# Первоначальные настройки безопасности VPS: вход по ключу вместо пароля и нестандартный SSH-порт

Первоначальные настройки безопасности VPS: вход по ключу вместо пароля и нестандартный SSH-порт

Ubuntu 24.04 LTS · PuTTY (Windows) · UFW — пошаговая инструкция на реальном примере

# Зачем это нужно

Только что развёрнутая виртуальная машина с белым IP начинает «светиться» в интернете уже в первые минуты после создания: сканеры перебирают SSH-порты по всему диапазону, и как только находят порт 22, тут же начинают подбирать пароли. Показываю на живом примере — свежий сервер Ubuntu 24.04 (RUVDS), которому сутки. Смотрим, что происходит в журнале SSH:

```
journalctl _COMM=sshd | grep -Ei 'invalid|failed|accepted' | tail -30
```

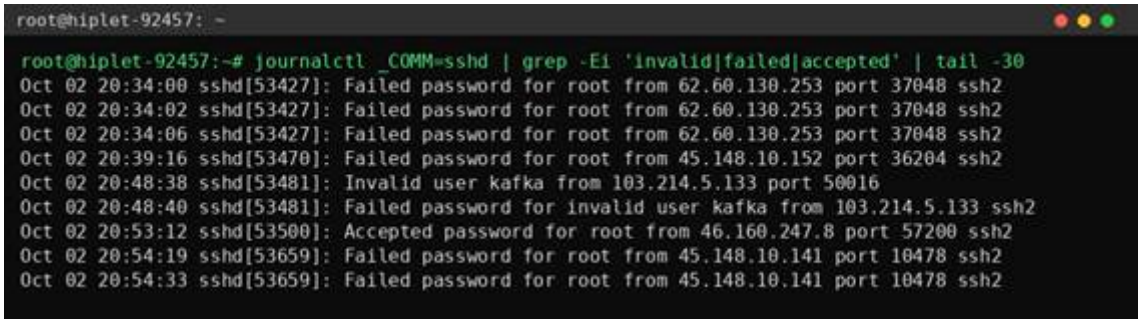


*Рис. 1. Журнал sshd: попытки перебора пароля и несуществующих пользователей идут непрерывно*

Посмотрим, с каких адресов идут переборы и какие имена подбирают:

```
journalctl _COMM=sshd | grep -oE 'from [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' | sort | uniq -c | sort -rn | head -15
```

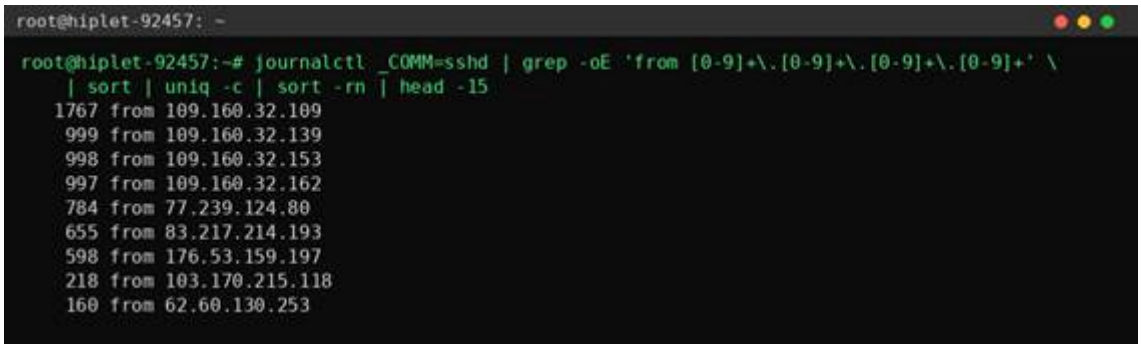


*Рис. 2. Топ адресов-источников перебора: до полутора тысяч попыток с одного IP за сутки*

```
journalctl _COMM=sshd | grep -i invalid | grep -oE 'invalid user \S+' | sort | uniq -c | sort -rn | head -15
```




*Рис. 3. Топ перебираемых имён пользователей: в первую очередь admin, user, deploy*

Вывод простой: сервер с парольным входом по SSH найден и обрабатывается ботами в круглосуточном режиме. Пароли будут подбираться до тех пор, пока не совпадут. Наша цель — сделать перебор бессмысленным: выключить пароли совсем, пустить вход только по ключу и увести SSH с порта 22, чтобы боты не доходили даже до попытки авторизации.

