**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)

# Содержание

[**Содержание 2**](#_heading=h.3jbepr30actf)

[**Введение 3**](#_heading=h.gjdgxs)

[**Задание 4**](#_heading=h.1fob9te)

[**Ход работы 6**](#_heading=h.3znysh7)

[1. Этап 1: Установка и настройка межсетевого экрана 6](#_heading=h.k303886ij2nw)

[1.1. Подготовка практического стенда 6](#_heading=h.43qer6akrcwb)

[1.2. Установка iptables и инструментов для тестирования 7](#_heading=h.gu9s9v538wg6)

[1.3. Настройка iptables 8](#_heading=h.o1s22ts6hser)

[1.4. Проверка настройки iptables 9](#_heading=h.mtg2am50zp26)

[2. Этап 2. Установка и настройка средств анализа сетевого трафика и шифрованного канала 10](#_heading=h.6c2qhrp7ch0g)

[2.1. Создание сертификатов с easy-rsa (выполняется на PC1) 11](#_heading=h.92nl0yqgf1jk)

[2.2. Настройка OpenVPN сервера (PC1) 14](#_heading=h.wgxwsv6y1niu)

[2.3. Настройка OpenVPN клиента (PC2) 16](#_heading=h.a7vgnjaqeftn)

[2.4. Перехват пакетов до шифрования. 17](#_heading=h.aiopqfrn76zh)

[2.5. Запуск OpenVPN (шифрование данных). 19](#_heading=h.xfjldosa5vz9)

[3. Этап 3. Настройка и демонстрация использования цифровой подписи 23](#_heading=h.e78y1evq0kg9)

[3.1. Установка OpenSSL 23](#_heading=h.bbkzs87r2a2b)

[3.2. Создание пары ключей RSA (2048 бит) 24](#_heading=h.au6m7y6up9q7)

[3.3. Создание текстового файла и его подписание 24](#_heading=h.77dvwa4paxgz)

[3.4. Проверка подписи с помощью публичного ключа 24](#_heading=h.g6qkgg1vd7ic)

[3.5. Измените содержимое файла 25](#_heading=h.cez3fhwg58uu)

[**Заключение 26**](#_heading=h.jhow0r2henwe)

