the client
    location / {
        proxy_pass $scheme://$http_host$request_uri;
        proxy_set_header Host      $http_host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

Проверяем

```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -x http://192.168.179.129:8888 -I http://example.com
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Wed, 01 Oct 2025 16:58:33 GMT
Content-Type: text/html
Connection: keep-alive
ETag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
Last-Modified: Mon, 13 Jan 2025 20:11:20 GMT
Cache-Control: max-age=86000

```

Для обеспечения безопасности данных настроить HTTPS-подключение с использованием самоподписанных SSL-сертификатов в Nginx
Создаем конфиг для OpenSSL для выпуска сертификата со всеми нужными нам полями

```
GNU nano 7.2
[ req ]
default_bits      = 4096
prompt            = no
default_md         = sha256
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
C   = RU
ST  = MOSCOW
L   = MOSCOW
O   = Local
OU  = IT
CN  = 192.168.179.129

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
IP.1   = 192.168.179.129
DNS.1  = 192.168.179.129
```

Я решил сделать сертификат с SAN и использовать корневой центр сертификации CA (чтобы на windows мой Edge корректно обработал сертификат, учесть неточности в выпуске сертификатов при настройке Apache (см выше))

Далее создаем ключ СА и сертификат СА (корневой)

```
vladimir-kuryndin@vLadimir-kuryndin-VMware-Virtual-Platform:~$ openssl genrsa -out /etc/nginx/ssl/myCA.key 4096 && openssl req -x509 -new -nodes -key /etc/nginx/ssl/myCA.key -sha256 -days 3650 -out /etc/nginx/ssl/myCA.pem -subj "/C=RU/ST=MOSCOW/L=MOSCOW/O=Local-CA/OU=IT/CN=Local-Test-CA"
```

Делаем CSR и ключ сервера по нашему файлу конфигурации OpenSSL

[illegible]

Подписываем CSR нашим СА и получаем доверенный серверный сертификат

```

vladlntr-kuryndin@vladlntr-kuryndin-Vmware-Virtual-Platform:~$ openssl x509 -req -in /etc/nginx/ssl/server.csr -CA /etc/nginx/ssl/myCA.pem -CAkey /etc/nginx/ssl/myC
A.key -CAcreateserial -out /etc/nginx/ssl/server.crt -days 825 -sha256 -extfile /etc/nginx/ssl/openssl_local.cnf -extensions req_ext
Certificate request self-signature ok
subject=C = RU, ST = MOSCOW, L = MOSCOW, O = Local, OU = IT, CN = 192.168.179.129

```

Проверяем, что SAN присутствует

```

Vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ openssl s_x09 -in /etc/nginx/ssl/server.crt -noout -text | grep -A1 -i "Subject Alternative Name"
X509v3 Subject Alternative Name:
IP Address:192.168.179.129, DNS:192.168.179.129

```

Также проверим ,что с сертификатом все ок

```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ openssl x509 -in /etc/nginx/ssl/server.crt -noout -text | sed -n '/Subject:/,/X509v3 Subject Alternat
ive Name:/p'
    Subject: C = RU, ST = MOSCOW, L = MOSCOW, O = Local, OU = IT, CN = 192.168.179.129
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (4096 bit)
      Modulus:
        00:a3:56:f7:f0:a3:8d:2f:d0:e4:75:43:28:20:40:
        a7:10:56:78:82:bd:7d:a3:4d:c9:50:94:9a:a3:d3:
        e0:a3:d4:aa:27:8c:5c:da:69:f6:2c:01:6f:b4:cb:
        71:f8:e5:06:48:e2:a9:01:a8:9c:7c:2b:72:c1:54:
        61:65:31:f5:cb:b8:e8:36:9c:7d:7e:34:04:53:3d:
        72:8c:16:02:7b:b9:76:a2:25:7c:50:60:17:fc:36:
        79:2e:87:6e:82:bf:57:37:3d:02:3f:af:87:d0:0b:
        7d:49:0f:91:11:12:e7:8a:27:03:57:b8:c4:a7:d4:
        64:b7:ca:d1:b0:a5:4d:7a:33:0d:b2:76:be:0a:54:
        3b:73:56:df:e8:7b:68:22:55:71:93:1f:7f:4a:28:

```

Добавим новую секцию в файл конфигурации nginx

```

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name 192.168.179.129;

    # my certificates
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;

    # basic TLS settings
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;

    # ksite root the same as HTTP
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;

    # main page http
    location / {
        try_files $uri $uri/ =404;
    }

    # reverse proxy via HTTPS
    location /app/ {
        allow 192.168.179.0/24;
        deny all;

        proxy_pass http://127.0.0.1:8080/;
        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;
    }
}

