

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Организация и обеспечение аудита настроек средств защиты информации»

**ОТЧЕТ ПО МОДУЛЮ №5**

«Защита информации при работе с каналами связи»

**Выполнил:**

Чу Ван Доан, студент группы N3347



\_\_\_\_\_  
(подпись)

**Проверил:**

Пенин Андрей Семенович

\_\_\_\_\_  
(отметка о выполнении)

\_\_\_\_\_  
(подпись)

Санкт-Петербург

2025 г.

## СОДЕРЖАНИЕ

<b>Содержание.....</b>	<b>2</b>
<b>Введение.....</b>	<b>3</b>
<b>Задание.....</b>	<b>4</b>
<b>Ход работы.....</b>	<b>6</b>
1. Этап 1: Установка и настройка межсетевого экрана.....	6
1.1. Подготовка практического стенда.....	6
1.2. Установка iptables и инструментов для тестирования.....	7
1.3. Настройка iptables.....	8
1.4. Проверка настройки iptables.....	9
2. Этап 2. Установка и настройка средств анализа сетевого трафика и зашифрованного канала.....	10
2.1. Создание сертификатов с easy-rsa (выполняется на PC1).....	11
2.2. Настройка OpenVPN сервера (PC1).....	14
2.3. Настройка OpenVPN клиента (PC2).....	16
2.4. Перехват пакетов до шифрования.....	17
2.5. Запуск OpenVPN (шифрование данных).....	19
3. Этап 3. Настройка и демонстрация использования цифровой подписи.....	23
3.1. Установка OpenSSL.....	23
3.2. Создание пары ключей RSA (2048 бит).....	24
3.3. Создание текстового файла и его подписание.....	24
3.4. Проверка подписи с помощью публичного ключа.....	24
3.5. Измените содержимое файла.....	25
<b>Заключение.....</b>	<b>26</b>

## **ВВЕДЕНИЕ**

В современных условиях активного использования информационных технологий особое значение приобретает защита данных, передаваемых по сетям связи. Обеспечение безопасности информации при её передаче включает в себя целый комплекс мер, направленных на предотвращение несанкционированного доступа, обеспечение подлинности источника и конфиденциальности передаваемых данных.

Одним из ключевых аспектов информационной безопасности является организация защищённых каналов связи с использованием технологий шифрования, средств фильтрации трафика и цифровой подписи. Корректная настройка межсетевого экрана позволяет ограничить несанкционированные подключения, анализ трафика — выявить возможные угрозы, а цифровая подпись — гарантировать целостность и подлинность информации.

Цель данной лабораторной работы — практическое освоение средств защиты информации при передаче данных по сети. В ходе выполнения задания предполагается установка и настройка межсетевого экрана, анализ сетевого трафика, организация защищённого канала связи с использованием OpenVPN, а также применение цифровой подписи для проверки подлинности данных.

## ЗАДАНИЕ

### Этап 1. Установка и настройка межсетевого экрана (до 3 баллов)

- Установите программное обеспечение для межсетевого экрана (например, iptables для Linux, Windows Firewall для Windows или pfSense).
- Настройте правила фильтрации трафика: (разрешите доступ к портам, необходимым для шифрованного канала (например, 1194 для OpenVPN); заблокируйте входящие подключения ко всем остальным портам, кроме необходимых для тестирования.)
- Проведите тестирование: попробуйте выполнить подключение к заблокированному порту (например, с помощью telnet или nmap) и зафиксируйте результат блокировки.

### Этап 2. Установка и настройка средств анализа сетевого трафика и шифрованного канала (до 4 баллов)

- Установите средство анализа сетевого трафика (например, Wireshark, tcpdump или аналоги).
- Настройте фильтры для мониторинга трафика на интерфейсе виртуальной машины (например, фильтр по порту OpenVPN).
- Установите и настройте OpenVPN для создания шифрованного канала: настройте сервер и клиент OpenVPN на одной или двух ВМ; используйте сертификаты для аутентификации (например, через easy-rsa).
- Проведите тестирование: перехватите трафик с помощью средства анализа до и после включения шифрования; продемонстрируйте, что данные в канале зашифрованы (например, сравните содержимое пакетов).

### Этап 3. Настройка и демонстрация использования цифровой подписи (до 3 баллов)

- Установите программное обеспечение для создания и проверки цифровой подписи (например, OpenSSL, GPG или встроенные средства Windows).
- Настройте систему: сгенерируйте пару ключей (открытый и закрытый) для цифровой подписи; подпишите тестовый файл (например, текстовый документ) с использованием закрытого ключа.

- Проведите тестирование: проверьте подлинность подписанного файла с использованием открытого ключа; измените файл и убедитесь, что проверка подписи не проходит.

Отчёт должен содержать описание действий, скриншоты и выводы по каждому этапу.

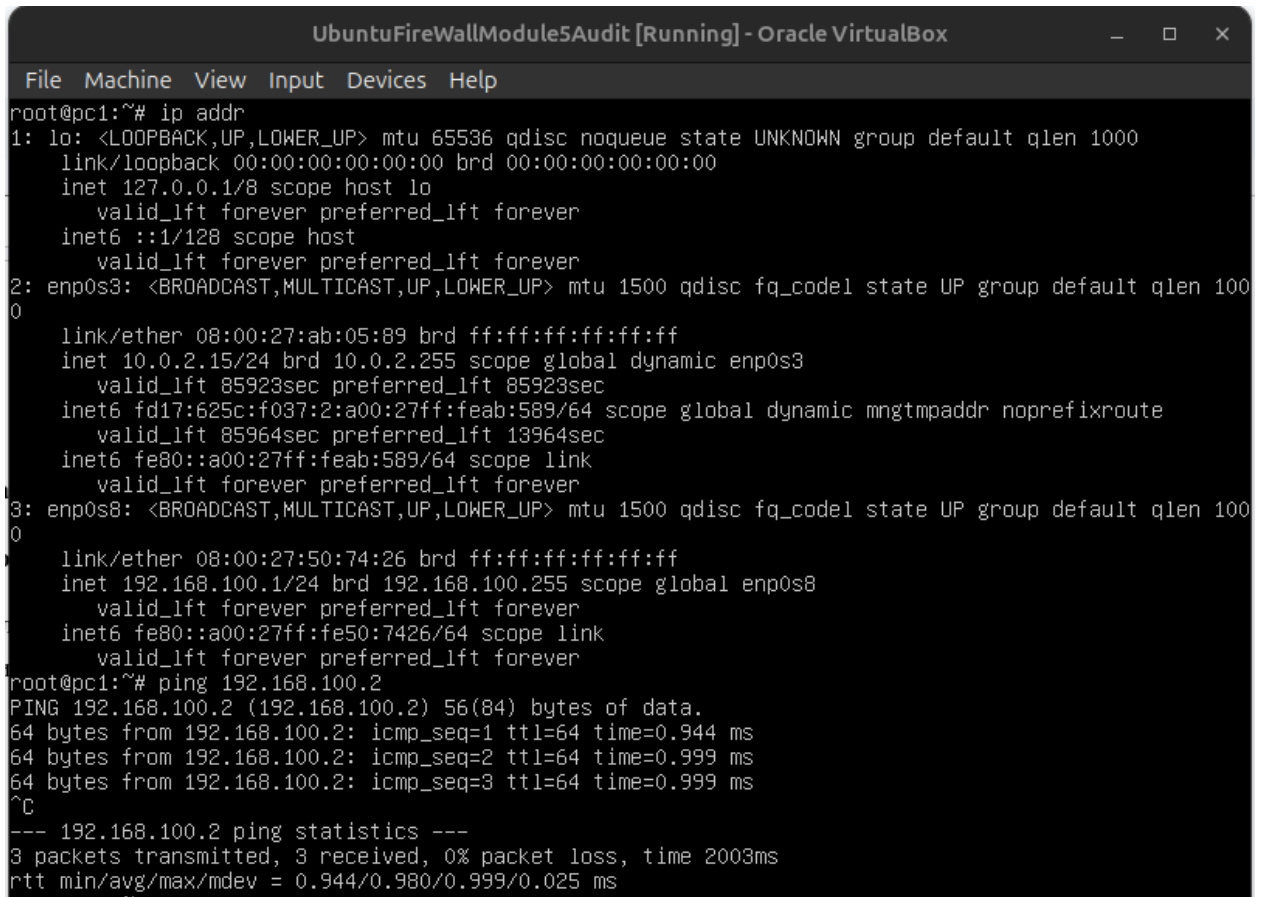
Кроме того быть готовым к демонстрации ВМ на защите работы.