# 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

## Этап 1: Установка и настройка межсетевого экрана

### Подготовка практического стенда

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

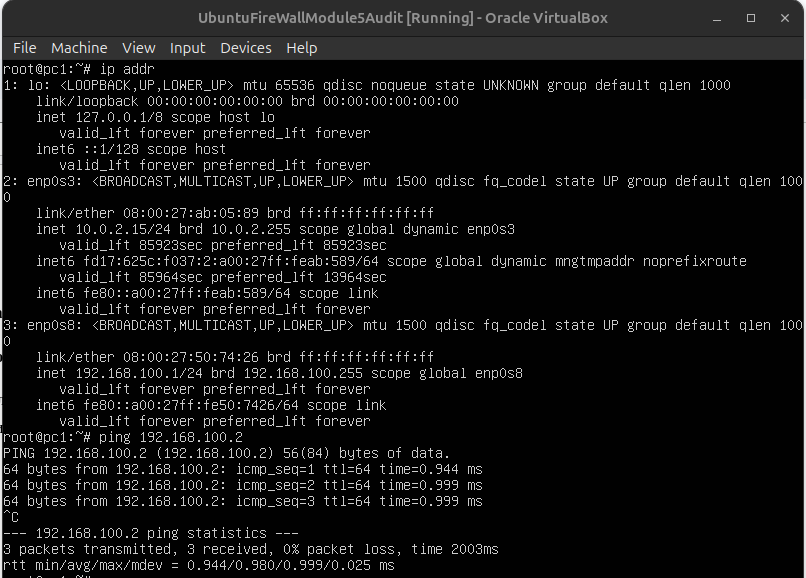


Рисунок 1 - Настройка сети на PC1

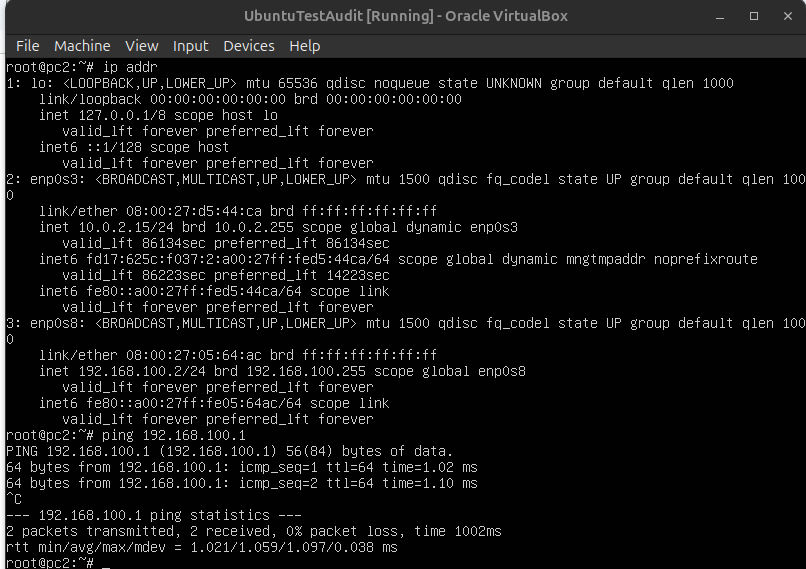


Рисунок 2 - Настройка сети на PC2

### Установка iptables и инструментов для тестирования

sudo apt update

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

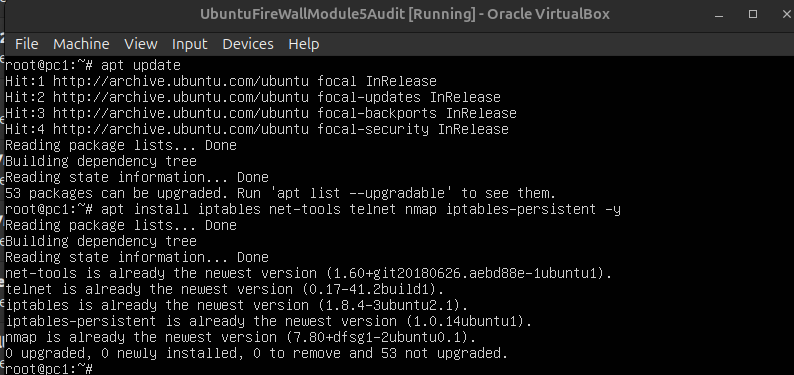


Рисунок 3 - Установка iptables и инструментов для тестирования

* telnet, nmap: используются для проверки портов
* iptables-persistent: сохраняет конфигурацию iptables после перезагрузки