```

Проверка

```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -vki https://192.168.179.129/
* Trying 192.168.179.129:443...
* Connected to 192.168.179.129 (192.168.179.129) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: C=RU; ST=MOSCOW; L=MOSCOW; O=Local; OU=IT; CN=192.168.179.129
* start date: Oct  1 17:19:08 2025 GMT
* expire date: Jan  4 17:19:08 2028 GMT
* issuer: C=RU; ST=MOSCOW; L=MOSCOW; O=Local-CA; OU=IT; CN=Local-Test-CA
* SSL certificate verify result: unable to get local issuer certificate (20), continuing anyway.
* Certificate level 0: Public key type RSA (4096/152 Bits/secBits), signed using sha256WithRSAEncryption
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://192.168.179.129/
* [HTTP/2] [1] [:method: HEAD]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: 192.168.179.129]
* [HTTP/2] [1] [:path: /]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> HEAD / HTTP/2
> Host: 192.168.179.129
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/2 200
HTTP/2 200
< server: nginx/1.24.0 (Ubuntu)
server: nginx/1.24.0 (Ubuntu)
< date: Wed, 01 Oct 2025 18:26:33 GMT
date: Wed, 01 Oct 2025 18:26:33 GMT
< content-type: text/html
content-type: text/html
< content-length: 169
content-length: 169
< last-modified: Tue, 30 Sep 2025 12:28:59 GMT
last-modified: Tue, 30 Sep 2025 12:28:59 GMT
< etag: "68dbcd0b-a9"
etag: "68dbcd0b-a9"
< accept-ranges: bytes
accept-ranges: bytes
<
* Connection #0 to host 192.168.179.129 left intact
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$

```

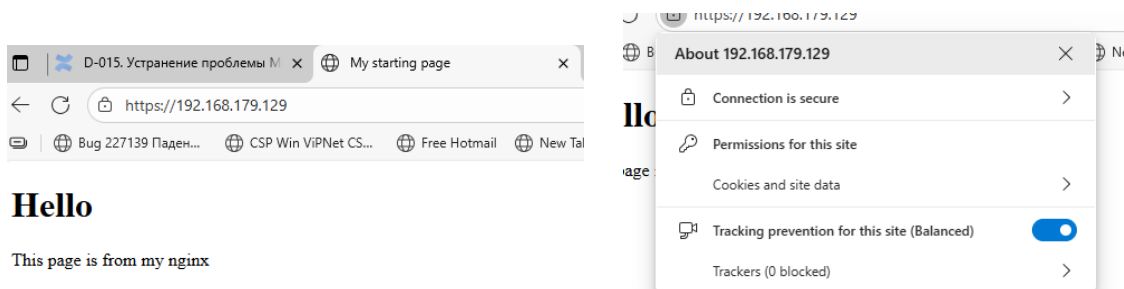
Проверка, что работает reverse proxy на /app/

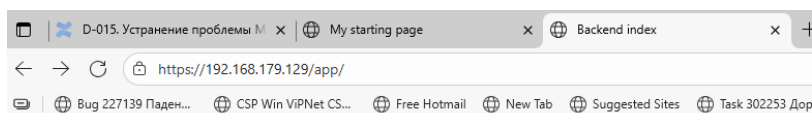
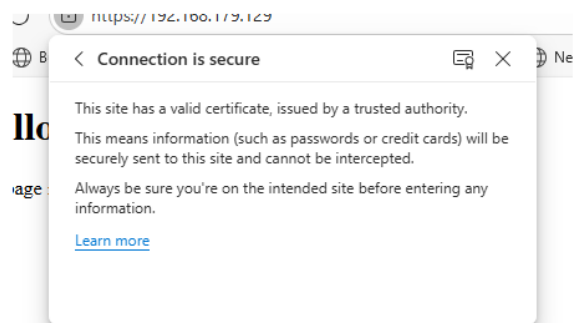
```

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform:~$ curl -vki https://192.168.179.129/app/
* Trying 192.168.179.129:443...
* Connected to 192.168.179.129 (192.168.179.129) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384 / X25519 / RSASSA-PSS
* ALPN: server accepted h2
* Server certificate:
* subject: C=RU; ST=MOSCOW; L=MOSCOW; O=Local; OU=IT; CN=192.168.179.129
* start date: Oct  1 17:19:08 2025 GMT
* expire date: Jan  4 17:19:08 2028 GMT
* issuer: C=RU; ST=MOSCOW; L=MOSCOW; O=Local-CA; OU=IT; CN=Local-Test-CA
* SSL certificate verify result: unable to get local issuer certificate (20), continuing anyway.
* Certificate level 0: Public key type RSA (4096/152 Bits/secBits), signed using sha256WithRSAEncryption
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
* old SSL session ID is stale, removing
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://192.168.179.129/app/
* [HTTP/2] [1] [:method: HEAD]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: 192.168.179.129]
* [HTTP/2] [1] [:path: /app/]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> HEAD /app/ HTTP/2
> Host: 192.168.179.129
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/2 200
HTTP/2 200
< server: nginx/1.24.0 (Ubuntu)
server: nginx/1.24.0 (Ubuntu)
< date: Wed, 01 Oct 2025 18:27:22 GMT
date: Wed, 01 Oct 2025 18:27:22 GMT
< content-type: text/html
content-type: text/html
< content-length: 484
content-length: 484
< last-modified: Tue, 30 Sep 2025 14:20:42 GMT
last-modified: Tue, 30 Sep 2025 14:20:42 GMT
<
* Connection #0 to host 192.168.179.129 left intact