# Шаг 1. Выпускаем ключ на своём компьютере (PuTTYgen)

1. Откройте PuTTYgen (поставляется вместе с PuTTY).

2. В блоке «Type of key to generate» выберите Ed25519 и нажмите Generate, поводив мышью по окну — так собирается случайность для ключа.

3. В поле Key passphrase задайте пароль на сам ключ (по желанию, но лучше — тогда украденный .ppk без этого пароля бесполезен).

4. Нажмите Save private key и сохраните файл, например id_ed25519_new.ppk — это ключ для PuTTY. Храните его только у себя.

5. Скопируйте публичный ключ из верхнего большого поля «Public key for pasting into OpenSSH authorized_keys file» — целиком, одной строкой, начиная с ssh-ed25519.

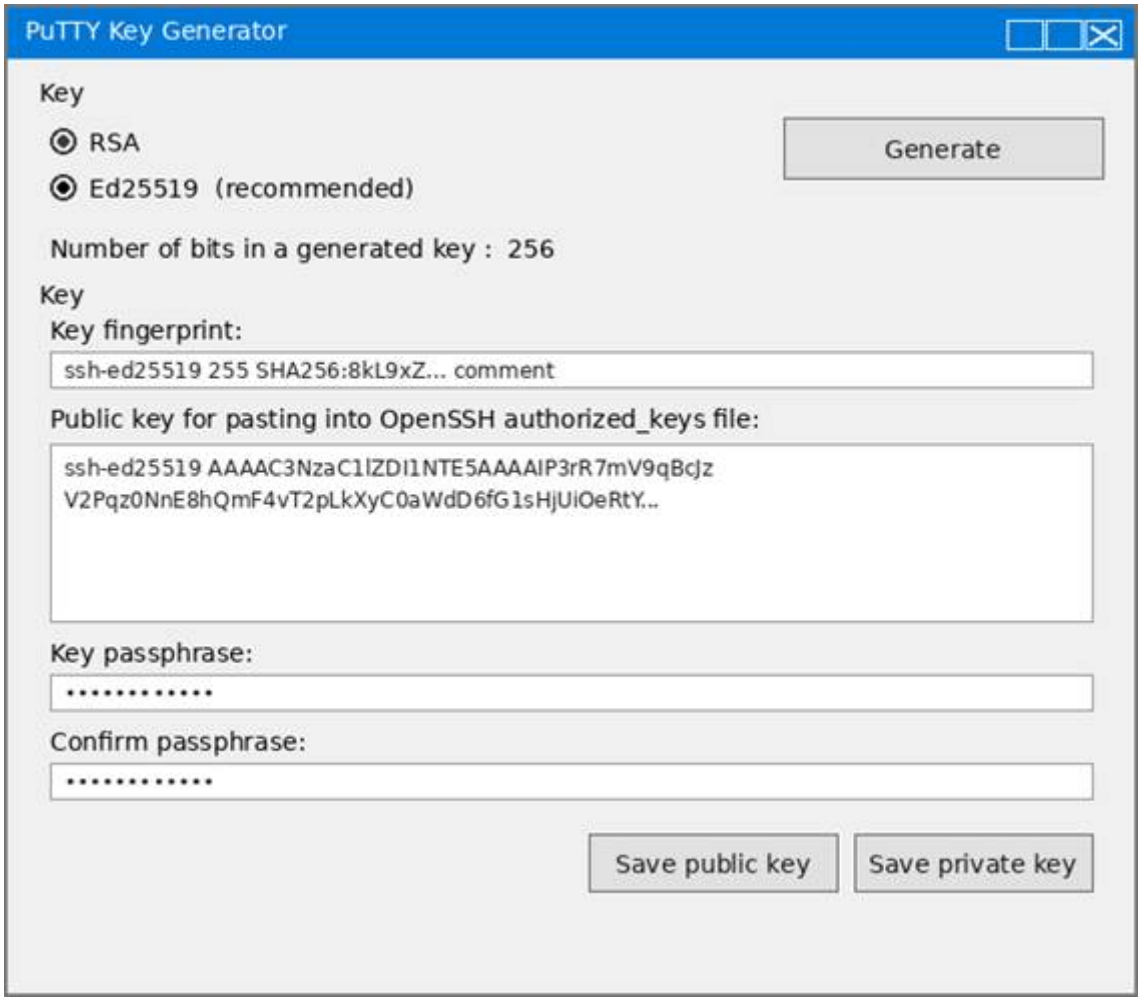


*Рис. 4. Окно PuTTYgen: выбран тип Ed25519, ниже — публичный ключ для вставки на сервер и поля пароля ключа*

*Почему Ed25519, а не RSA: это современный тип ключа — короче, быстрее и без исторических уязвимостей алгоритма ssh-rsa (SHA-1). Кстати, если сервер в логах пишет «signature algorithm ssh-rsa not in PubkeyAcceptedAlgorithms» — это как раз про устаревший RSA-ключ: sshd по умолчанию его больше не принимает.*