## Ход работы

### 1. Этап 1: Установка и настройка межсетевого экрана

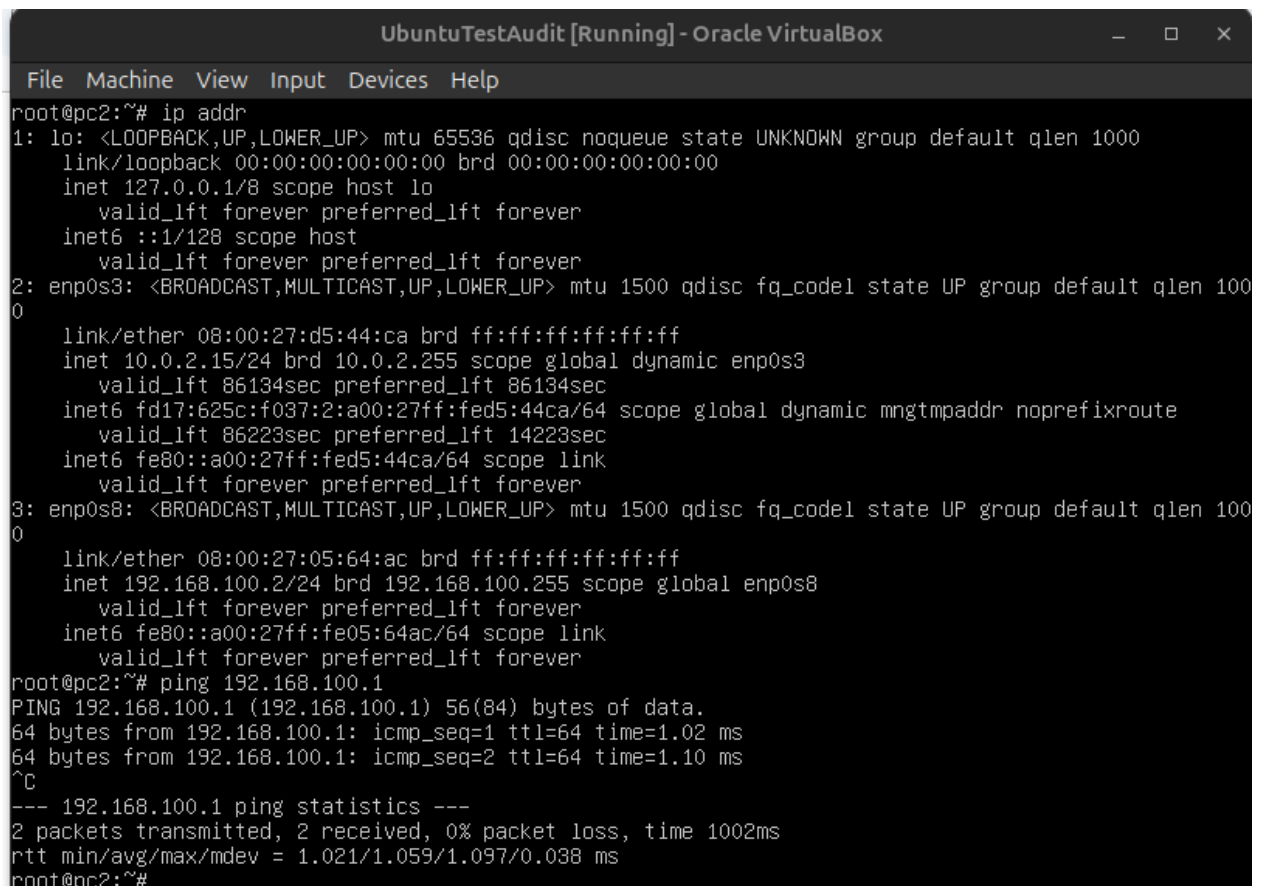
#### 1.1. Подготовка практического стенда

Я подготовил два компьютера с Ubuntu, работающих в VirtualBox. Они подключены друг к другу по сети.



```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@pc1:~# ip addr
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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ab:05:89 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 85923sec preferred_lft 85923sec
    inet6 fd17:625c:f037:2:a00:27ff:feab:589/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 85964sec preferred_lft 13964sec
    inet6 fe80::a00:27ff:feab:589/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:50:74:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/24 brd 192.168.100.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe50:7426/64 scope link
        valid_lft forever preferred_lft forever
root@pc1:~# ping 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=0.944 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=0.999 ms
64 bytes from 192.168.100.2: icmp_seq=3 ttl=64 time=0.999 ms
^C
--- 192.168.100.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.944/0.980/0.999/0.025 ms
```