```

Ну и самая главная проверка на windows



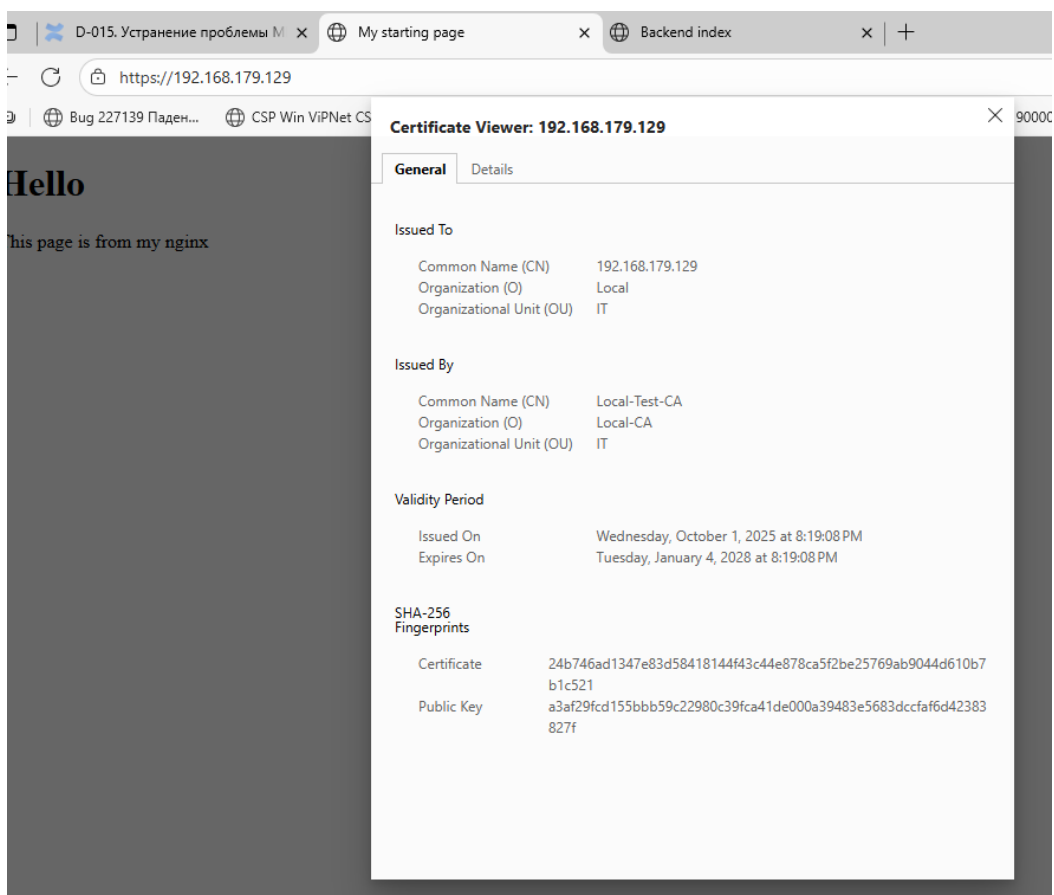


Backend index

Файл: /srv/app/index.html (отдаётся через обратный прокси по URL /app/).