# Шаг 2. Добавляем публичный ключ на сервер

Подключитесь к серверу по логину и паролю (последний раз) и выполните команды. Важно: домашняя директория root — это /root, а не /home/root:

```
mkdir -p /root/.ssh
nano /root/.ssh/authorized_keys
```

В nano вставьте скопированный публичный ключ (правая кнопка мыши в PuTTY) и сохраните файл: Ctrl+O, Enter, Ctrl+X. Затем выставьте права:

```
chown -R root:root /root/.ssh
chmod 700 /root/.ssh
chmod 600 /root/.ssh/authorized_keys
```

Проверьте результат:

```
ls -la /root/.ssh/
```

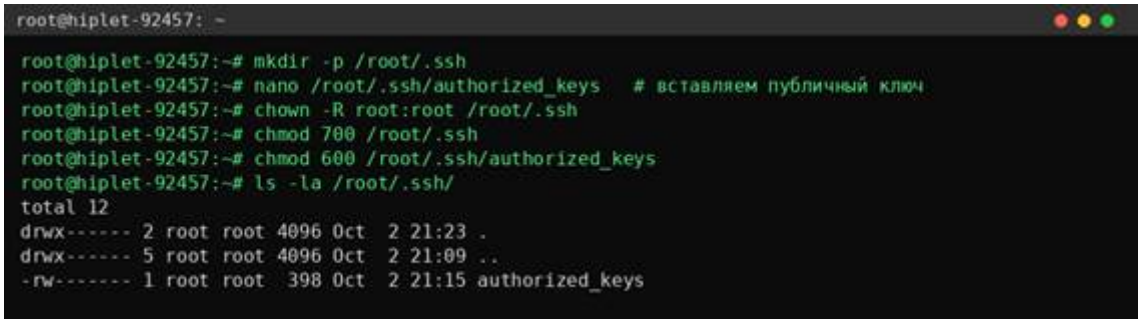


*Рис. 5. Ключ лежит в /root/.ssh/authorized_keys с правами 600, владелец root — как и должно быть*

*Частая ловушка: ключ в файле должен быть ровно одной строкой без переносов. Если вставка в PuTTY разорвала строку — sshd отвергнет ключ. Проверить можно командой cat -A /root/.ssh/authorized_keys: в конце строки должен быть только $ (перевод строки), без ^M (символов Windows-переносов). Если создали лишние директории вроде /home/user1 — их можно удалить: rm -rf /home/user1.*