Рисунок 1 - Настройка сети на PC1



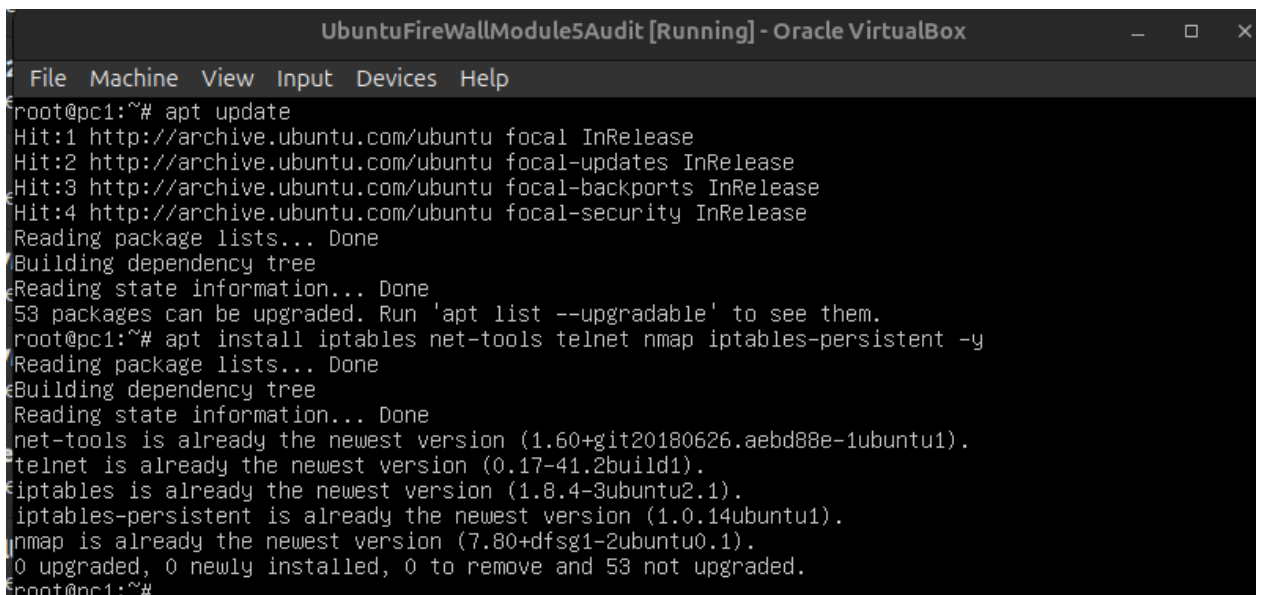
```
root@pc2:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d5:44:ca brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86134sec preferred_lft 86134sec
    inet6 fd17:625c:f037:2:a00:27ff:fed5:44ca/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86223sec preferred_lft 14223sec
    inet6 fe80::a00:27ff:fed5:44ca/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:05:64:ac brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.2/24 brd 192.168.100.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe05:64ac/64 scope link
        valid_lft forever preferred_lft forever
root@pc2:~# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=1.10 ms
^C
--- 192.168.100.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.021/1.059/1.097/0.038 ms
root@pc2:~#
```

Рисунок 2 - Настройка сети на PC2

## 1.2. Установка iptables и инструментов для тестирования

sudo apt update

sudo apt install iptables net-tools telnet nmap iptables-persistent -y



```
root@pc1:~# apt update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
53 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@pc1:~# apt install iptables net-tools telnet nmap iptables-persistent -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
net-tools is already the newest version (1.60+git20180626.aebd88e-1ubuntu1).
telnet is already the newest version (0.17-41.2build1).
iptables is already the newest version (1.8.4-3ubuntu2.1).
iptables-persistent is already the newest version (1.0.14ubuntu1).
nmap is already the newest version (7.80+dfsg1-2ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 53 not upgraded.
root@pc1:~#
```

Рисунок 3 - Установка iptables и инструментов для тестирования

- telnet, nmap: используются для проверки портов

- iptables-persistent: сохраняет конфигурацию iptables после перезагрузки

### 1.3. Настройка iptables

Скрипт для автоматической настройки:

Создайте файл firewall\_setup.sh на PC1: `nano firewall_setup.sh`

```
#!/bin/bash
```

```
# Reset iptables
```

```
iptables -F
```

```
iptables -P INPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
# Allow loopback and established connections
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

```
# Allow OpenVPN port
```

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

```
# (Optional) Allow SSH
```

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
# (Optional) Allow ping
```

```
iptables -A INPUT -p icmp -j ACCEPT
```

```
# Save permanently
```

```
netfilter-persistent save
```

Выполнение:

```
chmod +x firewall_setup.sh
```

```
sudo ./firewall_setup.sh
```



```

UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@pc1:~# iptables -L -n -v --line-numbers
Chain INPUT (policy DROP 2 packets, 152 bytes)
num  pkts bytes target    prot opt in     out     source destination
1          0      0 ACCEPT    all  --  lo     *       0.0.0.0/0      0.0.0.0/0
2          0      0 ACCEPT    udp  --  *      *       0.0.0.0/0      0.0.0.0/0          udp dp
t:1194
3          0      0 ACCEPT    tcp  --  *      *       0.0.0.0/0      0.0.0.0/0          tcp dp
t:22
4          0      0 ACCEPT    icmp --  *      *       0.0.0.0/0      0.0.0.0/0
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination
Chain OUTPUT (policy ACCEPT 2 packets, 152 bytes)
num  pkts bytes target    prot opt in     out     source destination
root@pc1:~# _

```

Рисунок 4 - Настройка iptables

#### 1.4. Проверка настройки iptables

```

UbuntuTestAudit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@pc2:~# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.689 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=1.02 ms
^C
--- 192.168.100.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1017ms
rtt min/avg/max/mdev = 0.689/0.852/1.015/0.163 ms
root@pc2:~# telnet 192.168.100.1 80
Trying 192.168.100.1...
^C
root@pc2:~# nmap -p 1-1000 192.168.100.1
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-25 08:18 UTC
Nmap scan report for 192.168.100.1
Host is up (0.00054s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
MAC Address: 08:00:27:50:74:26 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.68 seconds
root@pc2:~#

```

Рисунок 5 - На PC2

```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@pc1:~# iptables -L -n -v --line-numbers
Chain INPUT (policy DROP 2012 packets, 88848 bytes)
num  pkts bytes target    prot opt in     out     source destination
1          0      0 ACCEPT    all  --  lo      *       0.0.0.0/0  0.0.0.0/0
2          0      0 ACCEPT    udp  --  *       *       0.0.0.0/0  0.0.0.0/0          udp dp
t:1194
3          4    176 ACCEPT    tcp  --  *       *       0.0.0.0/0  0.0.0.0/0          tcp dp
t:22
4         10     840 ACCEPT    icmp --  *       *       0.0.0.0/0  0.0.0.0/0
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source destination
Chain OUTPUT (policy ACCEPT 20 packets, 1456 bytes)
num  pkts bytes target    prot opt in     out     source destination
root@pc1:~#
```

Рисунок 6 - На PC1

Анализ результатов на ПК2 (тестирование):

ping 192.168.100.1

Успешно → потому что вы добавили строку iptables -A INPUT -p icmp -j ACCEPT на ПК1.

telnet 192.168.100.1 80

Не удалось подключиться → ВЕРНО! Потому что порт 80 не открыт в iptables.

nmap -p 1-1000 192.168.100.1

Показывает только порт 22/tcp: closed, остальные порты находятся в состоянии filtered.