- </app/test.html>
- </app/api/data.json>
- </app/docs/>

[Вернуться на главную Nginx](#)



Установить и настроить ModSecurity, в том числе настроить фильтрацию запросов для обеспечения безопасности от SQL-запросов

Установка ModSecurity

Ставим Modsecurity (попытка поставить из исходников не увенчалась успехом)

```
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:/usr/local/src/ModSecurity$ sudo apt update && sudo apt install -y libmodsecurity3 libnginx-mod-http-modsecurity modsecurity-crs
```

Настройка ModSecurity

Далее нам надо настроить modsecurity.conf. Я нашел файл modsecurity.conf-recommended и скопировал его в modsecurity.conf.

```
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:/usr/local/src/ModSecurity$ sudo cp /usr/local/src/ModSecurity/modsecurity.conf-recommended /etc/nginx/modsec/modsecurity.conf
```

```
GNU nano 7.2
# -- 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

# -- Request body handling -----
# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On

# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#
SecRule REQUEST_HEADERS:Content-Type "^(:application(?:/soap|/)|text/xml)" \
    "id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=XML"

# Enable JSON request body parser.
# Initiate JSON Processor in case of JSON content-type; change accordingly
# if your application does not use 'application/json'
#
SecRule REQUEST_HEADERS:Content-Type "application/json" \
    "id:'200001',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=JSON"

# Sample rule to enable JSON request body parser for more subtypes.
# Uncomment or adapt this rule if you want to engage the JSON
# Processor for "+json" subtypes
#
#SecRule REQUEST_HEADERS:Content-Type "application/[a-z0-9.-]+json" \
#    "id:'200006',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=JSON"

# Maximum request body size we will accept for buffering. If you support
# file uploads then the value given on the first line has to be as large
# as the largest file you are willing to accept. The second value refers
# to the size of data, with files excluded. You want to keep that value as
# low as practical.
#
SecRequestBodyLimit 13107200
SecRequestBodyNoFilesLimit 131072
```

Далее я настроил файл журнала событий для modsecurity. Для этого пришлось узнать под каким пользователем работает nginx и разрешить этому пользователю работать с логом modsecurity

```
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:/usr/local/src/ModSecurity$ grep -E "\suser\s+" /etc/nginx/nginx.conf |& echo "user not set (обычно www-data)"
ls -ld /var/log
ls -l /var/log/modsec_audit.log 2>/dev/null || echo "файла пока нет"
user www-data;
drwxrwxr-x 17 root syslog 4096 Oct 1 00:00 /var/log
файла пока нет
```

Далее создал папку и файл лога

```
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo mkdir -p /var/log/modsecurity
sudo chown www-data:adm /var/log/modsecurity
sudo chmod 750 /var/log/modsecurity
sudo sed -i 's@^SecAuditLog .*@SecAuditLog /var/log/modsecurity/modsec_audit.log@' /etc/nginx/modsec/modsecurity.conf
```

Проверил, что правильно поставили права на папку с логами

```
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ ls -ld /var/log/modsecurity
ls -l /var/log/modsecurity/
drwxr-x--- 2 www-data adm 4096 Oct  2 12:28 /var/log/modsecurity
total 0
```

Далее внес изменения в файл modsecurity.conf и проверил, что все ОК. В том числе убедился, что папка с логами указана верно.

```
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo grep -n -E "SecRuleEngine|SecRequestBodyAccess|SecAuditLog|SecAuditEngine|SecAuditLogParts" /etc/nginx/modsec/modsecurity.conf
7:SecRuleEngine On
16:SecRequestBodyAccess On
50:# when SecRuleEngine is set to DetectionOnly mode in order to minimize
242:SecAuditEngine RelevantOnly
243:SecAuditLogRelevantStatus "^(?:5|4|7|84)"
246:SecAuditLogParts ABIDDEFHZ
251:SecAuditLogType Serial
252:SecAuditLog /var/log/modsecurity/modsec_audit.log
255:SecAuditLogStorageDir /opt/modsecurity/var/audit/
```

