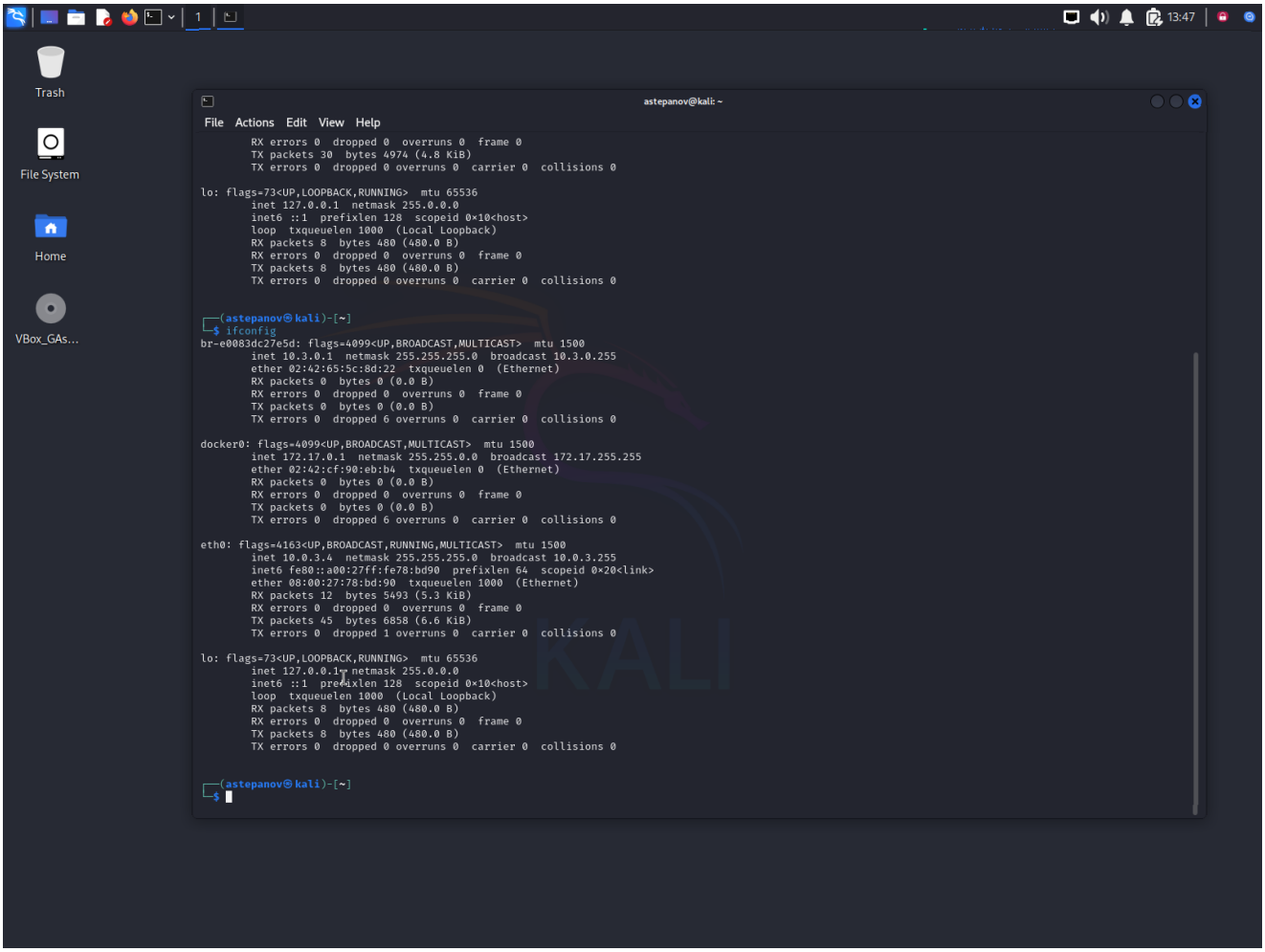
Модуль 3. Виртуальные защищенные каналы связи, DMZ, Wi-Fi (HW)

Лабораторная работа №1 (HW)

Шаг 1. Подготовьте две виртуальные машины

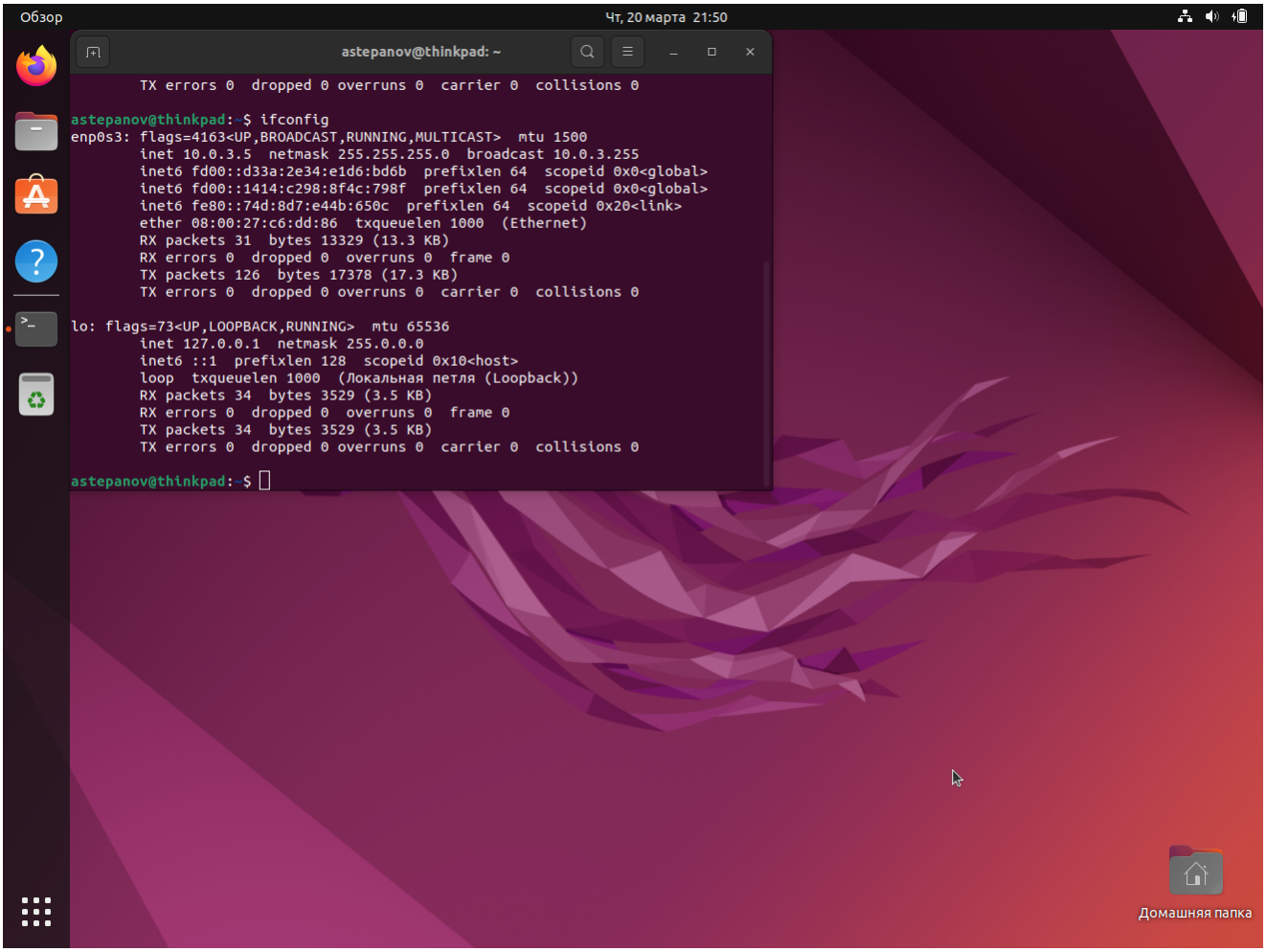
В качестве сервера подготавливаем машину с Kali Linux:

IP-адрес: 10.0.3.4



Для клиентской машины подготовим Ubuntu:

IP-адрес: 10.0.2.5

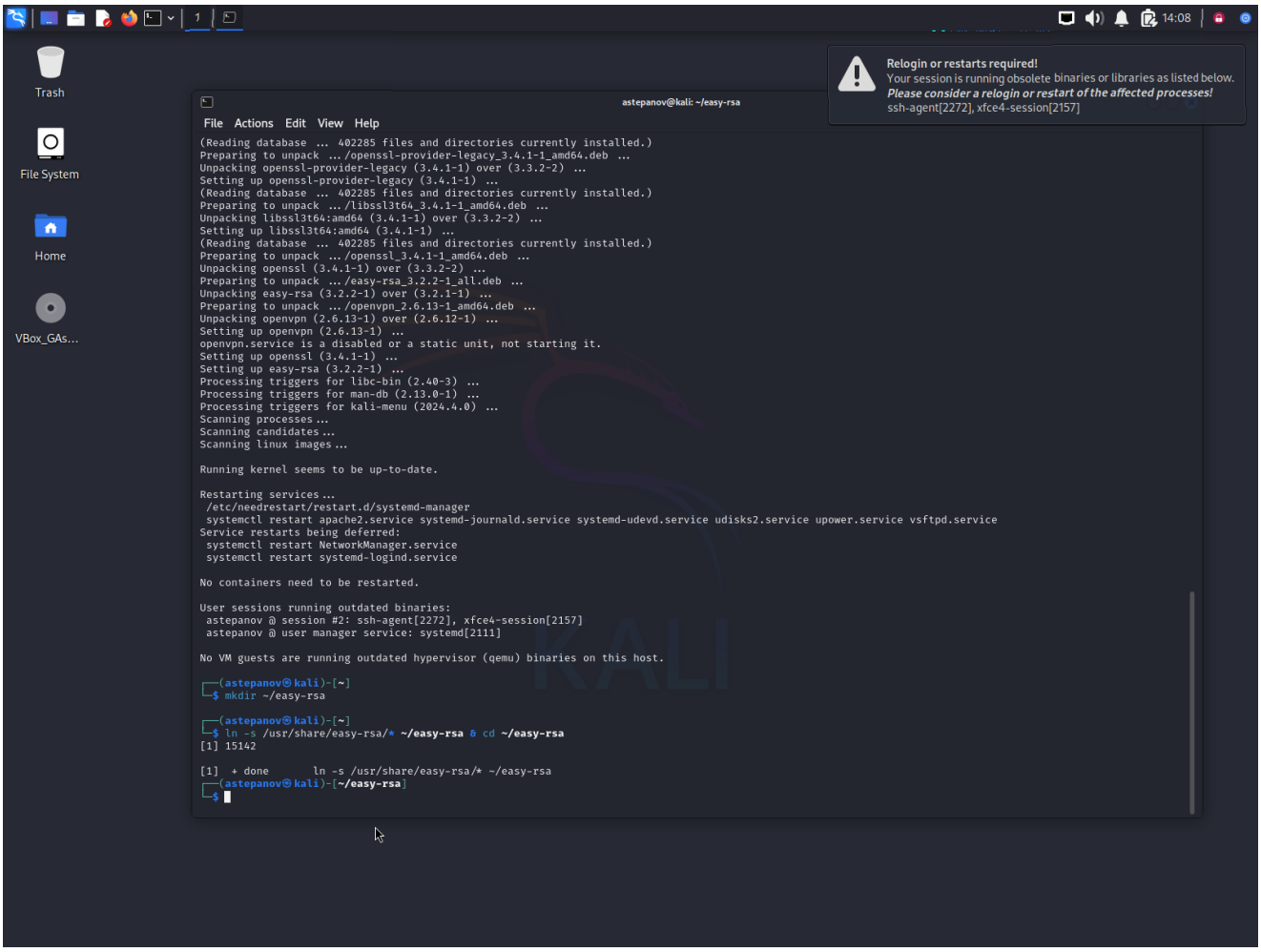


Шаг 2. Проведите настройку OpenVPN

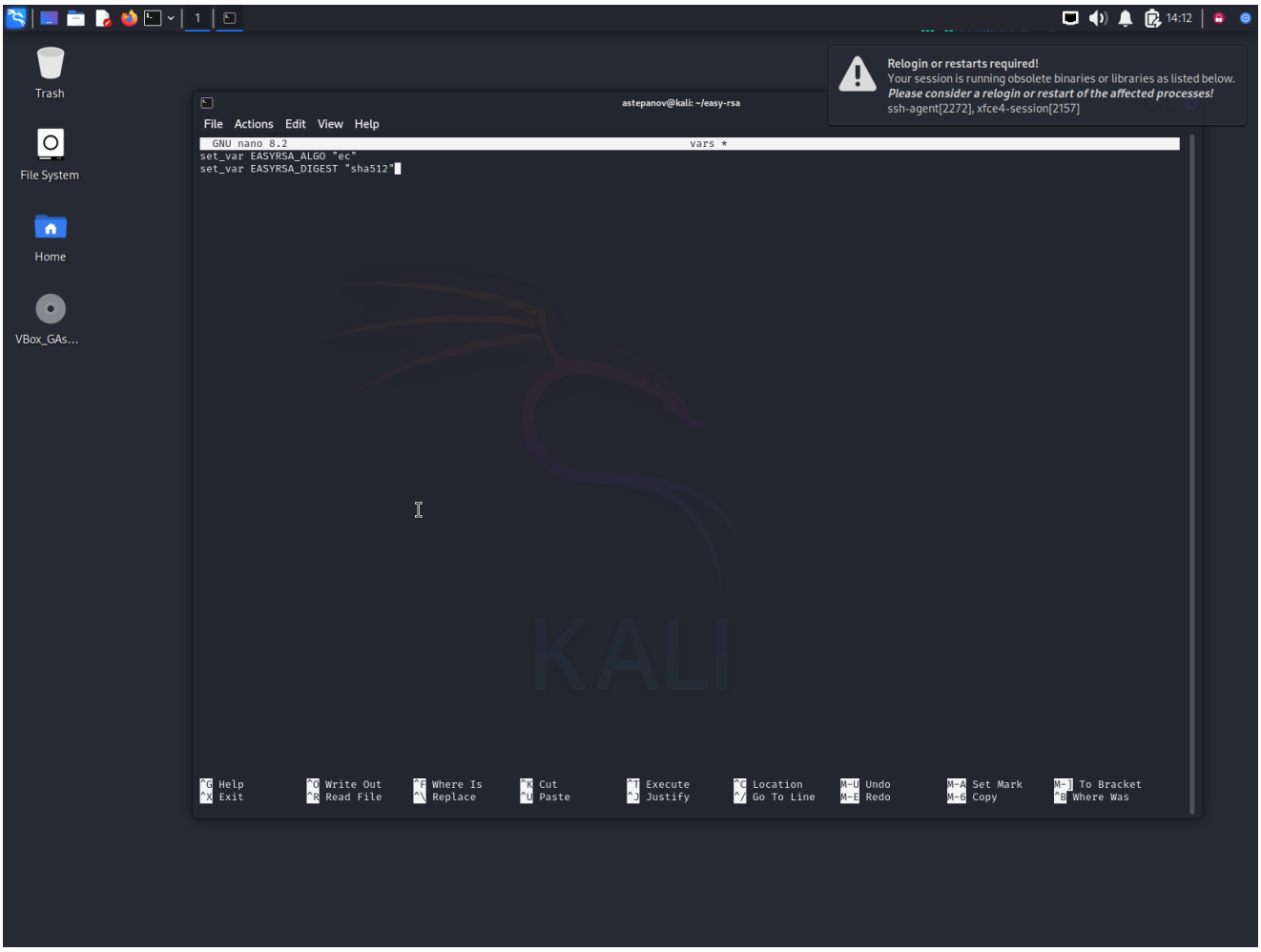
Устанавливаем пакеты на сервер-хосте

a: Создание инфраструктуры открытых ключей (PKI), центра сертификации;

Создаем директорию для инфраструктуры публичных ключей и копируем туда все файлы пакета easy-rsa

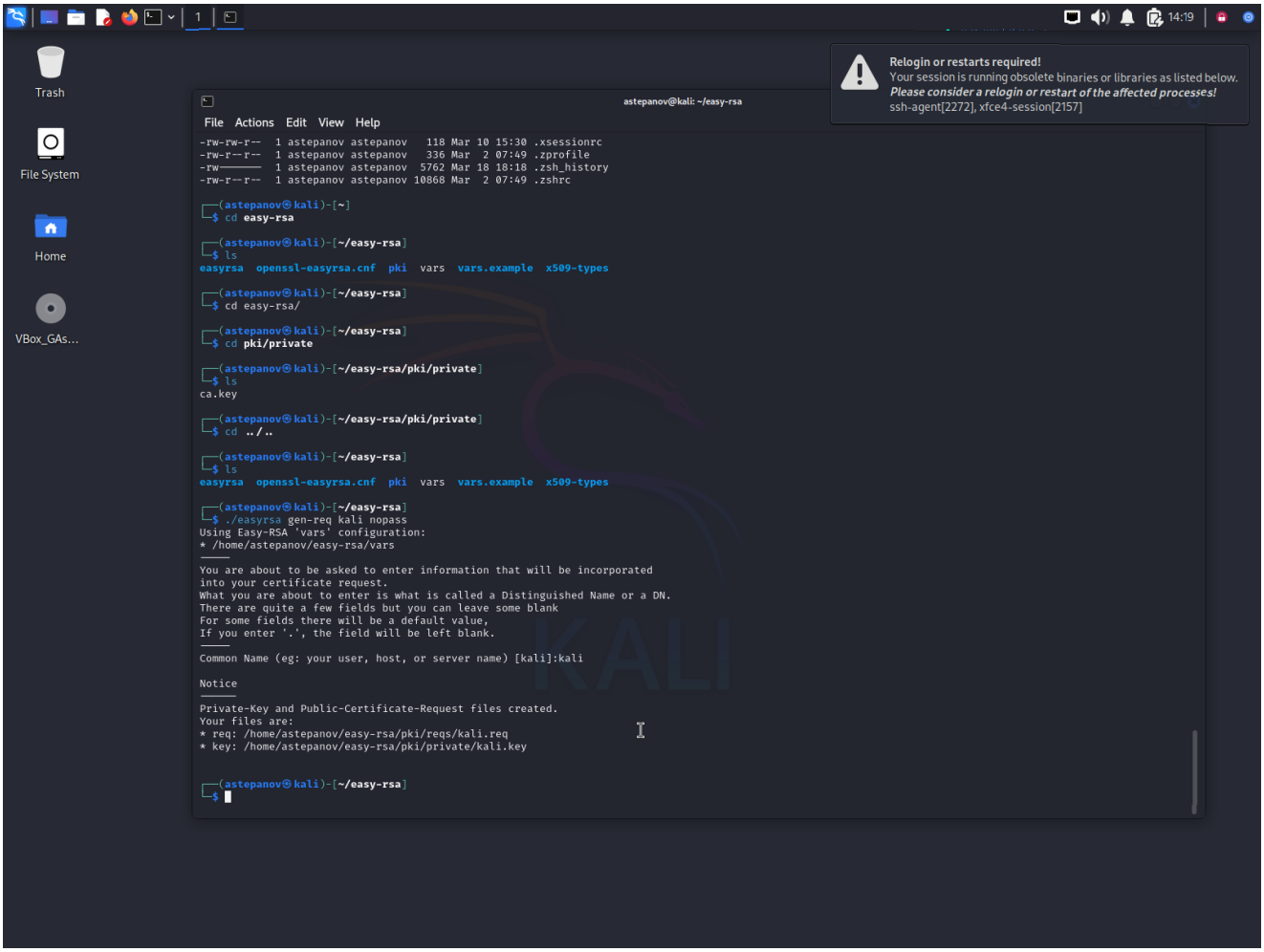


Указание криптографических алгоритмов, которые будут использоваться при генерации ключей и подписей


Trash
File System
Home
VBox_GAs...
1
14:12
Relogin or restarts required!
Your session is running obsolete binaries or libraries as listed below.
Please consider a relogin or restart of the affected processes!
ssh-agent[2272], xfce4-session[2157]
astepanov@kali: ~/easy-rsa
File Actions Edit View Help
GNU nano 8.2 vars *
set_var EASYRSA_ALGO "ec"
set_var EASYRSA_DIGEST "sha512"
KALI
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^B Where Was

Инициализация PKI

b: Создание запроса сертификата и закрытого ключа сервера OpenVPN



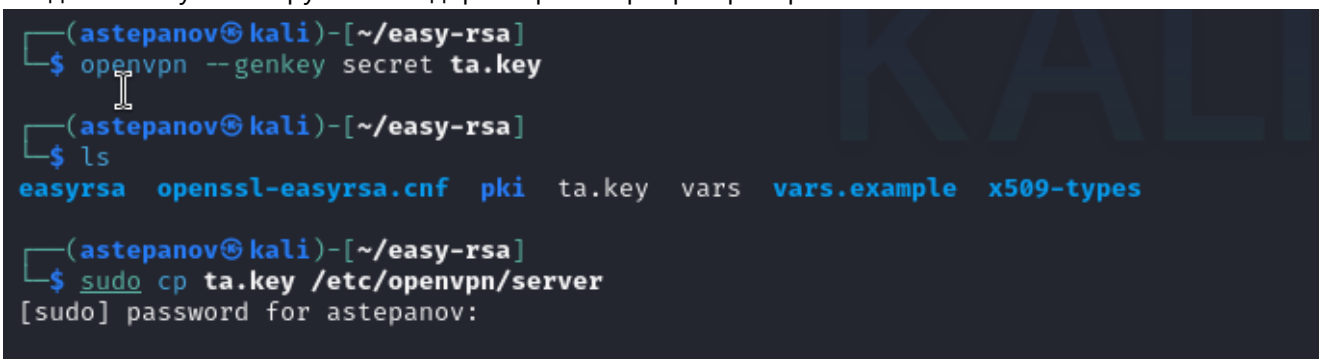
c: Подпись запроса сертификата сервера OpenVPN

Копируем ключ сервера в директорию конфигурации OpenVpn и подписываем запрос центром сертификации. Подписанный сертификат сервера kali.crt вместе с сертификатом центра

сертификации копируем в директорию конфигурации OpenVPN**

d: Настройка криптографических материалов OpenVPN

Создаем ta.key и копируем его в директорию сервера openvpn



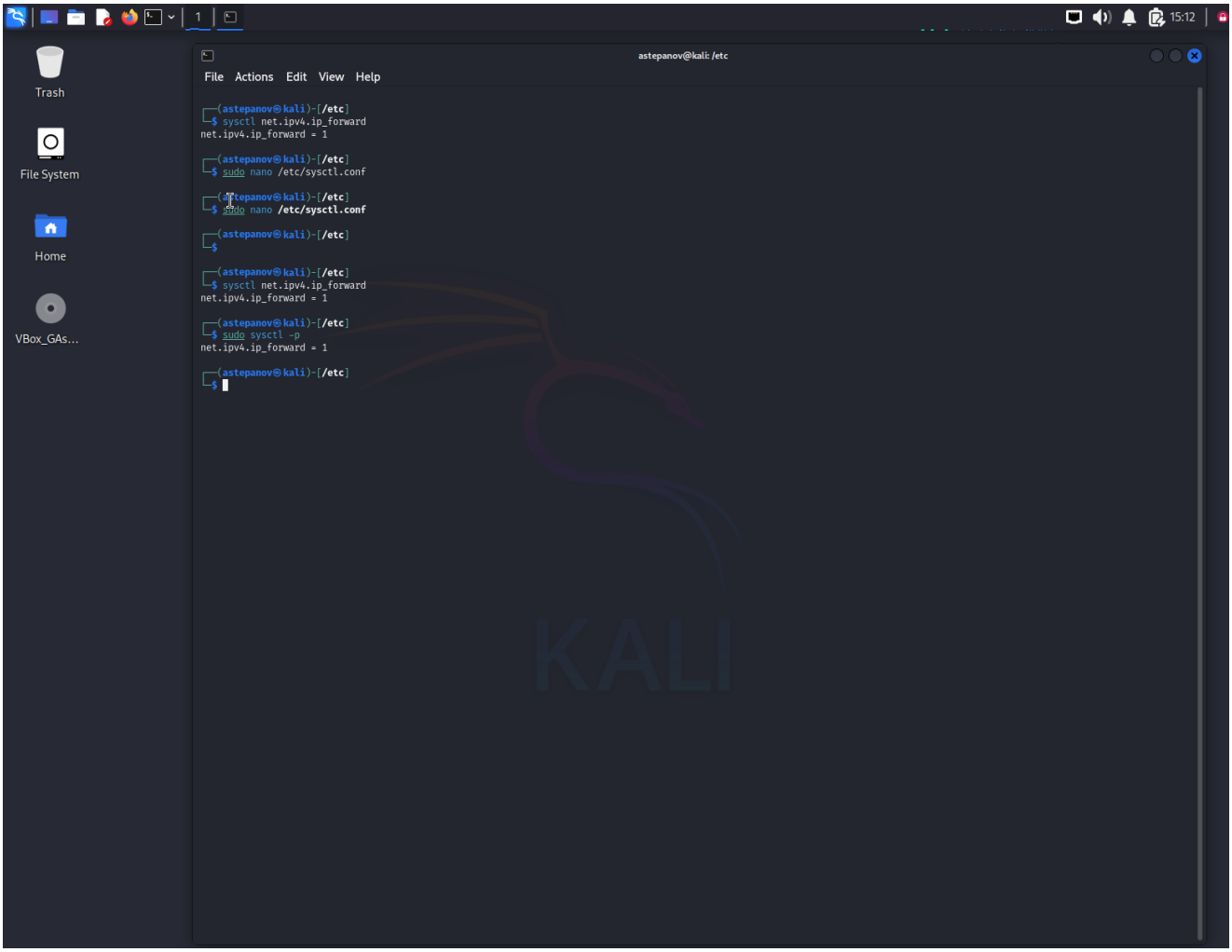
Создаем и подписываем клиентский ключ и копируем все в директорию ключей клиентского конфига