# Шаг 3. Проверяем вход по ключу в PuTTY

В PuTTY откройте настройки сессии: Connection → SSH → Auth → Credentials → Browse и укажите сохранённый файл ключа id_ed25519_new.ppk. В Session — IP сервера и порт 22 (пока старый). Подключайтесь:

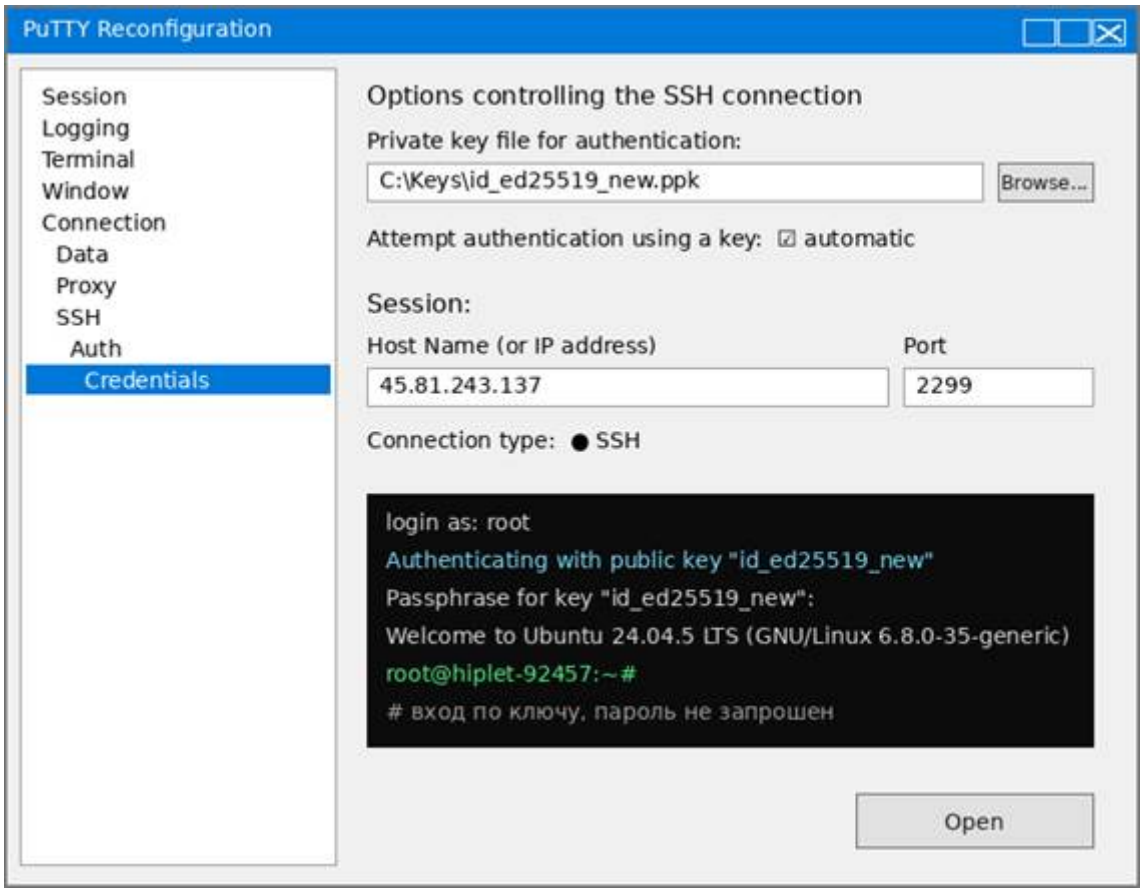


*Рис. 6. Настройки PuTTY: указан файл ключа, порт 2299 (его поменяем позже). Результат успешного входа — по ключу, без запроса пароля*

Если пускает по ключу — полдела сделано. Если видите «Server refused our key», не паникуйте: парольный вход ещё работает, чиним спокойно:

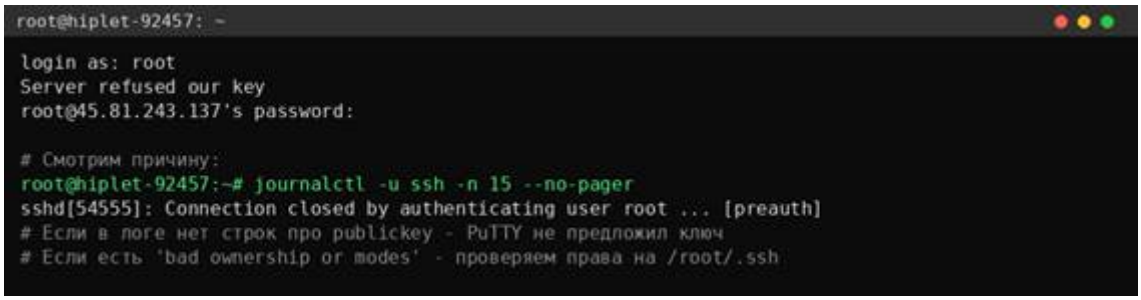


*Рис. 7. «Server refused our key» — типичная ошибка при настройке. Причина всегда ищется в journalctl сразу после неудачной попытки*

```
journalctl -u ssh -n 15 --no-pager
```

По журналу причина определяется однозначно: нет строк про publickey — PuTTY не предложил ключ (проверьте, тот ли .ppk выбран в сессии); «bad ownership or modes» — неверные права на /root/.ssh; разорванная строка в authorized_keys — пересоздайте файл.

*Золотое правило: держите открытой старую рабочую сессию, пока не убедитесь, что новая пускает по ключу. Все изменения дальше делаем из этой сессии.*

# Шаг 4. Отключаем парольный вход

Сначала смотрим, что уже настроено в системе — Ubuntu 24.04 с cloud-init приходит с готовыми файлами конфигурации:

```
ls /etc/ssh/sshd_config.d/
grep -rn 'PasswordAuthentication' /etc/ssh/sshd_config.d/ /etc/ssh/sshd_config
```

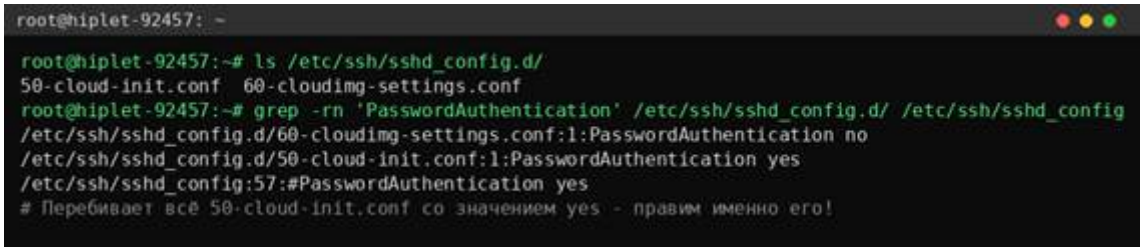


*Рис. 8. Аудит конфигурации: 60-cloudimg-settings.conf разрешает no, а 50-cloud-init.conf — yes. Именно он перебивает всё остальное*

*Важный момент, который часто понимают неправильно. В sshd действует правило «первое полученное значение выигрывает»: файлы из /etc/ssh/sshd_config.d/ подключаются директивой Include в самом начале основного конфига, и внутри каталога они читаются по алфавиту. Поэтому 50-cloud-init.conf (читается раньше) сильнее, чем 99-hardening.conf или строка в конце sshd_config — значения «no» в них просто не дойдут до sshd. Отключая пароли, правьте самый ранний файл или сам 50-cloud-init.conf.*