### 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**

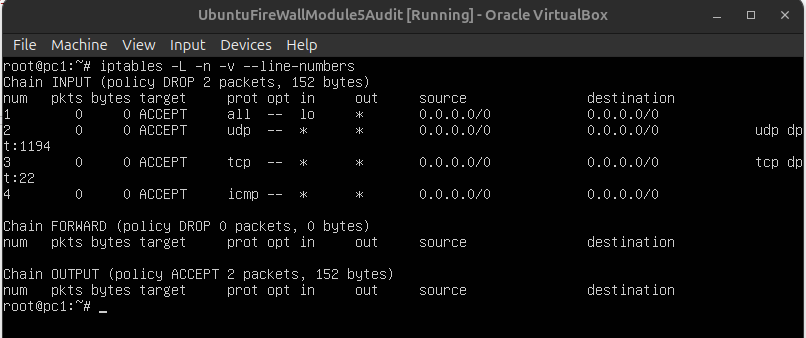
****

Рисунок 4 - Настройка iptables

### **Проверка настройки iptables**

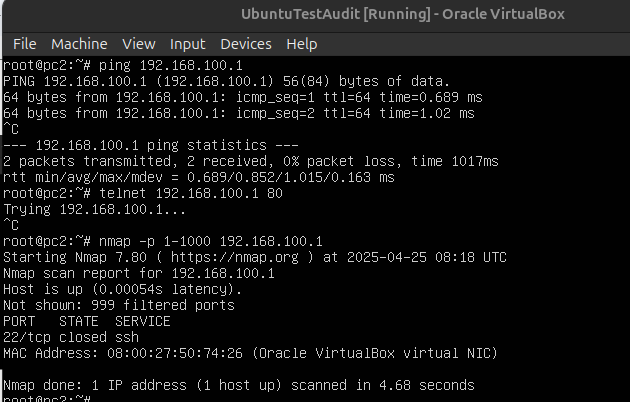


Рисунок 5 - Ha PC2

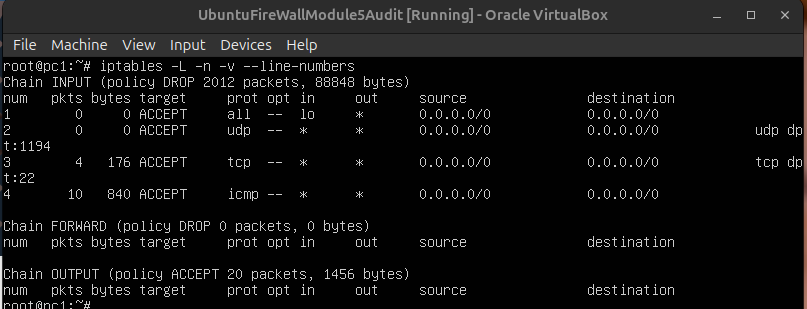


Рисунок 6 - Ha PC1

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)

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

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

|  |  |
| --- | --- |

Рисунок 7 - Установка tcpdump

Установка OpenVPN и easy-rsa

|  |  |
| --- | --- |

Рисунок 8 - Установка OpenVPN и easy-rsa

### **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"**

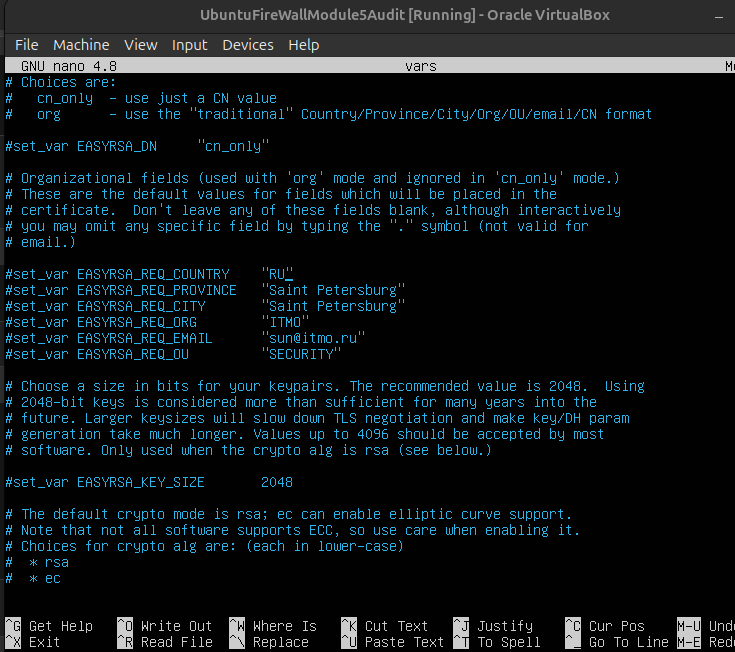


Рисунок 9 - Настройка параметров CA

* Инициализация CA и создание сертификата:

**./easyrsa init-pki**

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

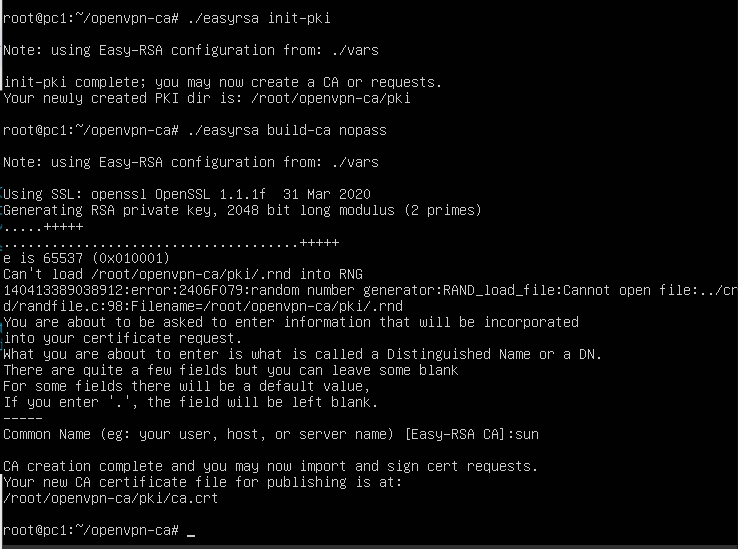


Рисунок 10 - Инициализация CA и создание сертификата

* Создание ключа для сервера:

**./easyrsa gen-req server nopass**

**./easyrsa sign-req server server**

|  |  |
| --- | --- |

Рисунок 11 - Создание ключа для сервера.