f: Настройка OpenVPN

Настраиваем конфиг openvpn, добавляем параметры шифрования


astepanov@kali: /etc
File Actions Edit View Help
GNU nano 8.2 /etc/openvpn/server/server.conf
#local a.b.c.d
port 1194
;proto tcp
proto udp
;dev tap
dev tun
;dev-node MyTap
ca ca.crt
cert kali.crt
key kali.key # This file should be kept secret
dh dh2048.pem
;data-ciphers AES-256-GCM:AES-128-GCM:?CHACHA20-POLY1305:AES-256-CBC
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100
;server-bridge
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
;learn-address ./script
;push "redirect-gateway def1 bypass-dhcp"
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
;client-to-client
;duplicate-cn
keepalive 10 120
;tls-auth ta.key 0 # This file is secret
tls-crypt ta.key
;cypher AES-256-CBC
cipher AES-256-GCM
auth SHA256
;dh dh2048.pem
dh none
port 1194
proto tcp
;max-clients 100
[Read 79 lines]
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-B Previous ◂ Back
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^B Where Was M-F Next ▸ Forward
Places
Computer
astepanov
Desktop
Recent
Trash
shared
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Net

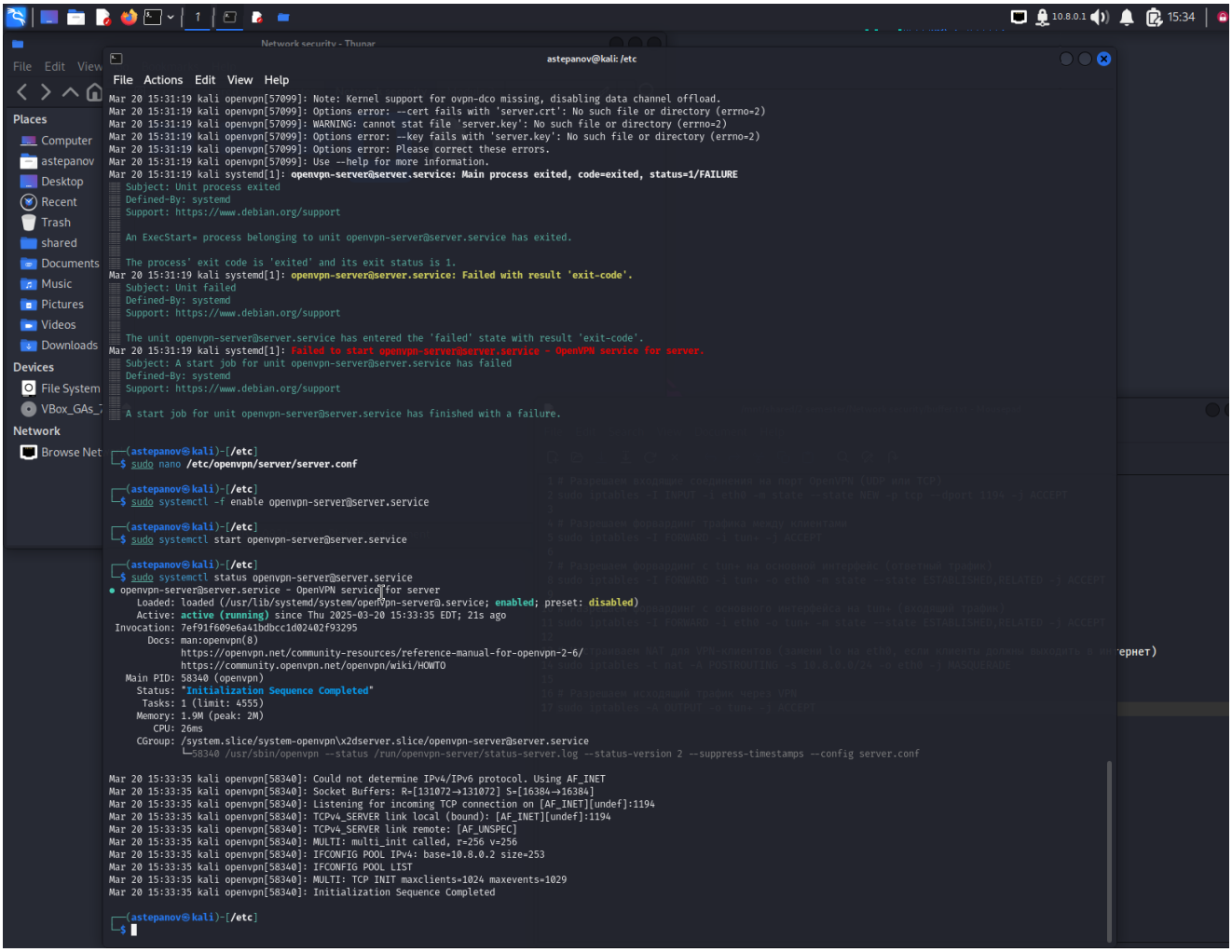
g: Настройка конфигурации сети сервера OpenVPN



h: Настройка межсетевого экрана

```
┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -I INPUT -i eth0 -m state --state NEW -p tcp --dport 1194 -j ACCEPT

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -I FORWARD -i tun+ -j ACCEPT

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -I FORWARD -i tun+ -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -I FORWARD -i eth0 -o tun+ -m state --state ESTABLISHED,RELATED -j ACCEPT

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -A OUTPUT -o tun+ -j ACCEPT

┌──(astepanov㉿kali)-[/etc]
└─$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:1194

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   tun+    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  tun+   eth0    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  tun+   *       0.0.0.0/0            0.0.0.0/0
    0     0 DOCKER-USER  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 DOCKER-ISOLATION-STAGE-1  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  *      docker0  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
    0     0 DOCKER     all  --  *      docker0  0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  docker0 !docker0  0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  docker0 docker0  0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  *      br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
    0     0 DOCKER     all  --  *      br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  br-e0083dc27e5d !br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  br-e0083dc27e5d br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      tun+    0.0.0.0/0            0.0.0.0/0

Chain DOCKER (2 references)
 pkts bytes target     prot opt in     out     source               destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DOCKER-ISOLATION-STAGE-2  all  --  docker0 !docker0  0.0.0.0/0            0.0.0.0/0
    0     0 DOCKER-ISOLATION-STAGE-2  all  --  br-e0083dc27e5d !br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain DOCKER-ISOLATION-STAGE-2 (2 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      docker0  0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  *      br-e0083dc27e5d  0.0.0.0/0            0.0.0.0/0
```

i: Запуск OpenVPN



j: Создание инфраструктуры конфигурации клиентских систем

Копируем шаблон для создания клиентского конфига

Редактируем шаблон клиентского конфига

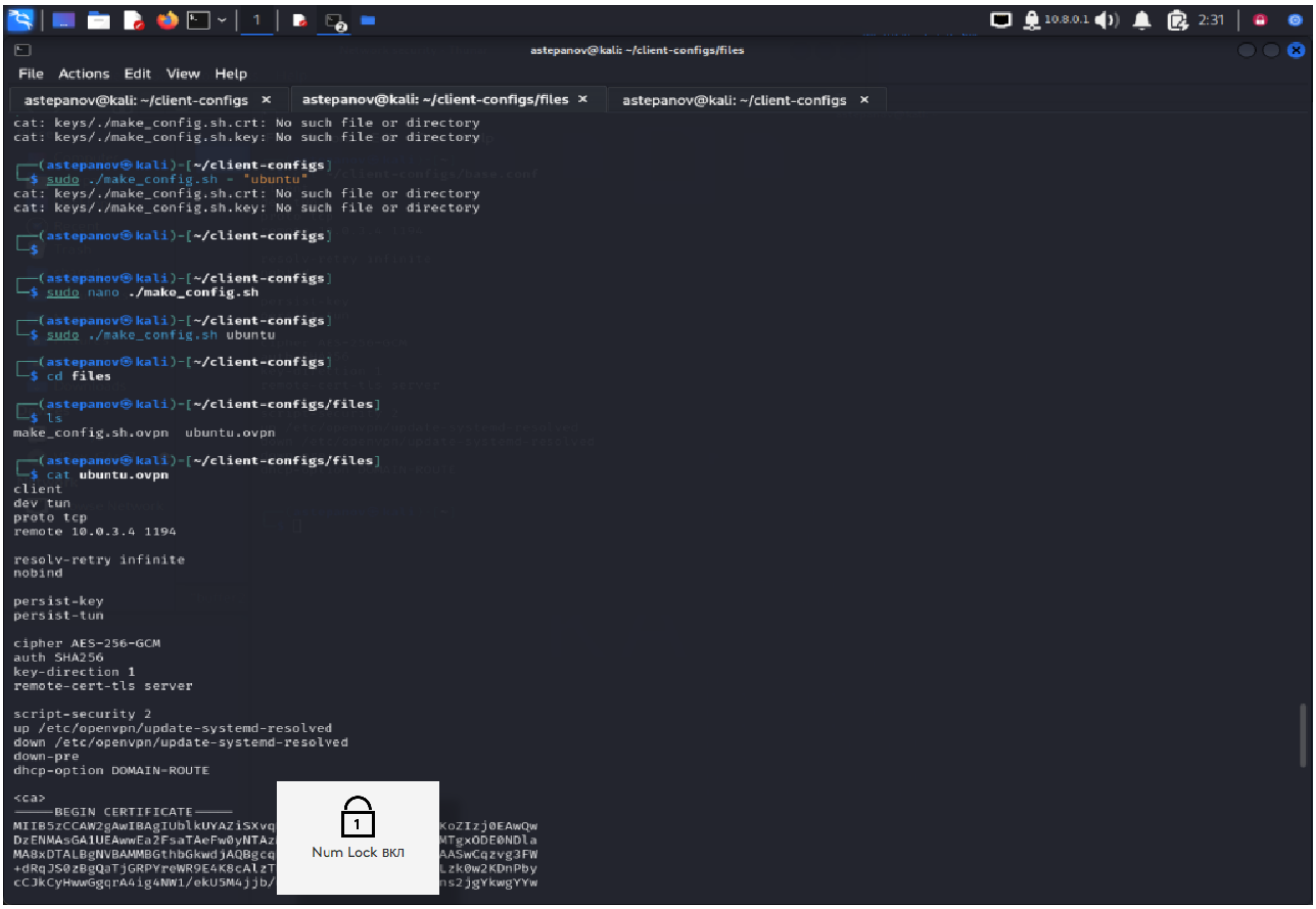
```
File  Actions  Edit  View  Help
┌──(astepanov㉿kali)-[~]
└─$ cat ~/client-configs/base.conf
client
dev tun
proto tcp
remote 10.0.3.4 1194

resolv-retry infinite
nobind

persist-key
persist-tun

cipher AES-256-GCM
auth SHA256
key-direction 1
remote-cert-tls server

script-security 2
up /etc/openvpn/update-systemd-resolved
down /etc/openvpn/update-systemd-resolved
down-pre
dhcp-option DOMAIN-ROUTE

┌──(astepanov㉿kali)-[~]
└─$
```