Это верно, потому что:

Порт 22 открыт (iptables правило 3)

Остальные порты сбрасываются (DROP)

## 2. Этап 2. Установка и настройка средств анализа сетевого трафика и шифрованного канала

Установка tcpdump для захвата пакетов на обоих ПК — ПК1 и ПК2

```
root@pc2:~# apt install tcpdump -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  tcpdump
1 upgraded, 0 newly installed, 0 to remove and 56 not upgraded.
Need to get 370 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 tcpdump amd64 4.9.3-4ubuntu0.3 [370 kB]
Fetched 370 kB in 1s (313 kB/s)
(Reading database ... 70329 files and directories currently installed.)
Preparing to unpack .../tcpdump_4.9.3-4ubuntu0.3_amd64.deb ...
Unpacking tcpdump (4.9.3-4ubuntu0.3) over (4.9.3-4ubuntu0.2) ...
Setting up tcpdump (4.9.3-4ubuntu0.3) ...
Installing new version of config file /etc/apparmor.d/usr.sbin.tcpdump ...
Processing triggers for man-db (2.9.1-1) ...
root@pc2:~#

root@pc1:~# apt install tcpdump -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  tcpdump
1 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.
Need to get 370 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 tcpdump amd64 4.9.3-4ubuntu0.3 [370 kB]
Fetched 370 kB in 2s (206 kB/s)
(Reading database ... 70397 files and directories currently installed.)
Preparing to unpack .../tcpdump_4.9.3-4ubuntu0.3_amd64.deb ...
Unpacking tcpdump (4.9.3-4ubuntu0.3) over (4.9.3-4ubuntu0.2) ...
Setting up tcpdump (4.9.3-4ubuntu0.3) ...
Installing new version of config file /etc/apparmor.d/usr.sbin.tcpdump ...
Processing triggers for man-db (2.9.1-1) ...
root@pc1:~#
```

Рисунок 7 - Установка tcpdump

Установка OpenVPN и easy-rsa

```

root@pc1:~# apt install openvpn easy-rsa -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
easy-rsa is already the newest version (3.0.6-1).
openvpn is already the newest version (2.4.12-0ubuntu0.20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.

```

```

root@pc2:~# apt install openvpn easy-rsa -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
easy-rsa is already the newest version (3.0.6-1).
openvpn is already the newest version (2.4.12-0ubuntu0.20.04.2).
0 upgraded, 0 newly installed, 0 to remove and 56 not upgraded.

```

Рисунок 8 - Установка OpenVPN и easy-rsa

## 2.1. Создание сертификатов с easy-rsa (выполняется на PC1)

- Создание директории CA- Certificate Authority (Центра Сертификации)

**make-cadir ~/openvpn-ca**

**cd ~/openvpn-ca**

- Настройка параметров CA.
- Отредактируйте конфигурационный файл: **nano vars**

**set\_var EASYRSA\_REQ\_COUNTRY "RU"**

**set\_var EASYRSA\_REQ\_PROVINCE "Saint Petersburg"**

**set\_var EASYRSA\_REQ\_CITY "Saint Petersburg"**

**set\_var EASYRSA\_REQ\_ORG "ITMO"**

**set\_var EASYRSA\_REQ\_EMAIL "sun@itmo.ru"**

**set\_var EASYRSA\_REQ\_OU "SECURITY"**

```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
GNU nano 4.8 vars
# Choices are:
#   cn_only - use just a CN value
#   org     - use the "traditional" Country/Province/City/Org/OU/email/CN format
#set_var EASYRSA_DN      "cn_only"

# Organizational fields (used with 'org' mode and ignored in 'cn_only' mode.)
# These are the default values for fields which will be placed in the
# certificate. Don't leave any of these fields blank, although interactively
# you may omit any specific field by typing the "." symbol (not valid for
# email.)

#set_var EASYRSA_REQ_COUNTRY    "RU"
#set_var EASYRSA_REQ_PROVINCE   "Saint Petersburg"
#set_var EASYRSA_REQ_CITY       "Saint Petersburg"
#set_var EASYRSA_REQ_ORG        "ITMO"
#set_var EASYRSA_REQ_EMAIL       "sun@itmo.ru"
#set_var EASYRSA_REQ_OU         "SECURITY"

# Choose a size in bits for your keypairs. The recommended value is 2048. Using
# 2048-bit keys is considered more than sufficient for many years into the
# future. Larger key sizes will slow down TLS negotiation and make key/DH param
# generation take much longer. Values up to 4096 should be accepted by most
# software. Only used when the crypto alg is rsa (see below.)
#set_var EASYRSA_KEY_SIZE      2048

# The default crypto mode is rsa; ec can enable elliptic curve support.
# Note that not all software supports ECC, so use care when enabling it.
# Choices for crypto alg are: (each in lower-case)
#   * rsa
#   * ec

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Unde
^X Exit      ^R Read File  ^_ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Red
```

Рисунок 9 - Настройка параметров СА

- Инициализация СА и создание сертификата:

**./easyrsa init-pki**

**./easyrsa build-ca nopass** # Создание СА без пароля

```

root@pc1:~/openvpn-ca# ./easysrsa init-pki
Note: using Easy-RSA configuration from: ./vars
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /root/openvpn-ca/pki

root@pc1:~/openvpn-ca# ./easysrsa build-ca nopass
Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Can't load /root/openvpn-ca/pki/.rnd into RNG
140413389038912:error:2406F079:random number generator:RAND_load_file:Cannot open file:../cr
d/randfile.c:98:Filename=/root/openvpn-ca/pki/.rnd
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:sun

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/openvpn-ca/pki/ca.crt

root@pc1:~/openvpn-ca# _

```

Рисунок 10 - Инициализация CA и создание сертификата

- Создание ключа для сервера:

**`./easysrsa gen-req server nopass`**

**`./easysrsa sign-req server server`**

```

root@pc1:~/openvpn-ca# ./easysrsa gen-req server nopass
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020

WARNING!!!
An existing private key was found at /root/openvpn-ca/pki/private/server.key
Continuing with key generation will replace this key.

Type the word 'yes' to continue, or any other input to abort.
Confirm key overwrite: yes
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/root/openvpn-ca/pki/private/server.key.d3x6R13X0'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [server]:sun

Keypair and certificate request completed. Your files are:
req: /root/openvpn-ca/pki/req/server.req
key: /root/openvpn-ca/pki/private/server.key

```

```

root@pc1:~/openvpn-ca# ./easysrsa sign-req server server
Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.1f 31 Mar 2020

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 1080 days:
subject=
  commonName = sun

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /root/openvpn-ca/pki/safessl-easysrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'sun'
Certificate is to be certified until Apr  9 11:46:19 2028 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /root/openvpn-ca/pki/issued/server.crt

```

Рисунок 11 - Создание ключа для сервера.