Правим именно 50-cloud-init.conf (или создаём свой файл с номером меньше 50, например 10-hardening.conf):

```
nano /etc/ssh/sshd_config.d/50-cloud-init.conf

# Содержимое:
PasswordAuthentication no
KbdInteractiveAuthentication no
PubkeyAuthentication yes
PermitRootLogin prohibit-password
```

Чтобы cloud-init не вернул пароли после перезагрузки, зафиксируем его настройку:

```
echo 'ssh_pwauth: false' > /etc/cloud/cloud.cfg.d/50-cloud-init-override.cfg
grep -rn 'ssh_pwauth' /etc/cloud/ 2>/dev/null
```

Проверяем итоговую конфигурацию и перезапускаем SSH:

```
sshd -t && echo OK
sshd -T | grep -Ei 'passwordauthentication|kbdinteractiveauthentication|permitrootlogin'
```




*Рис. 9. Контрольная проверка: passwordauthentication no, permitrootlogin without-password (это и есть «только по ключу»), затем перезапуск службы*

После перезапуска откройте новую PuTTY-сессию: вход должен пройти по ключу без запроса пароля. Только после этого шаг можно считать завершённым.

# Шаг 5. Меняем стандартный порт 22

Особенность Ubuntu 24.04: sshd работает через systemd socket activation. Значит, менять порт нужно не только (и не столько) в sshd_config, а в сокете, который и слушает входящие соединения. Выберите порт в диапазоне 1024–65535, не совпадающий с известными службами. Для примера возьмём 2299.

```
systemctl edit ssh.socket
```

В открывшемся редакторе вставьте ровно этот блок (первая пустая ListenStream очищает старое значение — порт 22):

```
[Socket]
ListenStream=
ListenStream=0.0.0.0:2299
ListenStream=[::]:2299
```

Применяем изменения и проверяем, что sshd слушает новый порт:

```
systemctl daemon-reload
systemctl restart ssh.socket
ss -tnlp | grep 2299
```

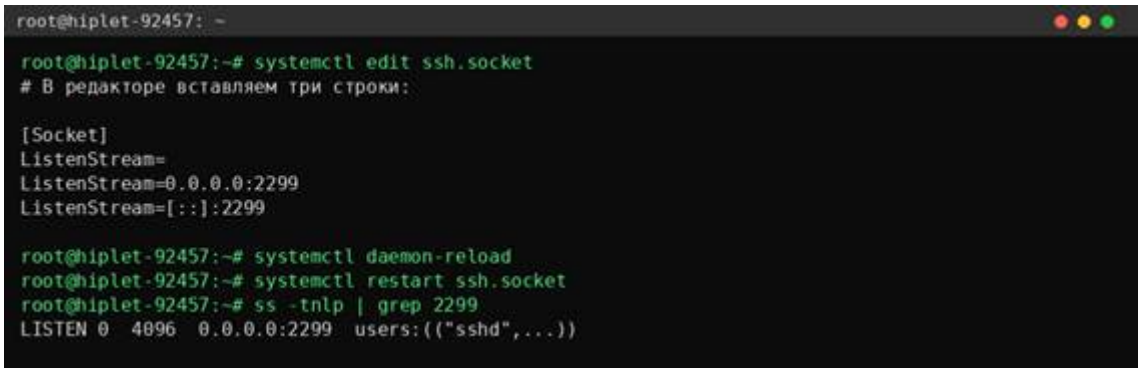


*Рис. 10. Override для ssh.socket и проверка: sshd слушает 2299, IPv4 и IPv6*

*Меняйте порт только после того, как вход по ключу работает, а пароли отключены — иначе есть риск запереть самого себя. Отладка всегда ведётся в одной открытой сессии с проверкой новой.*

# Шаг 6. Включаем фаервол UFW

Сначала разрешаем новый порт SSH (это критично: без правила при включении ufw вы потеряете доступ), затем запрещаем старый порт 22 и включаем фаервол:

```
apt update && apt install -y ufw
ufw allow 2299/tcp
ufw deny 22/tcp
ufw --force enable
ufw status verbose
```