Создаем скрипт для создания .ovpn конфига из base.conf

```
┌──(astepanov㉿kali)-[~/client-configs]
└─$ cat ./make_config.sh
#!/bin/bash

  KEY_DIR=keys
  OUTPUT_DIR=files
  BASE_CONFIG=base.conf

  CLIENT_NAME=$1

  cat ${BASE_CONFIG} \
  <(echo -e '<ca>') \
  ${KEY_DIR}/ca.crt \
  <(echo -e '</ca>\n<cert>') \
  ${KEY_DIR}/${CLIENT_NAME}.crt \
  <(echo -e '</cert>\n<key>') \
  ${KEY_DIR}/${CLIENT_NAME}.key \
  <(echo -e '</key>\n<tls-crypt>') \
  ${KEY_DIR}/ta.key \
  <(echo -e '</tls-crypt>') \
  > ${OUTPUT_DIR}/${CLIENT_NAME}.ovpn

┌──(astepanov㉿kali)-[~/client-configs]
└─$
```

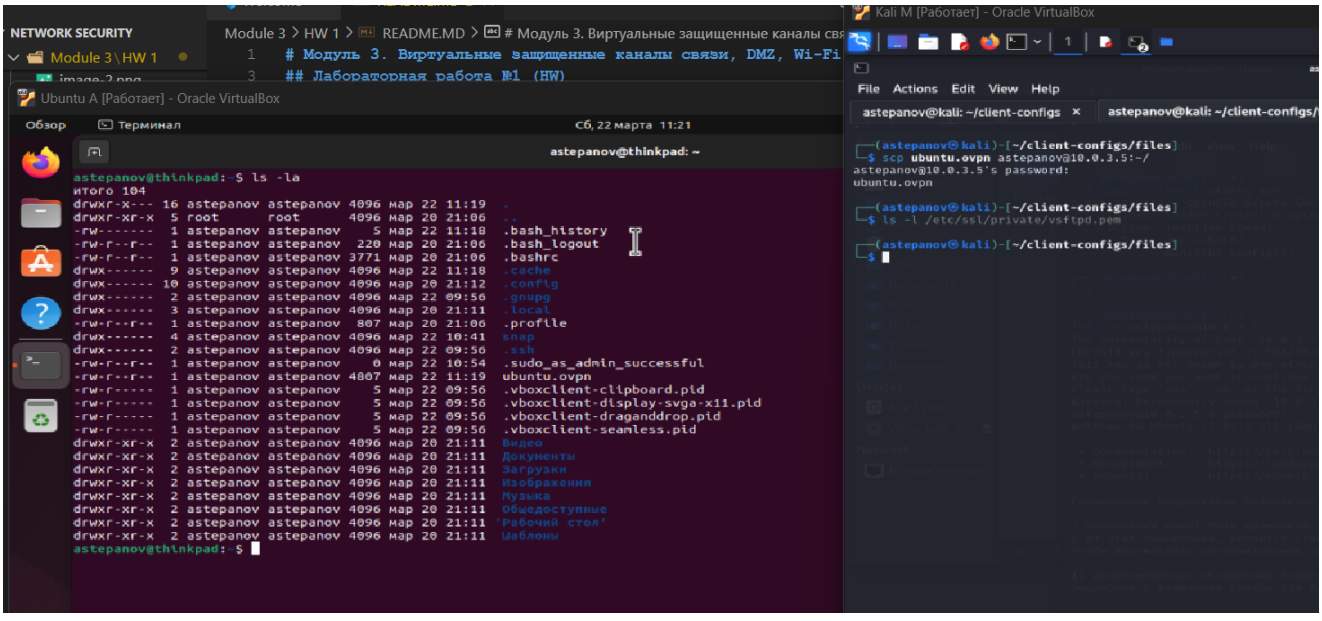
k: создание конфигураций клиентов

Запускаем скрипт make_config.sh и генерируем файл конфига ubuntu.ovpn



i: установка клиентской конфигурации

Копируем .ovpn файл на клиентскую машину

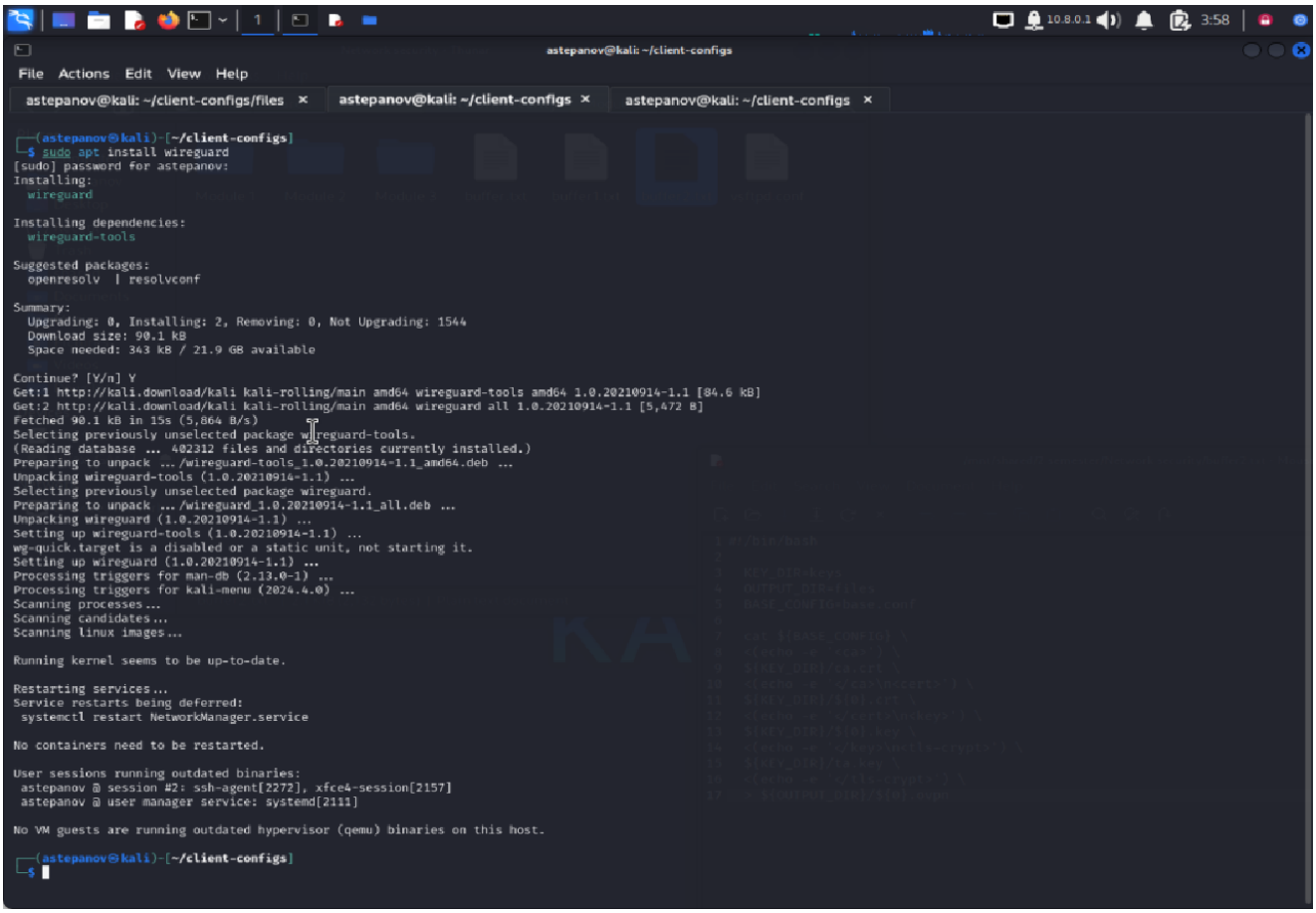


Устанавливаем openvpn клиентскую службу

```
astepanov@thinkpad:~$ sudo apt install -y openvpn-systemd-resolved
[sudo] пароль для astepanov:
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Будут установлены следующие дополнительные пакеты:
  libnss-resolve
Следующие НОВЫЕ пакеты будут установлены:
  libnss-resolve openvpn-systemd-resolved
Обновлено 0 пакетов, установлено 2 новых пакетов, для удаления отмечено 0 пакетов, и 3 пакетов не обновлено.
Необходимо скачать 76,9 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 352 kB.
Пол:1 http://ru.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libnss-resolve amd64 249.11-0ubuntu3.12 [64,5 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 openvpn-systemd-resolved amd64 1.3.0-3.1 [12,4 kB]
Получено 76,9 kB за 11с (6 818 B/s)
Выбор ранее не выбранного пакета libnss-resolve:amd64.
(Чтение базы данных … на данный момент установлено 194568 файлов и каталогов.)
Подготовка к распаковке …/libnss-resolve_249.11-0ubuntu3.12_amd64.deb …
Распаковывается libnss-resolve:amd64 (249.11-0ubuntu3.12) …
Выбор ранее не выбранного пакета openvpn-systemd-resolved.
Подготовка к распаковке …/openvpn-systemd-resolved_1.3.0-3.1_amd64.deb …
Распаковывается openvpn-systemd-resolved (1.3.0-3.1) …
Настраивается пакет libnss-resolve:amd64 (249.11-0ubuntu3.12) …
First installation detected...
Checking NSS setup...
Настраивается пакет openvpn-systemd-resolved (1.3.0-3.1) …
Обрабатываются триггеры для libc-bin (2.35-0ubuntu3.9) …
Обрабатываются триггеры для man-db (2.10.2-1) …
```