- Создание ключа для клиента:

**`./easysrsa gen-req client1 nopass`**

**`./easysrsa sign-req client client1`**

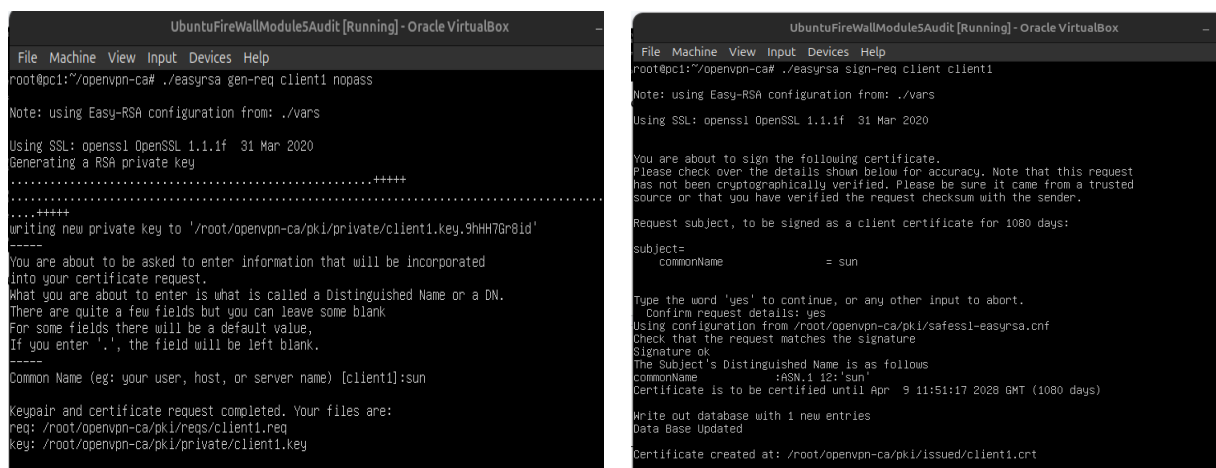


Рисунок 12 - Создание ключа для клиента.

- Генерация Diffie-Hellman и tls-auth

**./easysrsa gen-dh**

**openvpn --genkey --secret ta.key**



## 2.2. Настройка OpenVPN сервера (PC1)

- Создание каталога конфигурации:

**sudo cp -r /usr/share/doc/openvpn/examples/sample-config-files /etc/openvpn**

**cd /etc/openvpn/sample-config-files**

**sudo cp server.conf.gz /etc/openvpn/**

**cd /etc/openvpn**

**sudo gzip -d server.conf.gz**

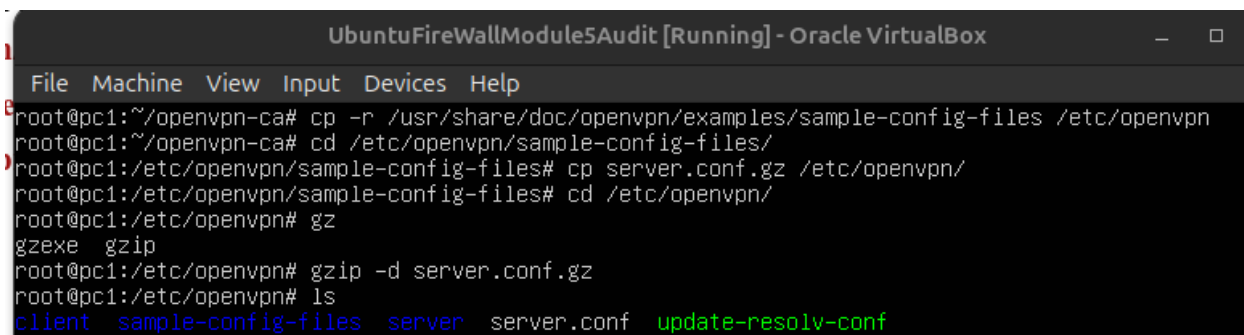


Рисунок 13 - Создание каталога конфигурации.

- Открытие и редактирование файла server.conf

**port 1194**

**proto udp**

**dev tun**

**ca /etc/openvpn/keys/ca.crt**

**cert /etc/openvpn/keys/server.crt**

**key /etc/openvpn/keys/server.key**

**dh /etc/openvpn/keys/dh.pem**

**tls-auth /etc/openvpn/keys/ta.key 0**

**server 10.8.0.0 255.255.255.0**

**ifconfig-pool-persist /var/log/openvpn/ipp.txt**

**keepalive 10 120**

**cipher AES-256-CBC**

**auth SHA256**

**user nobody**

**group nogroup**

**persist-key**

**persist-tun**

**status /var/log/openvpn/openvpn-status.log**

**log /var/log/openvpn/openvpn.log**

**verb 3**

- Копирование сертификатов в правильное место:

**cp ~/openvpn-ca/ta.key /etc/openvpn/keys/**

**sudo mkdir /etc/openvpn/keys**

**sudo cp ~/openvpn-ca/pki/{ca.crt,dh.pem,private/server.key,issued/server.crt}  
/etc/openvpn/keys/**

```

root@pc1:/etc/openvpn# cd ~/openvpn-ca/
root@pc1:~/openvpn-ca# s
s: command not found
root@pc1:~/openvpn-ca# ls
easyrsa openssl-easyrsa.cnf pki ta.key vars x509-types
root@pc1:~/openvpn-ca# cp ta.key /etc/openvpn/keys/
root@pc1:~/openvpn-ca# cp ~/openvpn-ca/pki/{ca.crt,dh.pem,ta.key,private/server.key,issued/server.crt} /etc/openvpn/keys/
cp: cannot stat '/root/openvpn-ca/pki/ta.key': No such file or directory
root@pc1:~/openvpn-ca# cp ~/openvpn-ca/pki/{ca.crt,dh.pem,private/server.key,issued/server.crt} /etc/openvpn/keys/
root@pc1:~/openvpn-ca# _

```

Рисунок 14 -Копирование сертификатов в правильное место.