Создание файла main.conf

Создал основной файл конфигурации ModSecurity для Nginx - main.conf. Экспериментировал с путями, сначала немножко напутал с путями, но потом разобрался, неправильные пути оставил для себя закомментированными.

```
GNU nano 7.2
Main ModSecurity configuration file for Nginx

# Base ModSecurity config (core)
Include "/etc/nginx/modsec/modsecurity.conf"

# CRS settings
#Include "/etc/nginx/modsec/crs-setup.conf"

# OWASP CRS rules
#Include "/usr/share/modsecurity-crs/rules/*.conf"

# Base ModSecurity config (core)
#Include "/etc/modsecurity/modsecurity.conf"

# CRS settings (package location)
Include "/etc/modsecurity/crs/crs-setup.conf"

# OWASP CRS rules (package location)
Include "/usr/share/modsecurity-crs/rules/*.conf"

# Demo rule for SQLi
#SecRule REQUEST_URI|ARGS "@detectSQLi" \
# "id:1000001,phase:2,deny,log,status:403,msg:'SQLi detected in URI/ARGS'"

```

Далее возникла проблема с Unicode mapping, я ее решил поиском файла и явном указании на него в файле /etc/nginx/sites-available/default

```
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ ls -l /usr/share/modsecurity-crs/unicode.mapping 2>/dev/null || ls -l /usr/local/src/ModSecurity/unicode.mapping 2>/dev/null || echo "unicode.mapping not found"
-rw-r--r-- 1 root root 53146 Oct  2 11:59 /usr/local/src/ModSecurity/unicode.mapping
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo nano /etc/nginx/sites-available/default
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo nano /etc/nginx/modsec/modsecurity.conf
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo nginx -t
2025/10/02 13:01:10 [notice] 17082#17082: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/833/0)
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
vladinir-kuryndin@vladinir-kuryndin-VMware-Virtual-Platform:/usr/local/src/ModSecurity$ sudo systemctl reload nginx
```

Проверил, что все работает

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ sudo nginx -t
2025/10/02 12:41:59 [notice] 17476#17476: ModSecurity-nginx v1.0.3 (rules loaded inline/local/remote: 0/833/0)
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Продемонстрировать успешную блокировку SQL-инъекций

Демонстрация работы Modsecurity (в примере показана XSS, вам необходимо показать на SQLi)

Показать, что при отправке запросов командой curl выводится ошибка 403

XSS

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -I 'http://192.168.179.129/?param="<script><alert(1);</script>' --insecure
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 09:06:15 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -I 'http://192.168.179.129/?param="<script><alert(1);</script>'
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 11:48:37 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
```

SQLi

Показываем на https

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -tk 'https://192.168.179.129/?q=UNION%20SELECT%20username%2C%20password%20FROM%20users'
HTTP/2 403
server: nginx/1.24.0 (Ubuntu)
date: Thu, 02 Oct 2025 14:38:54 GMT
content-type: text/html
content-length: 162

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.24.0 (Ubuntu)</center>
</body>
</html>
```

И на обычном http

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -i 'http://192.168.179.129/?q=UNION%20SELECT%20username%2C%20password%20FROM%20users' --insecure
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:54:36 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.24.0 (Ubuntu)</center>
</body>
</html>
```

```
vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -I -G --data-urlencode "id=1' OR '1'='1' 'http://192.168.179.129/'
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:56:40 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -I -G --data-urlencode "q=1 UNION SELECT 1,0@version-- " "http://192.168.179.129/"
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:56:54 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive

vladimir-kuryndin@vladimir-kuryndin-VMware-Virtual-Platform: /usr/local/src/ModSecurity$ curl -I -G --data-urlencode "id=1 AND updatexml(1,concat(0x7e,(select database()),0x7e),1)" "http://192.168.179.129/"
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:57:11 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
```

Закомментировать правило в файле конфигурации Modsecurity и показать, что Nginx возвращает код 200

```
GNU nano 7.2
-- 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 Off

# -- Request body handling -----

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security

vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:/usr/local/src/ModSecurity$ curl -i 'http://192.168.179.129/?q=UNION%20SELECT%20username%20%20password%20FROM%20users' --insecure
HTTP/1.1 200 OK
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:37:34 GMT
Content-Type: text/html
Content-Length: 169
Last-Modified: Tue, 30 Sep 2025 12:28:59 GMT
Connection: keep-alive
ETag: "68dbcd8b-a9"
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>My starting page</title>
</head>
<body>
  <h1>Hello</h1>
  <p>This page is from my nginx</p>
</body>
</html>
```

Дополнительная проверка различных паттернов

```
vladimir-kuryndin@vladimir-kuryndin-Virtual-Platform:/usr/local/src/ModSecurity$ curl -I "http://192.168.179.129/?page=../../../../etc/passwd"
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Thu, 02 Oct 2025 14:41:18 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
```