Шаг 3. Проведите настройку WireGuard

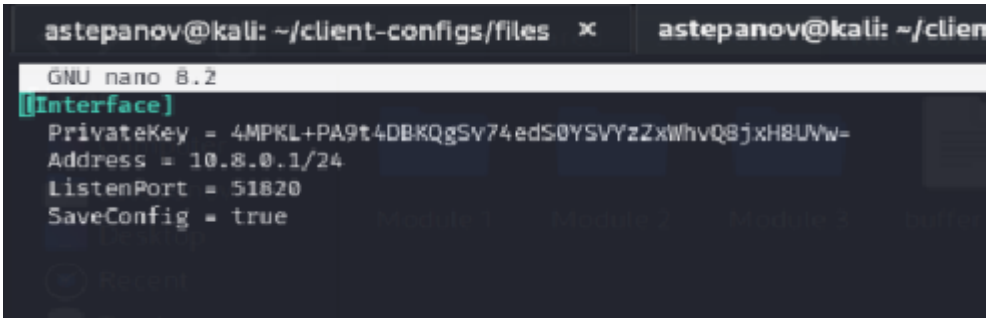
Устанавливаем Wireguard на сервере



a. генерация пары ключей для сервера Wireguard

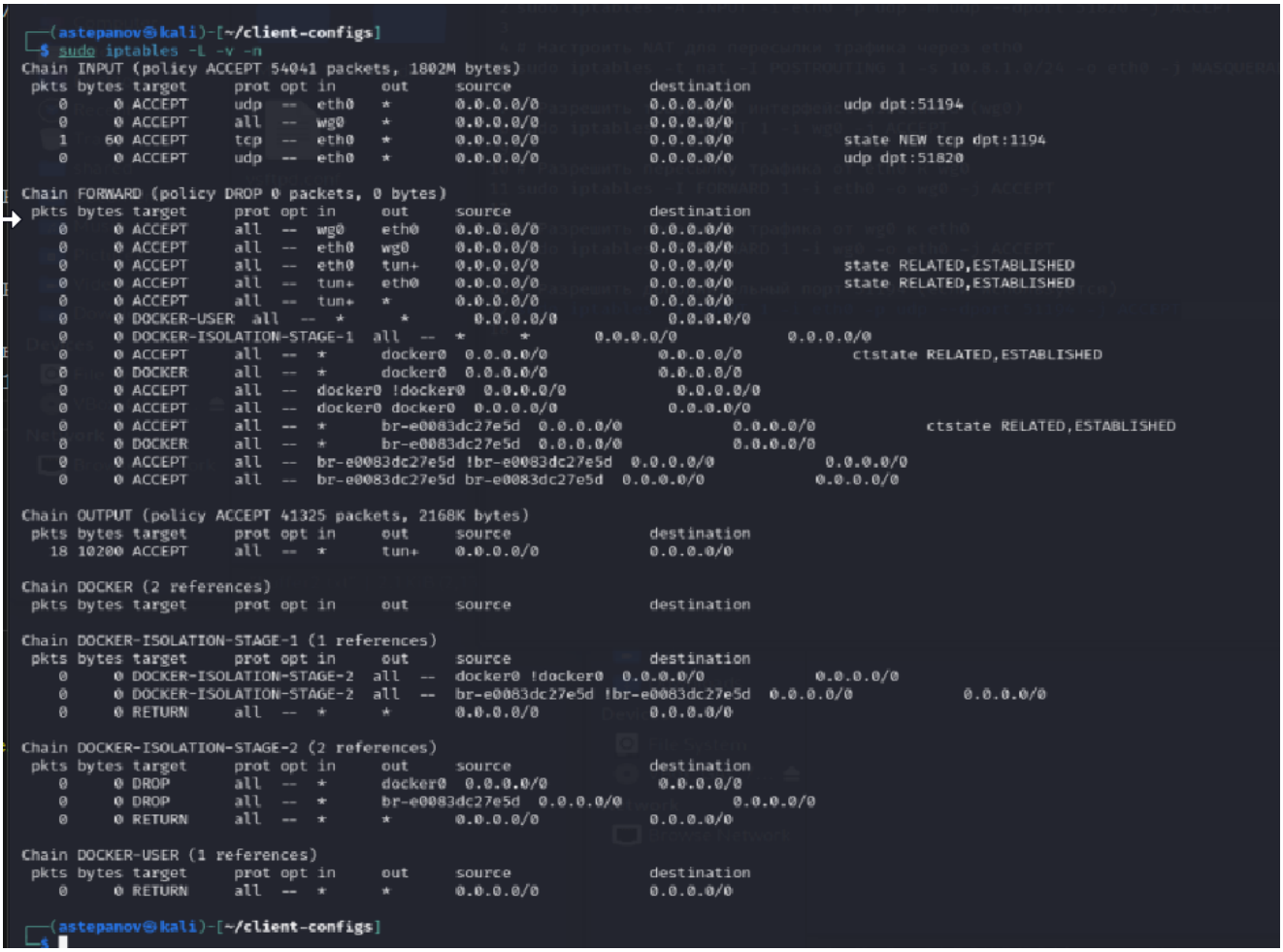
```
┌──(astepanov㉿kali)-[~/client-configs]
└─$ wg genkey | sudo tee /etc/wireguard/private.key
4MPKL+PA9t4DBKQgSv74ed50YSVYzZxWhvQ8jxH8UVw=

┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo chmod go= /etc/wireguard/private.key

┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo cat /etc/wireguard/private.key | wg pubkey | sudo tee /etc/wireguard/public.key
lO50Y4etv+UxNJ1EEHQV9YStkQ5HNL4V3L48DTFKNiM=

┌──(astepanov㉿kali)-[~/client-configs]
└─$
```

b. создание файла конфигурации сервера



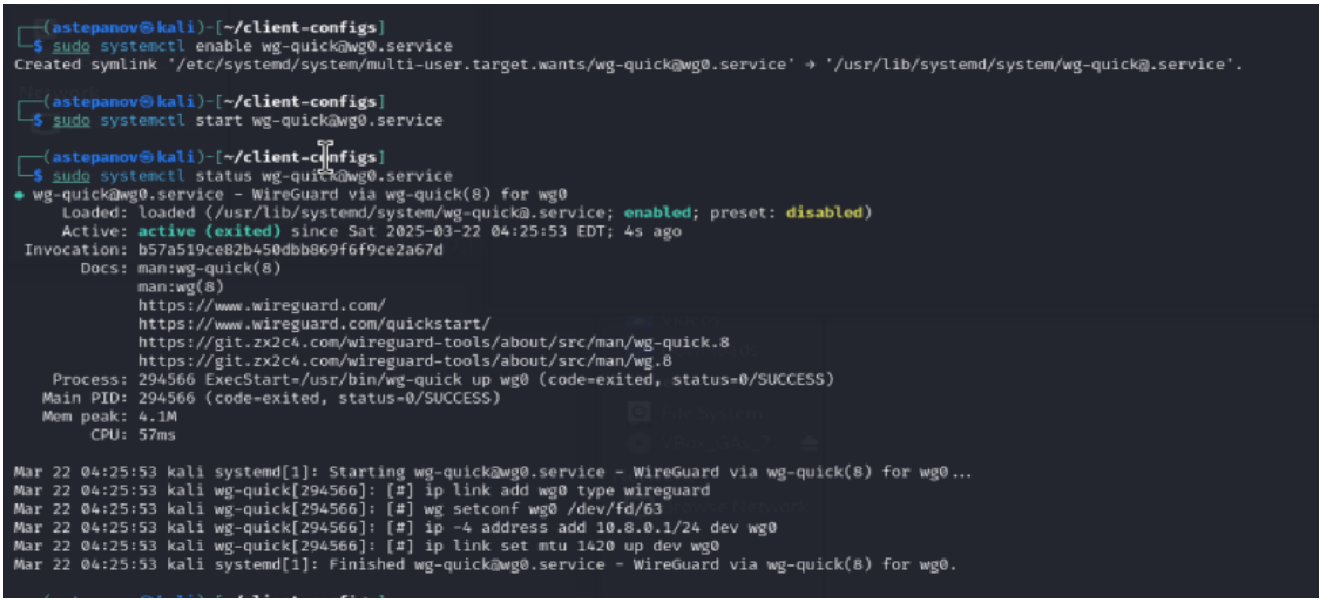
c. настройка межсетевого экрана

Добавляем необходимые настройки в iptables

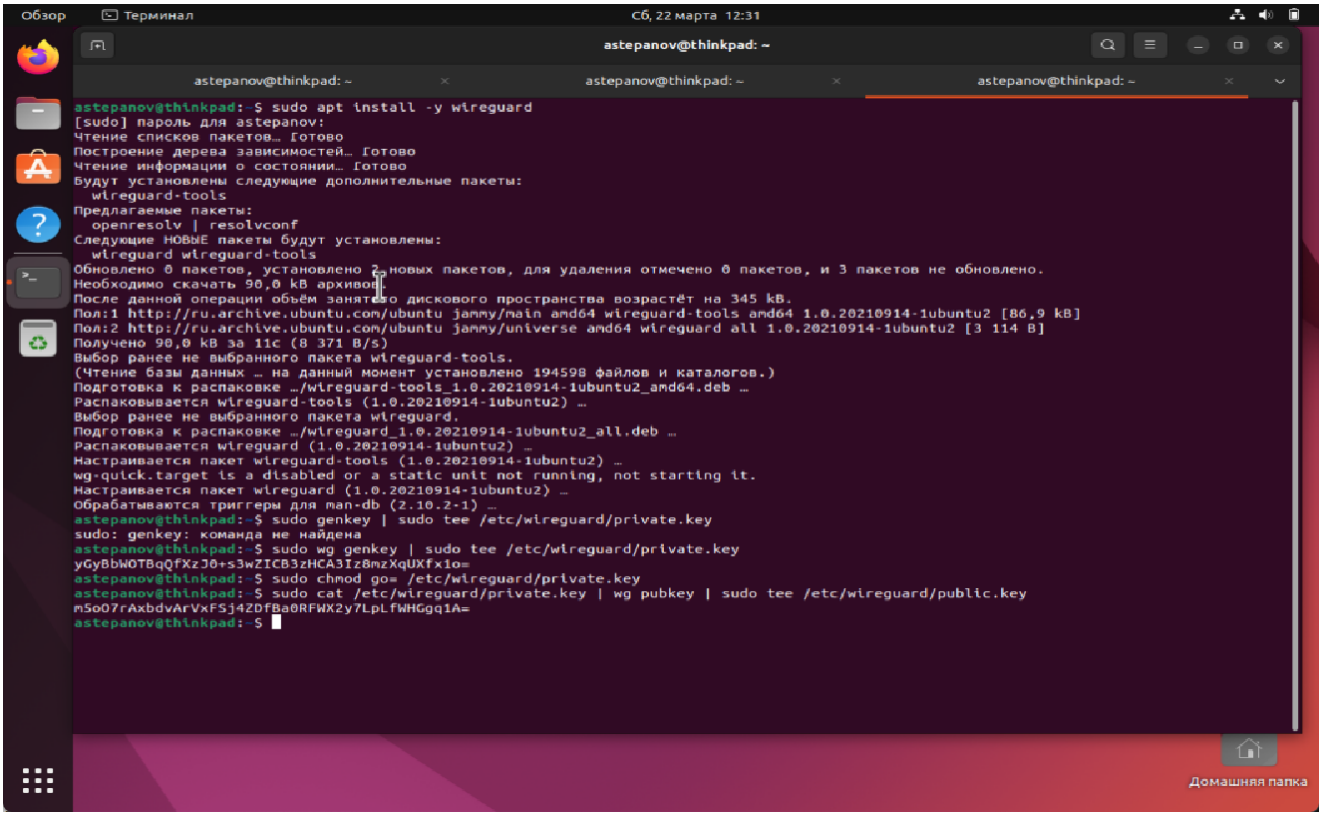


d. запуск Wireguard

Поднимаем wg-сервер



e. генерация пары ключей для клиента Wireguard



f. создание файла конфигурации клиента

g. добавление публичного ключа клиента на сервер Wireguard

```
astepanov@thinkpad:~$ sudo wg set wg0 peer m5oO7rAxbdvArVxF5j4ZDfBa0RFWX2y7LpLfWHGgq1A= allowed-ips 10.8.0.2
Unable to modify interface: No such device
astepanov@thinkpad:~$ sudo wg-quick up wg0
wg-quick: `/etc/wireguard/wg0.conf' does not exist
astepanov@thinkpad:~$ sudo wg show
astepanov@thinkpad:~$ sudo wg-quick up ~/client-configs/wg0.conf
Warning: `/home/astepanov/client-configs/wg0.conf' is world accessible
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.8.0.2/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
astepanov@thinkpad:~$ sudo wg show
interface: wg0
  public key: m5oO7rAxbdvArVxF5j4ZDfBa0RFWX2y7LpLfWHGgq1A=
  private key: (hidden)
  listening port: 51030

peer: lO50Y4etv+UxNJ1EEHQV9Y5tkQ5HNL4V3L48DTFKNiM=
  endpoint: 10.0.3.4:51820
  allowed ips: 10.8.0.0/24
astepanov@thinkpad:~$ sudo cat ~/client-configs/wg0.conf
[Interface]
  PrivateKey = yGyBbWOTBqQfXzJ0+s3wZICB3zHCA3Iz8mzXqUXfx1o=
  Address = 10.8.0.2/24
  [Peer]
  PublicKey = lO50Y4etv+UxNJ1EEHQV9Y5tkQ5HNL4V3L48DTFKNiM=
  AllowedIPs = 10.8.0.0/24
  Endpoint = 10.0.3.4:51820
astepanov@thinkpad:~$
```

Шаг 4. Произведите соединение поочередно каждой из технологий. Замерьте скорость

OpenVPN

Подключаемся через openvpn

```
astepanov@thinkpad:~$ sudo openvpn --config ubuntu.ovpn
2025-03-22 11:47:36 OpenVPN 2.5.11 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Sep 17
 2024
2025-03-22 11:47:36 library versions: OpenSSL 3.0.2 15 Mar 2022, LZO 2.10
2025-03-22 11:47:36 NOTE: the current --script-security setting may allow this configuration to call user-defined scripts
2025-03-22 11:47:36 TCP/UDP: Preserving recently used remote address: [AF_INET]10.0.3.4:1194
2025-03-22 11:47:36 Attempting to establish TCP connection with [AF_INET]10.0.3.4:1194 [nonblock]
2025-03-22 11:47:36 TCP connection established with [AF_INET]10.0.3.4:1194
2025-03-22 11:47:36 TCP_CLIENT link local: (not bound)
2025-03-22 11:47:36 TCP_CLIENT link remote: [AF_INET]10.0.3.4:1194
2025-03-22 11:47:36 [kali] Peer Connection Initiated with [AF_INET]10.0.3.4:1194
2025-03-22 11:47:36 TUN/TAP device tun0 opened
2025-03-22 11:47:36 net_iface_mtu_set: mtu 1500 for tun0
2025-03-22 11:47:36 net_iface_up: set tun0 up
2025-03-22 11:47:36 net_addr_v4_add: 10.8.0.2/24 dev tun0
2025-03-22 11:47:36 /etc/openvpn/update-systemd-resolved tun0 1500 1626 10.8.0.2 255.255.255.0 init
<14>Mar 22 11:47:36 update-systemd-resolved: Link 'tun0' coming up
<14>Mar 22 11:47:36 update-systemd-resolved: Adding DNS Routed Domain DOMAIN-ROUTE
<14>Mar 22 11:47:36 update-systemd-resolved: SetLinkDomains(3 1 DOMAIN-ROUTE true)
2025-03-22 11:47:36 Initialization Sequence Completed
^[
```

Соединение успешно установлено...