### 2.3. Настройка OpenVPN клиента (PC2)

На PC2 создайте файл client.ovpn

Содержимое файла:

**client**

**dev tun**

**proto udp**

**remote 192.168.100.1 1194**

**resolv-retry infinite**

**nobind**

**persist-key**

**persist-tun**

**ca ca.crt**

**cert Client1.crt**

**key Client1.key**

**tls-auth ta.key 1**

**cipher AES-256-CBC**

**verb 3**

- Скопируйте следующие файлы с PC1 на PC2 с помощью SCP: ca.crt, client1.crt, client1.key, ta.key

**scp ~/openvpn-ca/pki/ca.crt pc2@192.168.100.2:~**

**scp ~/openvpn-ca/pki/issued/client1.crt pc2@192.168.100.2:~**

**scp ~/openvpn-ca/pki/private/client1.key pc2@192.168.100.2:~**

**scp ~/openvpn-ca/ta.key pc2@192.168.100.2:~**



```

root@pc1:~/openvpn-ca# scp /root/openvpn-ca/pki/ca.crt pc2@192.168.100.2:~
The authenticity of host '192.168.100.2 (192.168.100.2)' can't be established.
ECDSA key fingerprint is SHA256:pZhAcgLYsYa6aDEm07Kkkftbdo5Z2f20ba4S2yTwigU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.2' (ECDSA) to the list of known hosts.
pc2@192.168.100.2's password:
ca.crt                                                                 100% 1172      2.1MB/s   00:00
root@pc1:~/openvpn-ca# scp /root/openvpn-ca/pki/issued/client1.crt pc2@192.168.100.2:~
pc2@192.168.100.2's password:
client1.crt                                                           100% 4445      415.8KB/s 00:00
root@pc1:~/openvpn-ca# scp /root/openvpn-ca/ta.key pc2@192.168.100.2:~
pc2@192.168.100.2's password:
ta.key                                                                 100% 636       75.8KB/s  00:00
root@pc1:~/openvpn-ca# _

```

Рисунок 15 -Копирование файлы с PC1 на PC2 с помощью SCP

#### 2.4. Перехват пакетов до шифрования.

На обоих ПК1 и ПК2 убедитесь, что OpenVPN не запущен

На ПК1: `sudo systemctl stop openvpn@server`

На ПК2: Не запускайте `openvpn --config`.

Перехват пакетов на ПК1, например, по порту SSH (22):

`sudo tcpdump -i enp0s8 port 22 -n -vv`

Теперь вы можете выполнить на ПК2:

`ssh pc1@192.168.100.1`

```
UbuntuTestAudit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
* Support: https://ubuntu.com/advantage

System information as of Fri 25 Apr 2025 02:32:42 PM UTC

System load: 0.0
Usage of /: 40.9% of 11.21GB
Memory usage: 25%
Swap usage: 0%
Processes: 114
Users logged in: 1
IPv4 address for enp0s3: 10.0.2.15
IPv6 address for enp0s3: fd17:625c:f037:2:a00:27ff:feab:589
IPv4 address for enp0s8: 192.168.100.1

* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr 25 11:07:54 2025
pc1@pc1:~$
```

Рисунок 16 - Login на ПК2

В выводе tcpdump мы увидим открытый полезный трафик, этапы рукопожатия SSH (в виде простого текста или в легко анализируемом формате)

```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
14:32:43.105148 IP (tos 0x10, ttl 64, id 18905, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x499f (incorrect -> 0xc586), seq 3223
:3259, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 36
14:32:43.105192 IP (tos 0x10, ttl 64, id 43335, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.100.2.43286 > 192.168.100.1.22: Flags [.], cksum 0x424e (correct), seq 2387, ack 3187, w
in 501, options [nop,nop,TS val 3487995547 ecr 1390115127], length 0
14:32:43.105192 IP (tos 0x10, ttl 64, id 43336, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.100.2.43286 > 192.168.100.1.22: Flags [P.], cksum 0x4229 (correct), seq 2387, ack 3223, w
in 501, options [nop,nop,TS val 3487995547 ecr 1390115128], length 0
14:32:43.105215 IP (tos 0x10, ttl 64, id 18906, offset 0, flags [DF], proto TCP (6), length 120)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x49bf (incorrect -> 0xa733), seq 3259
:3327, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 68
14:32:43.105275 IP (tos 0x10, ttl 64, id 43337, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.100.2.43286 > 192.168.100.1.22: Flags [P.], cksum 0x4205 (correct), seq 2387, ack 3259, w
in 501, options [nop,nop,TS val 3487995547 ecr 1390115128], length 0
14:32:43.105304 IP (tos 0x10, ttl 64, id 18907, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x499f (incorrect -> 0x9250), seq 3327
:3363, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 36
14:32:43.105360 IP (tos 0x10, ttl 64, id 18908, offset 0, flags [DF], proto TCP (6), length 136)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x49cf (incorrect -> 0x461a), seq 3363
:3447, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 84
14:32:43.105414 IP (tos 0x10, ttl 64, id 18909, offset 0, flags [DF], proto TCP (6), length 88)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x499f (incorrect -> 0x8963), seq 3447
:3483, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 36
14:32:43.105470 IP (tos 0x10, ttl 64, id 18910, offset 0, flags [DF], proto TCP (6), length 120)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x49bf (incorrect -> 0xf4e6), seq 3483
:3551, ack 2387, win 501, options [nop,nop,TS val 1390115128 ecr 3487995547], length 68
14:32:43.114532 IP (tos 0x10, ttl 64, id 43339, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.100.2.43286 > 192.168.100.1.22: Flags [P.], cksum 0x364d (correct), seq 2387, ack 6247, w
in 501, options [nop,nop,TS val 3487995556 ecr 1390115131], length 0
14:32:43.262723 IP (tos 0x10, ttl 64, id 18918, offset 0, flags [DF], proto TCP (6), length 104)
    192.168.100.1.22 > 192.168.100.2.43286: Flags [P.], cksum 0x49af (incorrect -> 0x9275), seq 6247
:6299, ack 2387, win 501, options [nop,nop,TS val 1390115285 ecr 3487995556], length 52
14:32:43.263296 IP (tos 0x10, ttl 64, id 43340, offset 0, flags [DF], proto TCP (6), length 52)
    192.168.100.2.43286 > 192.168.100.1.22: Flags [P.], cksum 0x34ea (correct), seq 2387, ack 6299, w
in 501, options [nop,nop,TS val 3487995705 ecr 1390115285], length 0
```

Рисунок 17 - PC1

## 2.5. Запуск OpenVPN (шифрование данных).

На ПК1: `sudo systemctl start openvpn@server`

Проверка: `sudo systemctl status openvpn@server`

```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@pc1:~/openvpn-ca# systemctl start openvpn@server
root@pc1:~/openvpn-ca# systemctl status openvpn@server
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled-runtime; vendor preset: enabled)
   Active: active (running) since Fri 2025-04-25 14:43:56 UTC; 35s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 7288 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 1046)
    Memory: 1.0M
    CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
            └─7288 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 -
Apr 25 14:43:56 pc1 ovpn-server[7288]: Could not determine IPv4/IPv6 protocol. Using AF_INET
Apr 25 14:43:56 pc1 ovpn-server[7288]: Socket Buffers: R=[212992->212992] S=[212992->212992]
Apr 25 14:43:56 pc1 ovpn-server[7288]: UDPv4 link local (bound): [AF_INET] [undef]:1194
Apr 25 14:43:56 pc1 ovpn-server[7288]: UDPv4 link remote: [AF_UNSPEC]
Apr 25 14:43:56 pc1 ovpn-server[7288]: GID set to nogroup
Apr 25 14:43:56 pc1 ovpn-server[7288]: UID set to nobody
Apr 25 14:43:56 pc1 ovpn-server[7288]: MULTI: multi_init called, r=256 v=256
Apr 25 14:43:56 pc1 ovpn-server[7288]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
Apr 25 14:43:56 pc1 ovpn-server[7288]: IFCONFIG POOL LIST
Apr 25 14:43:56 pc1 ovpn-server[7288]: Initialization Sequence Completed
lines 1-23/23 (END)
```