* Создание ключа для клиента:

**./easyrsa gen-req client1 nopass**

**./easyrsa sign-req client client1**

|  |  |
| --- | --- |

Рисунок 12 - Создание ключа для клиента.

* Генерация Diffie-Hellman и tls-auth

**./easyrsa gen-dh**

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

|  |  |
| --- | --- |

### **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**

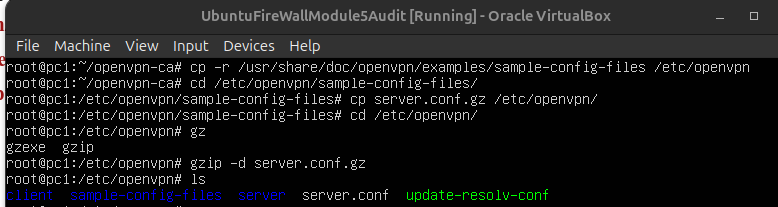
****

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/**

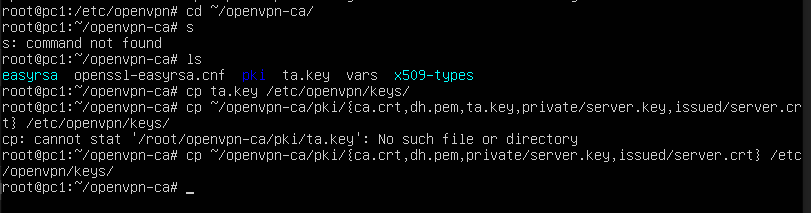


Рисунок 14 -Копирование сертификатов в правильное место.