Пингуем адрес сервера, для проверки работы vpn

```
astepanov@thinkpad: ~
astepanov@thinkpad: ~   astepanov@thinkpad: ~
astepanov@thinkpad:~$ ping 10.0.3.5
PING 10.0.3.5 (10.0.3.5) 56(84) bytes of data.
64 bytes from 10.0.3.5: icmp_seq=1 ttl=64 time=0.205 ms
64 bytes from 10.0.3.5: icmp_seq=2 ttl=64 time=0.386 ms
^C
--- 10.0.3.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.205/0.295/0.386/0.090 ms
astepanov@thinkpad:~$ ^C
astepanov@thinkpad:~$ ping 10.0.3.4
PING 10.0.3.4 (10.0.3.4) 56(84) bytes of data.
64 bytes from 10.0.3.4: icmp_seq=1 ttl=64 time=2.17 ms
64 bytes from 10.0.3.4: icmp_seq=2 ttl=64 time=3.25 ms
64 bytes from 10.0.3.4: icmp_seq=3 ttl=64 time=0.791 ms
64 bytes from 10.0.3.4: icmp_seq=4 ttl=64 time=0.802 ms
^C
--- 10.0.3.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3236ms
rtt min/avg/max/mdev = 0.791/1.752/3.246/1.028 ms
astepanov@thinkpad:~$
```

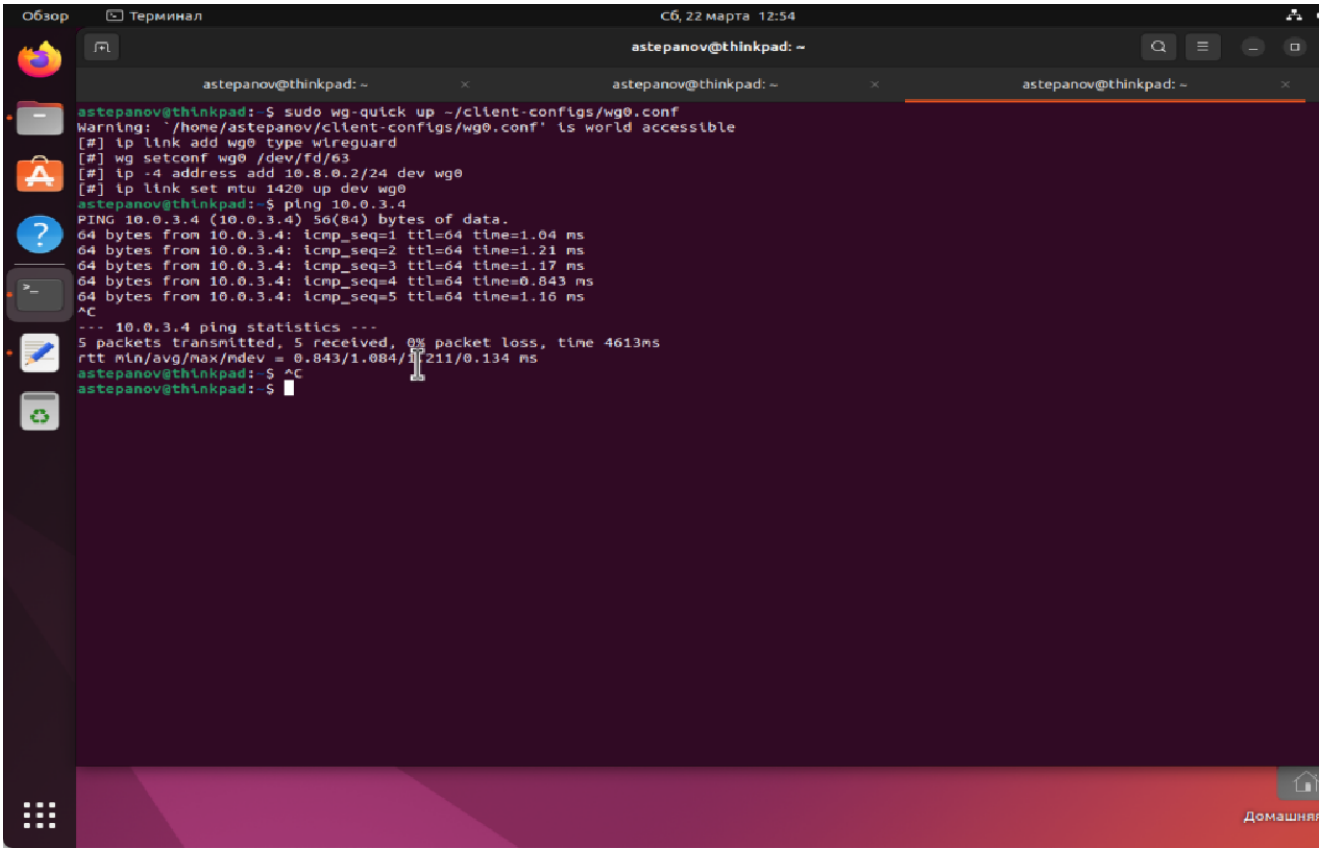
Сервер пингуется успешно...

Замеряем скорость соединения с помощью утилиты iperf

```
Обрабатываются триггеры для libc-bin (2.35-0ubuntu3.9) …
astepanov@thinkpad:~$ iperf3 -c 10.0.3.4
Connecting to host 10.0.3.4, port 5201
[  5] local 10.0.3.5 port 60562 connected to 10.0.3.4 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   167 MBytes  1.40 Gbits/sec  274    286 KBytes
[  5]   1.00-2.00   sec   168 MBytes  1.41 Gbits/sec  228    223 KBytes
[  5]   2.00-3.00   sec   181 MBytes  1.51 Gbits/sec  180    233 KBytes
[  5]   3.00-4.00   sec   184 MBytes  1.54 Gbits/sec   90    440 KBytes
[  5]   4.00-5.00   sec   167 MBytes  1.40 Gbits/sec   90    413 KBytes
[  5]   5.00-6.00   sec   166 MBytes  1.39 Gbits/sec  101    420 KBytes
[  5]   6.00-7.00   sec   154 MBytes  1.29 Gbits/sec  218    238 KBytes
[  5]   7.00-8.00   sec   155 MBytes  1.30 Gbits/sec  134    318 KBytes
[  5]   8.00-9.00   sec   166 MBytes  1.39 Gbits/sec  135    329 KBytes
[  5]   9.00-10.00  sec   144 MBytes  1.21 Gbits/sec  225    253 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.61 GBytes  1.38 Gbits/sec  1675             sender
[  5]   0.00-10.01  sec  1.61 GBytes  1.38 Gbits/sec                   receiver

iperf Done.
```

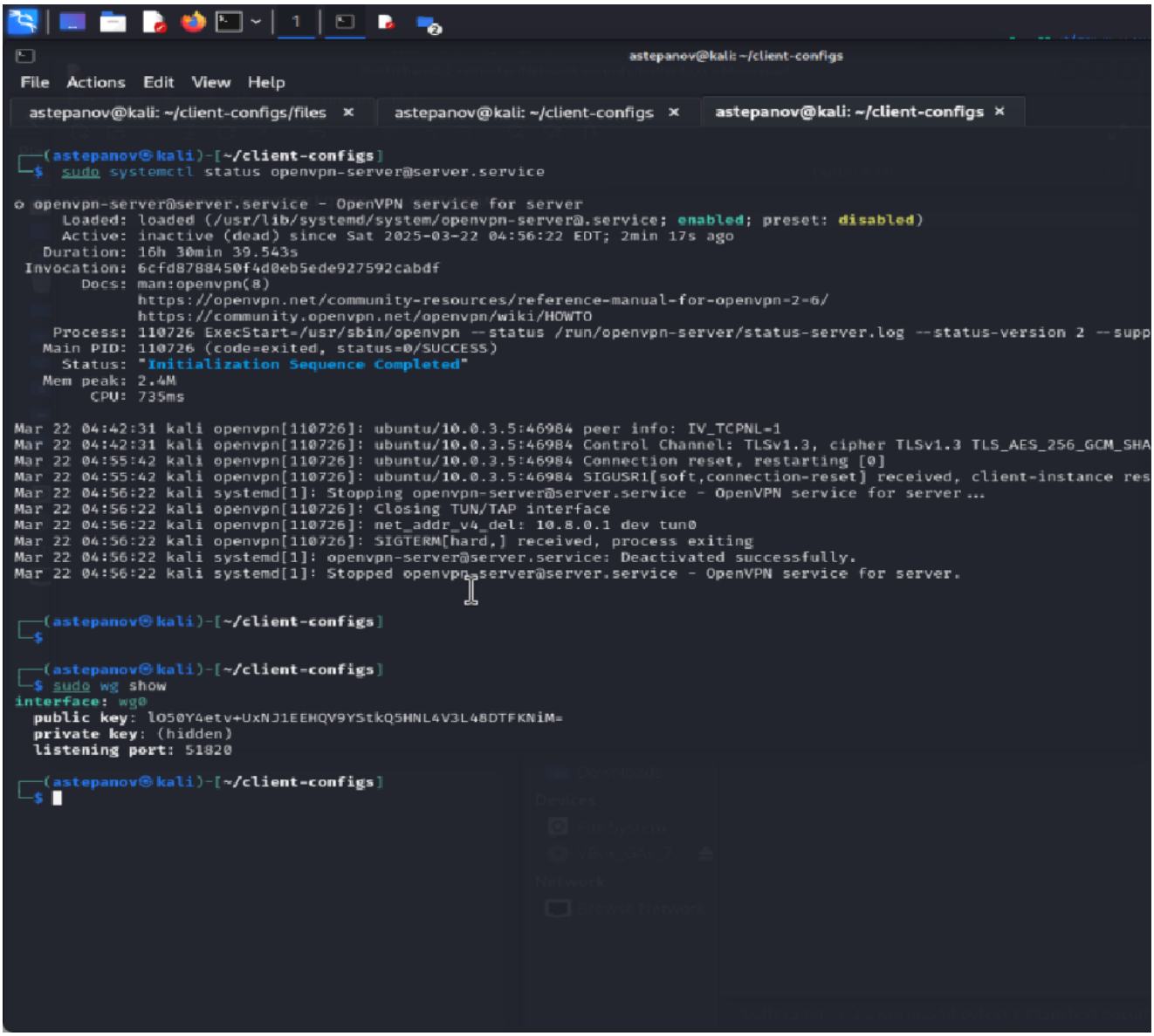
WireGuard

Подключаемся к wireguard серверу, проверяем пинг

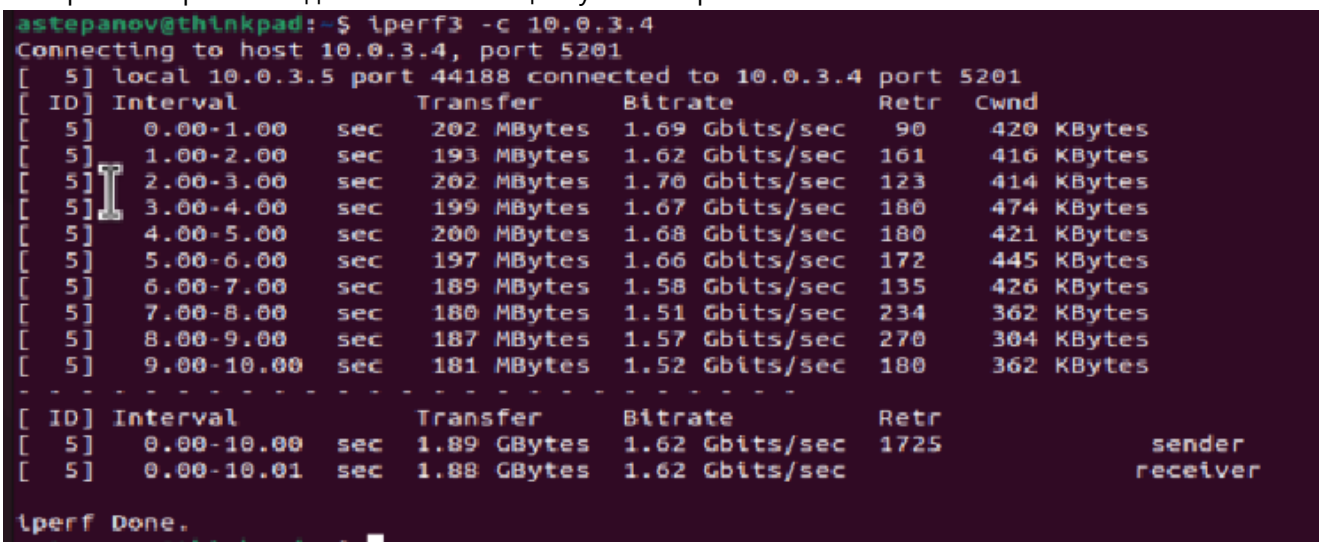


Соединение успешно, пинг проходит...