Рисунок 18 - Status PC1

```
UbuntuFireWallModule5Audit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Apr 25 15:05:03 pc1 ovpn-server[7378]: succeeded -> ifconfig_pool_set()
Apr 25 15:05:03 pc1 ovpn-server[7378]: IFCONFIG POOL LIST
Apr 25 15:05:03 pc1 ovpn-server[7378]: sun,10.8.0.4
Apr 25 15:05:03 pc1 ovpn-server[7378]: Initialization Sequence Completed
lines 1-23/23 (END)
^C
root@pc1:/etc/ovpn/keys# ip addr
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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ab:05:89 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 72112sec preferred_lft 72112sec
    inet6 fd17:625c:f037:2:a00:27ff:feab:589/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86052sec preferred_lft 14052sec
    inet6 fe80::a00:27ff:feab:589/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:50:74:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.1/24 brd 192.168.100.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe50:7426/64 scope link
        valid_lft forever preferred_lft forever
7: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::4a01:ea6c:2508:c489/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
root@pc1:/etc/ovpn/keys# _
```

Рисунок 19 - Ip PC1

На ПК2: `sudo openvpn --config client.ovpn`

```
UbuntuTestAudit [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Fri Apr 25 14:56:48 2025 Outgoing Control Channel Authentication: Using 160 bit message hash 'SHA1'
for HMAC authentication
Fri Apr 25 14:56:48 2025 Incoming Control Channel Authentication: Using 160 bit message hash 'SHA1'
for HMAC authentication
Fri Apr 25 14:56:48 2025 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.100.1:11
94
Fri Apr 25 14:56:48 2025 Socket Buffers: R=[212992->212992] S=[212992->212992]
Fri Apr 25 14:56:48 2025 UDP link local: (not bound)
Fri Apr 25 14:56:48 2025 UDP link remote: [AF_INET]192.168.100.1:1194
Fri Apr 25 14:56:48 2025 TLS: Initial packet from [AF_INET]192.168.100.1:1194, sid=691598bb 6e4fa884
Fri Apr 25 14:56:48 2025 VERIFY OK: depth=1, CN=sun
Fri Apr 25 14:56:48 2025 VERIFY OK: depth=0, CN=sun
Fri Apr 25 14:56:48 2025 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit R
SA
Fri Apr 25 14:56:48 2025 [sun] Peer Connection Initiated with [AF_INET]192.168.100.1:1194
Fri Apr 25 14:56:49 2025 SENT CONTROL [sun]: 'PUSH_REQUEST' (status=1)
Fri Apr 25 14:56:49 2025 PUSH: Received control message: 'PUSH_REPLY,route 10.8.0.1,topology net30,p
ing 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5,peer-id 1,cipher AES-256-GCM'
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: timers and/or timeouts modified
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: --ifconfig/up options modified
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: route options modified
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: peer-id set
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: adjusting link_mtu to 1624
Fri Apr 25 14:56:49 2025 OPTIONS IMPORT: data channel crypto options modified
Fri Apr 25 14:56:49 2025 Data Channel: using negotiated cipher 'AES-256-GCM'
Fri Apr 25 14:56:49 2025 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Fri Apr 25 14:56:49 2025 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
Fri Apr 25 14:56:49 2025 ROUTE_GATEWAY 10.0.2.2/255.255.255.0 IFACE=enp0s3 HWADDR=08:00:27:d5:44:ca
Fri Apr 25 14:56:49 2025 TUN/TAP device tun0 opened
Fri Apr 25 14:56:49 2025 TUN/TAP TX queue length set to 100
Fri Apr 25 14:56:49 2025 /sbin/ip link set dev tun0 up mtu 1500
Fri Apr 25 14:56:49 2025 /sbin/ip addr add dev tun0 local 10.8.0.6 peer 10.8.0.5
Fri Apr 25 14:56:49 2025 /sbin/ip route add 10.8.0.1/32 via 10.8.0.5
Fri Apr 25 14:56:49 2025 WARNING: this configuration may cache passwords in memory -- use the auth-n
ocache option to prevent this
Fri Apr 25 14:56:49 2025 Initialization Sequence Completed
```

Рисунок 20 - Соединение прошло успешно

На ПК1, захватите пакеты на физическом сетевом интерфейсе (enp0s8) и отфильтруйте по порту 1194 (UDP)

```
sudo tcpdump -i enp0s8 port 1194 -n -vv
```

Затем с ПК2 выполните ping IP-адреса VPN-сервера: ping 10.8.0.1

```
root@pc2:/home/pc2# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=2.05 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=2.91 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.612 ms
^C
--- 10.8.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.612/1.858/2.912/0.948 ms
root@pc2:/home/pc2# kill openvpn
```

Рисунок 21 - PC2 ping ip address VPN-server

```

root@pc1:/etc/openvpn/keys# tcpdump -i enp0s8 port 1194 -n -vv
tcpdump: listening on enp0s8, link-type EN10MB (Ethernet), capture size 262144 bytes
15:07:42.559153 IP (tos 0x0, ttl 64, id 49590, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 108
15:07:42.560780 IP (tos 0x0, ttl 64, id 25828, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x49da -> 0x058b!] UDP, length 108
15:07:43.562374 IP (tos 0x0, ttl 64, id 49729, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 108
15:07:43.562596 IP (tos 0x0, ttl 64, id 25993, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x49da -> 0xbc0f!] UDP, length 108
15:07:44.563169 IP (tos 0x0, ttl 64, id 49744, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 108
15:07:44.563360 IP (tos 0x0, ttl 64, id 26157, offset 0, flags [DF], proto UDP (17), length 136)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x49da -> 0xb723!] UDP, length 108
15:07:54.579460 IP (tos 0x0, ttl 64, id 50301, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 40
15:07:54.579721 IP (tos 0x0, ttl 64, id 27623, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x4996 -> 0xab24!] UDP, length 40
15:08:04.615044 IP (tos 0x0, ttl 64, id 27876, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x4996 -> 0x7fa4!] UDP, length 40
15:08:04.616262 IP (tos 0x0, ttl 64, id 52220, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 40
15:08:14.783348 IP (tos 0x0, ttl 64, id 28067, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.1.1194 > 192.168.100.2.51092: [bad udp cksum 0x4996 -> 0x093f!] UDP, length 40
15:08:14.783729 IP (tos 0x0, ttl 64, id 53660, offset 0, flags [DF], proto UDP (17), length 68)
  192.168.100.2.51092 > 192.168.100.1.1194: [udp sum ok] UDP, length 40

```

Рисунок 22 - tcpdump PC1

В выводе tcpdump мы не увидим содержимого пакетов, как раньше — это будут UDP-пакеты с зашифрованными данными.