### **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:~

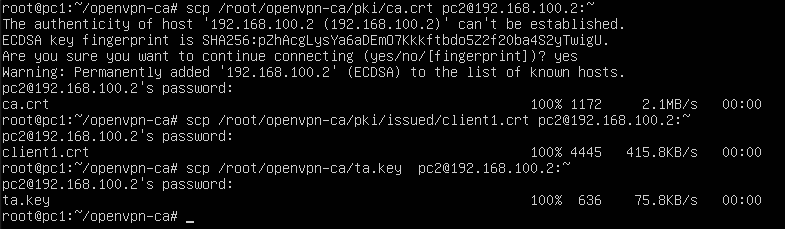


Рисунок 15 -Копирование файлы с PC1 на PC2 с помощью SCP

### **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

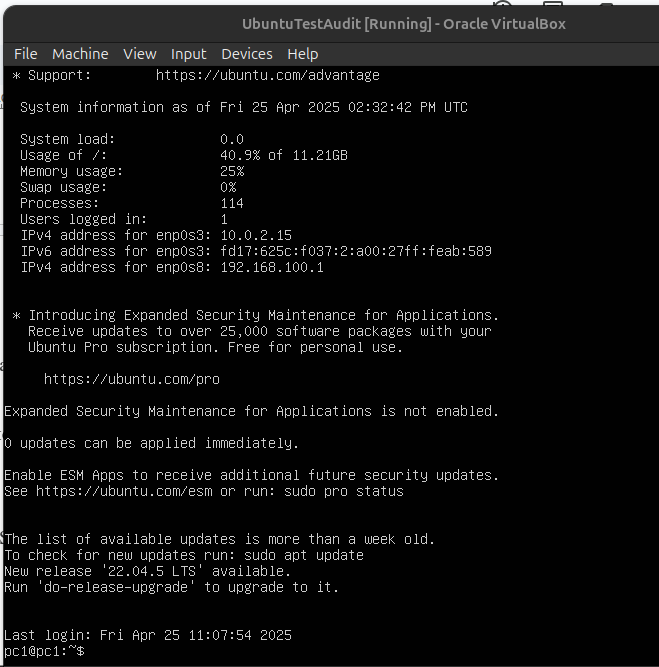


Рисунок 16 - Login на ПК2

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

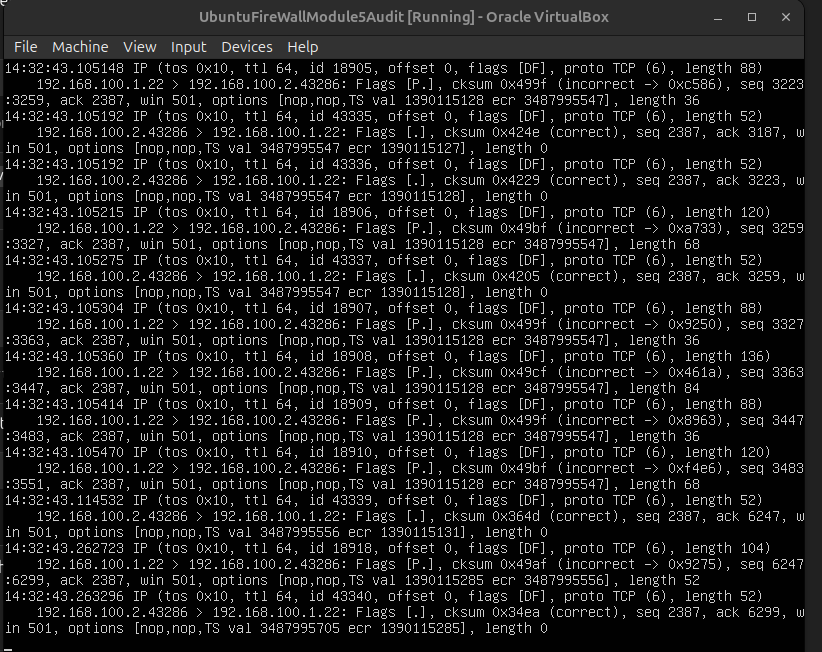


Рисунок 17 - PC1

### **Запуск OpenVPN (шифрование данных).**

На ПК1: sudo systemctl start openvpn@server

Проверка: sudo systemctl status openvpn@server

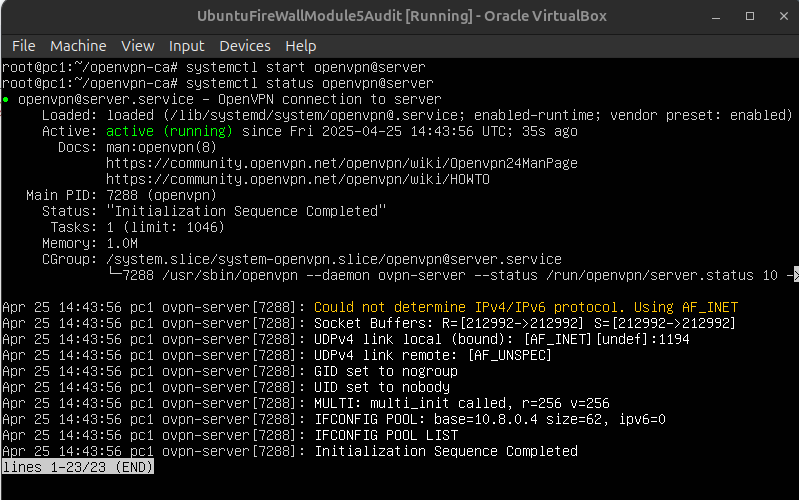


Рисунок 18 - Status PC1

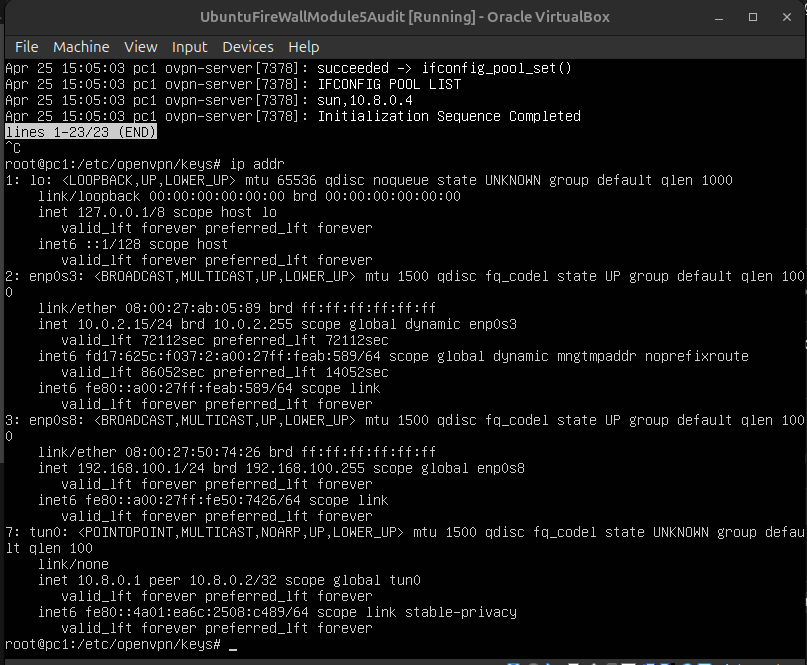


Рисунок 19 - Ip PC1

На ПК2: sudo openvpn --config client.ovpn

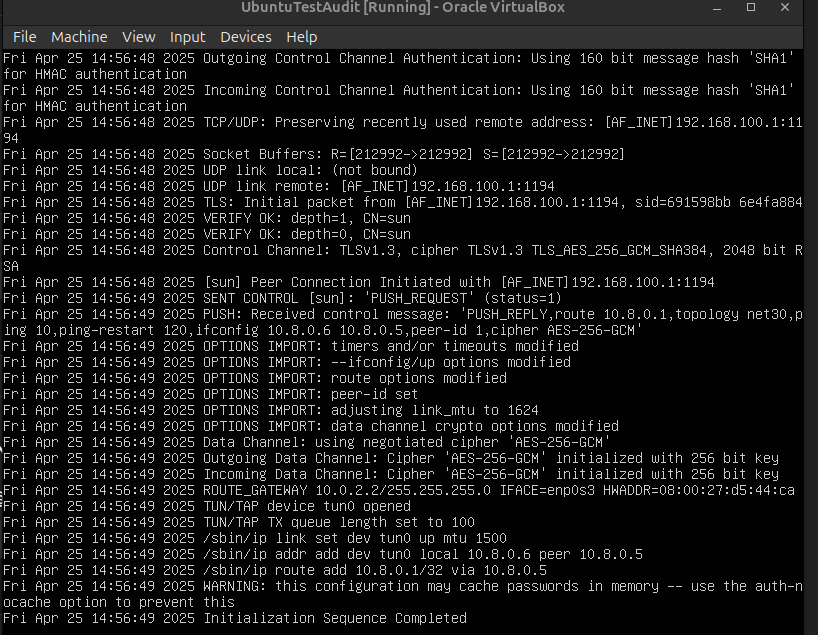


Рисунок 20 - Соединение прошло успешно

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

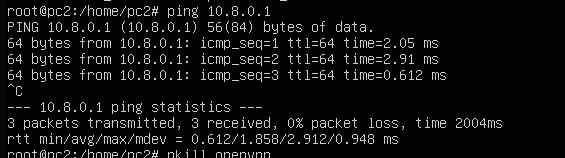


Рисунок 21 - PC2 ping ip address VPN-server