Перед тем как замерить скорость - следует отключить ранее запущенный openvpn, проверяем что сервис openvpn выключен



Замеряем скорость соединения с помощью утилиты iperf



Шаг 5. Сделайте выводы в соответствии с полученными результатами, основываясь на тонкостях двух изучаемых протоколов

Результатам сравнения скорости передачи данных с помощью утилиты **iperf** показали следующие результаты:

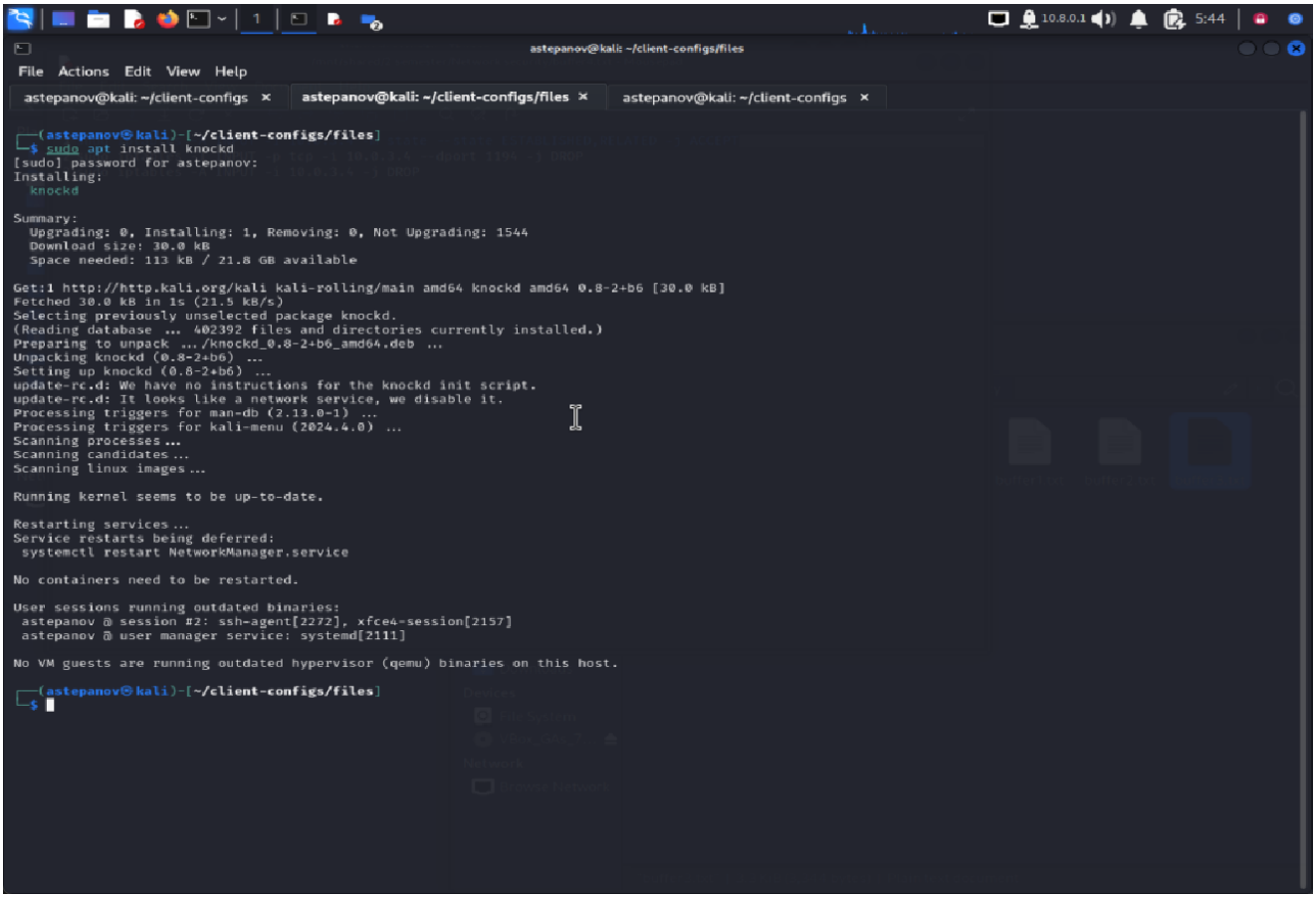
- Средняя скорость передачи данных через OpenVPN составила 1,38 Gb/sec
- Средняя скорость передачи данных через WireGuard составила 1,62 Gb/sec

Таким образом, практический замер скорости VPN-соединений данных протоколов подтверждает теоретическую информацию о том, что скорость передачи данных через WireGuard несколько выше.

WireGuard обычно быстрее, чем OpenVPN, благодаря меньшему коду, использованию современных криптографических алгоритмов и меньшему потреблению ресурсов. Он быстрее устанавливает соединение, имеет меньшие задержки и эффективнее работает на слабых устройствах

Шаг 6. Произведите настройку технологии Port Knocking для порта OpenVPN

Устанавливаем knockd на сервер



Производим настройку пакетного фильтра
После некоторых неудачных попыток настроить пакетный фильтр, окончательный iptables выглядит следующим образом:

```
┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo iptables -I INPUT -p tcp --dport 1194 -j DROP

┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo iptables -I INPUT -p udp --dport 1194 -j DROP

┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo iptables -L -v -n
Chain INPUT (policy ACCEPT 211K packets, 7624M bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:1194
    0     0 DROP       tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:1194
    0     0 ACCEPT     udp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           udp dpt:51194
    0     0 ACCEPT     all  --  wg0    *       0.0.0.0/0            0.0.0.0/0
   16   960 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:1194
    0     0 ACCEPT     udp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           udp dpt:51820
```

Где первые две строчки блокируют входящие tcp и udp пакеты на порт OpenVpn, в результате чего ни один ip-адрес не сможет установить соединение с OpenVpn сервером

Настраиваем конфиг knockd

```
File  Actions  Edit  View  Help
astepanov@kali: ~/client-configs ×   astepanov@kali: ~/client-configs/files ×   astepanov@
  GNU nano 8.2                                                    /etc/knockd.
[options]
  UseSyslog
  Interface = eth0

[openclosePort]
  sequence = 7000,8000,9000
  seq_timeout = 5

  start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 1194 -j ACCEPT
  cmd_timeout = 60
  stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 1194 -j ACCEPT
  tcpflags = syn
```

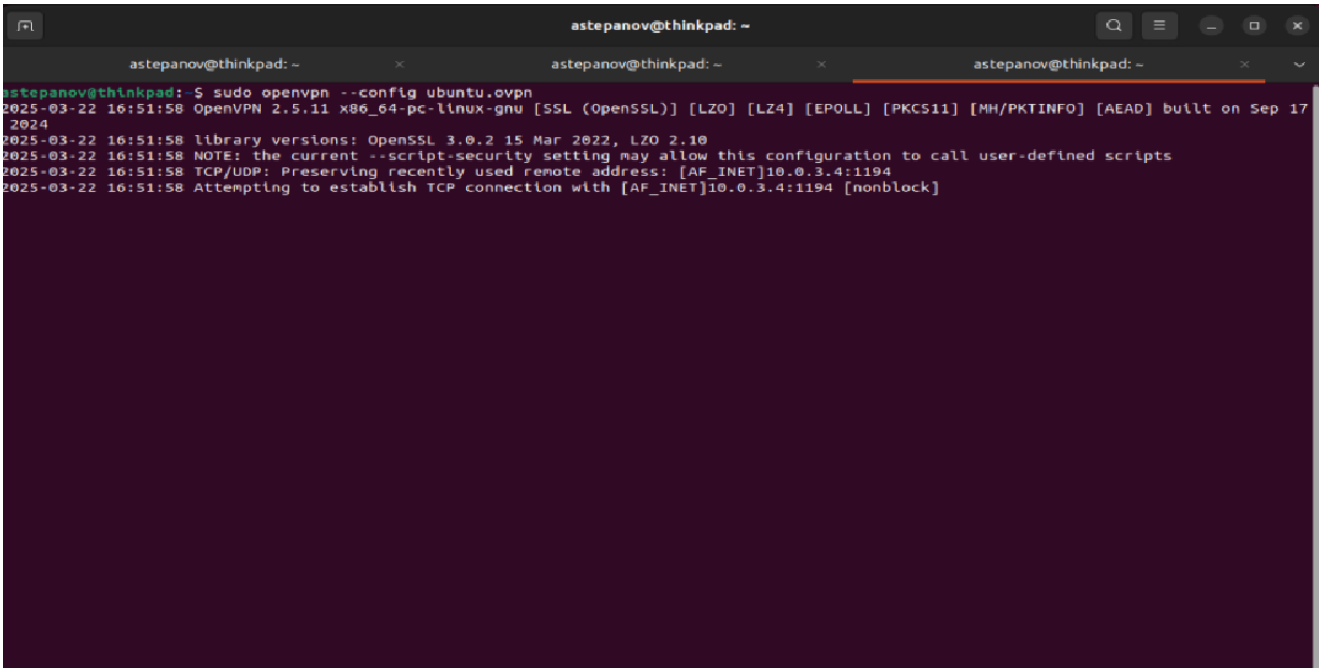
Конфиг открывает порт 1194 для ip-адреса, с которого поступил запрос на tcp-соединение при правильном "простукивании" портов с таймаутом закрытия порта в 60 секунд.