#### Сравнение результатов

Состояние	Отображение tcpdump
До VPN	Пакеты читаемые: TCP handshake, SSH...
После VPN	UDP-пакеты с нечитаемым содержимым — данные зашифрованы

### 3. Этап 3. Настройка и демонстрация использования цифровой подписи

#### 3.1. Установка OpenSSL

`sudo apt update`

`sudo apt install openssl -y`

```

root@pc1:~# apt install openssl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1f-1ubuntu2.24).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.
root@pc1:~#

```

Рисунок 23 - Установка OpenSSL

### 3.2. Создание пары ключей RSA (2048 бит)

# Создание приватного ключа

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048
```

# Извлечение публичного ключа из приватного

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

У нас будет 2 файла:

private\_key.pem — приватный ключ (только для подписанта)

public\_key.pem — публичный ключ (передаётся для проверки)

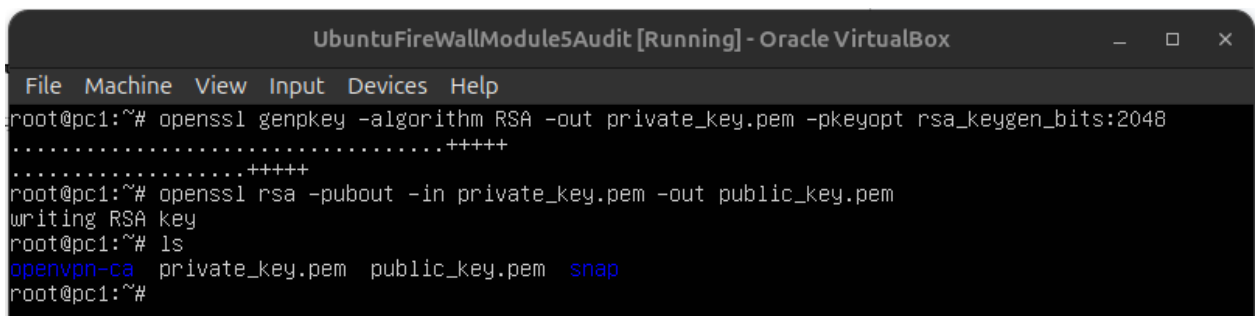


Рисунок 24 - Создание пары ключей RSA (2048 бит)

### 3.3. Создание текстового файла и его подписание

Создание текстового файла:

```
echo "This is the document that needs to be signed" > message.txt
```

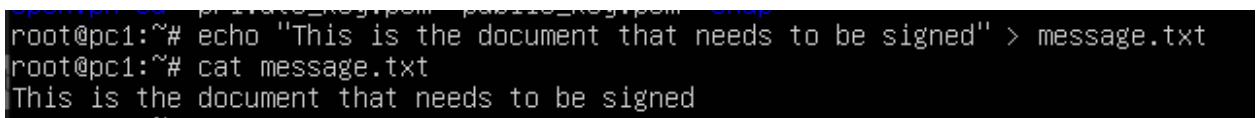


Рисунок 25 - Создание текстового файла

Создание цифровой подписи с помощью приватного ключа:

```
openssl dgst -sha256 -sign private_key.pem -out message.sig message.txt
```

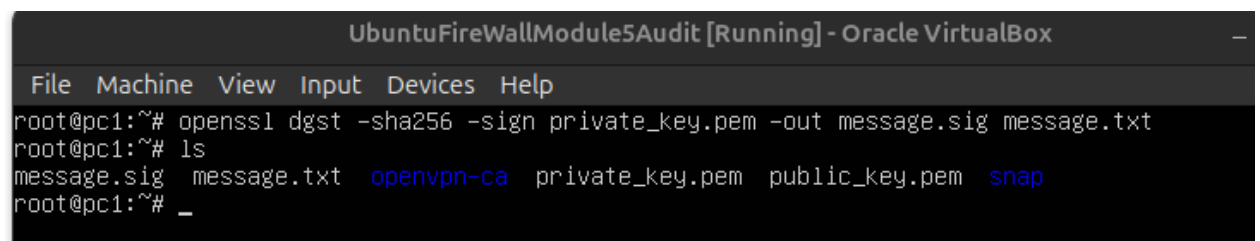


Рисунок 26 - Создание цифровой подписи с помощью приватного ключа

Файл message.sig — это цифровая подпись.

### 3.4. Проверка подписи с помощью публичного ключа

```
openssl dgst -sha256 -verify public_key.pem -signature message.sig message.txt
```



```
root@pc1:~# openssl dgst -sha256 -verify public_key.pem -signature message.sig message.txt
Verified OK
```

Рисунок 26 - подпись действительна

### 3.5. Измените содержимое файла

echo "I have modified this file" >> message.txt

openssl dgst -sha256 -verify public\_key.pem -signature message.sig message.txt

```
root@pc1:~# echo "I have modified this file" >> message.txt
root@pc1:~# cat message.txt
This is the document that needs to be signed
I have modified this file
root@pc1:~# openssl dgst -sha256 -verify public_key.pem -signature message.sig message.txt
Verification Failure
root@pc1:~#
```

Рисунок 27 - Результат

Итог тест-кейса:

Действие	Ожидаемый результат
Подписать исходный файл приватным ключом	Создан файл message.sig
Проверить исходный файл с публичным ключом	Verified OK
Изменить содержимое файла и проверить снова	Verification Failure

## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была реализована комплексная защита информации при передаче данных по сетевым каналам. На первом этапе был настроен межсетевой экран с использованием iptables, обеспечивающий фильтрацию трафика и блокировку несанкционированных подключений. На втором этапе был установлен и сконфигурирован OpenVPN для создания зашифрованного канала связи, проведен анализ трафика с помощью tcpdump, подтверждающий, что данные в туннеле действительно зашифрованы. На третьем этапе была изучена технология цифровой подписи: сгенерированы открытый и закрытый ключи, подписан файл и выполнена проверка подлинности с использованием открытого ключа, а также подтверждено, что любые изменения в файле приводят к недействительности подписи.

Таким образом, были получены практические навыки по защите информации с применением межсетевых экранов, VPN-технологий и криптографических методов, что имеет важное значение для обеспечения безопасности в современных сетях.

## **ПРИЛОЖЕНИЕ А**

### **Листинг А.1 – Код файла main.py**