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 |

## Этап 3. Настройка и демонстрация использования цифровой подписи

### **Установка OpenSSL**

sudo apt update

sudo apt install openssl -y

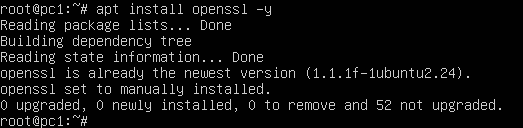


Рисунок 23 - Установка OpenSSL

### **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)

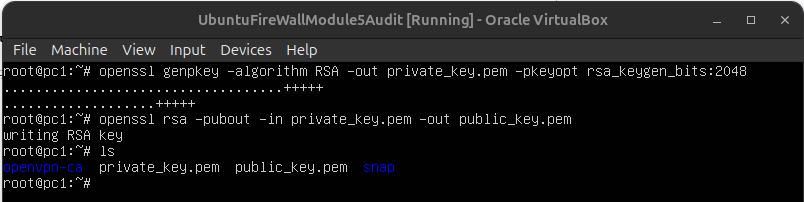


Рисунок 24 - Создание пары ключей RSA (2048 бит)

### **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

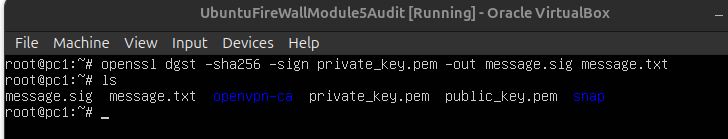


Рисунок 26 - Создание цифровой подписи с помощью приватного ключа

Файл message.sig — это цифровая подпись.

### **Проверка подписи с помощью публичного ключа**

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



Рисунок 26 - подпись действительна

### **Измените содержимое файла**

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

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



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**