Запускаем службу knockd и проверяем статус

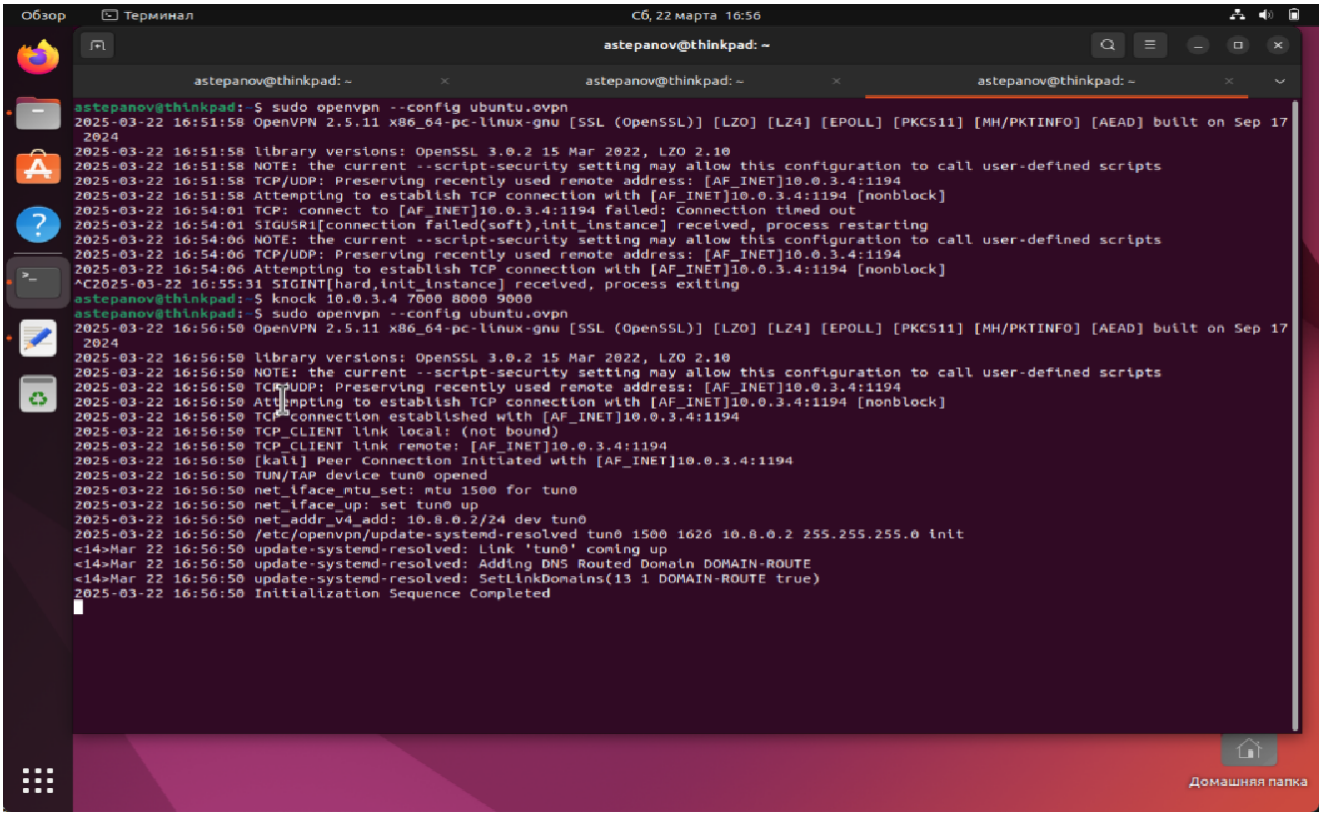
```
┌──(astepanov㉿kali)-[~/client-configs]
└─$ sudo systemctl status knockd.service
● knockd.service - Port-Knock Daemon
     Loaded: loaded (/usr/lib/systemd/system/knockd.service; enabled; preset: disabled)
     Active: active (running) since Sat 2025-03-22 08:40:27 EDT; 4s ago
 Invocation: c10f1c2f31b0441d8ca049a0ffa06721
       Docs: man:knockd(1)
   Main PID: 378213 (knockd)
      Tasks: 1 (limit: 4555)
     Memory: 392K (peak: 1.8M)
        CPU: 79ms
     CGroup: /system.slice/knockd.service
             └─378213 /usr/sbin/knockd

Mar 22 08:40:27 kali systemd[1]: knockd.service: Deactivated successfully.
Mar 22 08:40:27 kali systemd[1]: Stopped knockd.service - Port-Knock Daemon.
Mar 22 08:40:27 kali systemd[1]: Started knockd.service - Port-Knock Daemon.
Mar 22 08:40:27 kali (knockd)[378213]: knockd.service: Referenced but unset environment variable evaluates to an empty string: K
Mar 22 08:40:27 kali knockd[378213]: starting up, listening on eth0
```

Шаг 7. Произведите подключение по OpenVPN с технологией Port Knocking, проверьте работоспособность

Для начала проверим что без простукивания портов VPN-соединение не сможет быть установлено. Для этого пробуем подключиться с VPN-серверу и видим, что статус нашего соединениея "завиает" в состоянии **Attempting to establish TCP connection with [AF_INET]10.0.3.4:1194**
Дальше ничего не происходит, сообщения об успешном подключении к VPN-серверу нету, так как на стороне сервера tcp-пакеты фильтруются.



Дальше "простукиваем" сервер заданной последовательностью **7000 8000 9000" и повторяем попытку подключения к OpenVpn серверу.



Видим, что соединение моментально установлено.

Шаг 8. Просканируйте порт с помощью nmap до отправки последовательности и после, сравните результаты

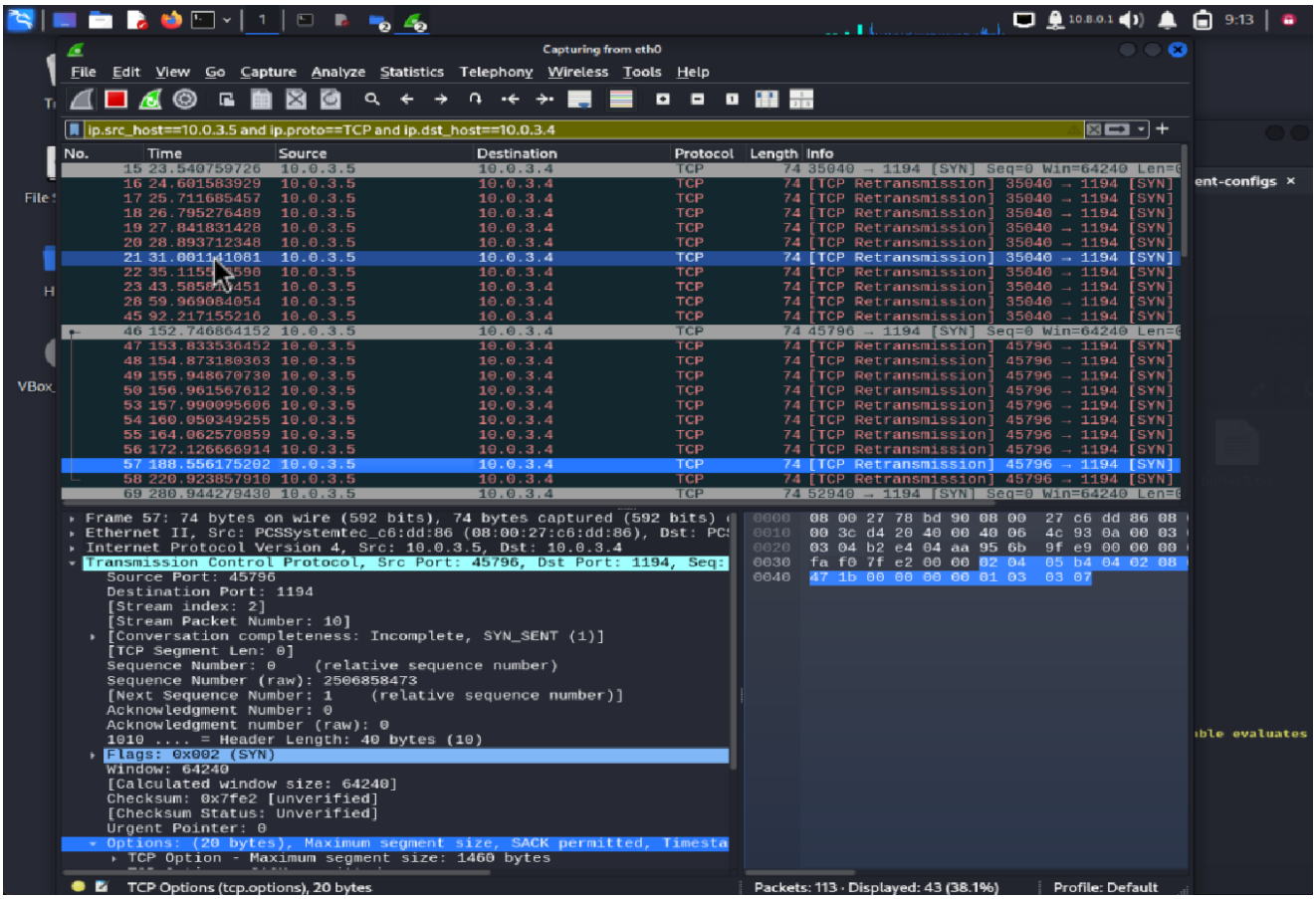
Результаты сканирования порта 1194 с помощью TCP SCAN
alt text

Как видим, до простукивания состояние порта "filtered", что говорит нам о том, что порт блокируется межсетевым фильтром и ответные пакеты от сканируемого хоста не возвращаются обратно.

После простукивания состояние порта "open", что говорит нам о том, что порт открыт и пакеты успешно перемещаются между хостами

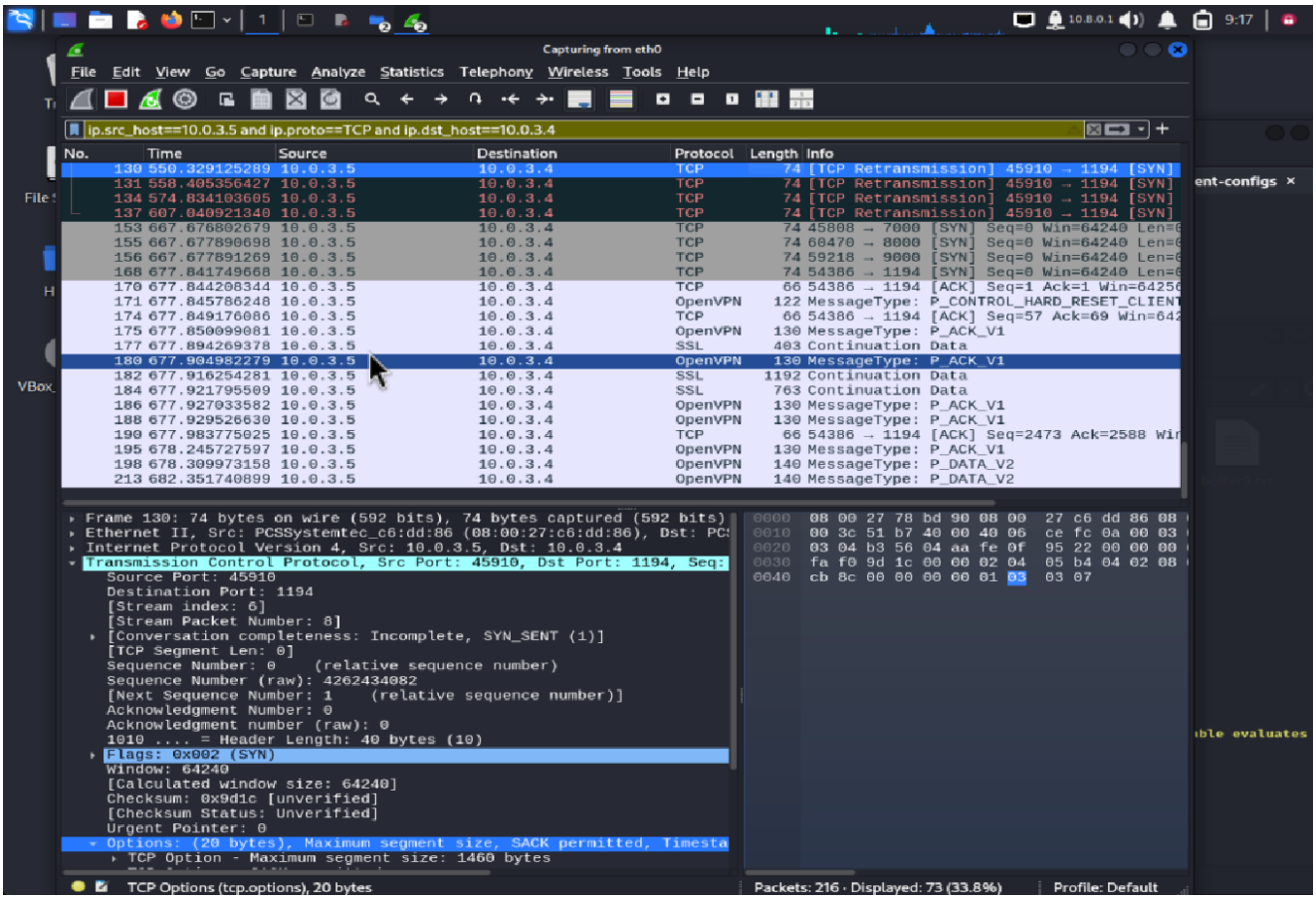
Шаг 9. Посмотрите в Wireshark, как отправляются пакеты при подключении через Port Knocking

До "простукивания" пакеты в сети выглядят следующим образом



1. Клиент отправляет SYN (первый пакет TCP handshake) на сервер.
2. Сервер не отвечает (firewall его просто дропает).
3. Клиент ждёт таймаут и повторяет SYN (TCP Retransmission).
4. Повторные SYN продолжаются несколько раз

А вот как выглядит "простукивание" портов с помощью port-knocking



До установления соединения с OpenVpn сервером идут 3 TCP-пакета на порты 7000, 8000 и 9000 соотвтественно.

После этого мы видим успешное tcp-соединение и дальше пакеты по протоколу OpenVPN передаются между клиентом и сервером