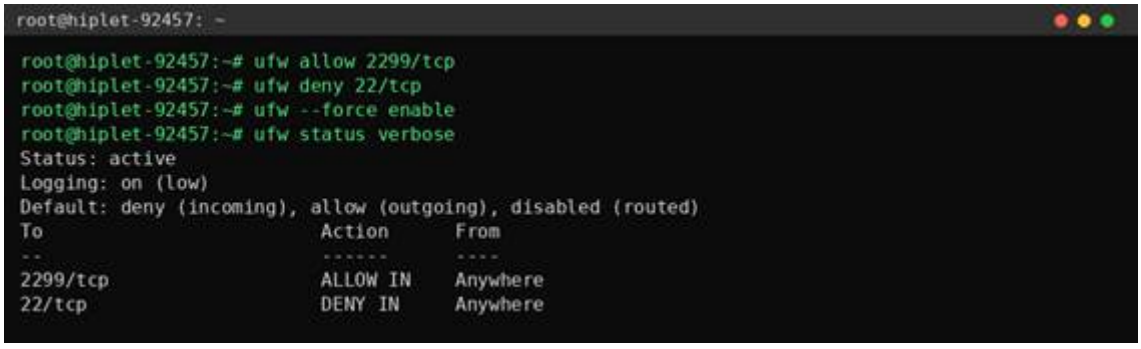


*Рис. 11. Статус UFW: активен, 2299 разрешён, 22 запрещён*

Контрольная проверка: ещё раз подключитесь из нового окна PuTTY по новому порту и ключу. Если работает — можно закрывать старую сессию.

# Шаг 7. Проверяем тишину

Через сутки после всех настроек глянем на итог. Команда для оценки:

```
journalctl _COMM=sshd --since yesterday | grep -c 'Failed password'
```

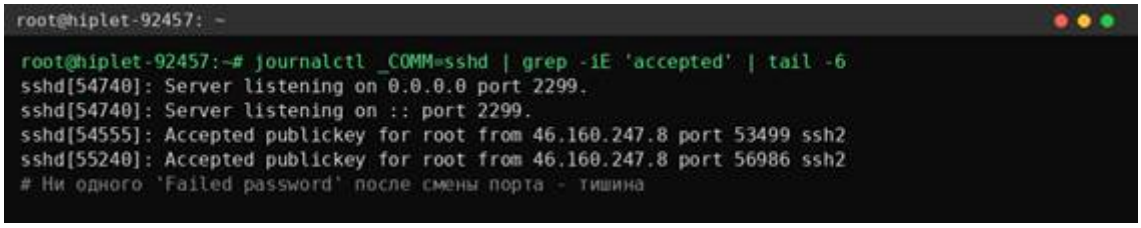


*Рис. 12. Результат: после смены порта — только успешные входы по ключу с одного адреса, ни одного подбора пароля*

Что говорить о ботах: они не исчезли, просто перестали доходить до sshd. Порт 22 отбивается фаерволом ещё на уровне сети — в kernel-журнале это видно как UFW BLOCK. Если захотите погасить и этот шум, выполните ufw logging off — на безопасность это не влияет, просто меньше строк в журнале.

# Итоговый чек-лист

- Вход по SSH — только по ключу Ed25519, пароли отключены (passwordauthentication no).
- PermitRootLogin prohibit-password — root может войти только по ключу, или no, если root-вход не нужен.
- SSH слушает нестандартный порт (socket override + при необходимости Port в sshd_config).
- UFW активен: новый порт разрешён, 22 запрещён.
- cloud-init зафиксирован (ssh_pwauth: false), настройки переживут перезагрузку.
- Файл .ppk с ключом хранится на компьютере, публичный ключ — на сервере в /root/.ssh/authorized_keys с правами 600.

И последняя проверка, которую стоит сделать после всего: перезагрузить сервер (reboot) и убедиться, что после старта всё поднялось само — SSH на новом порту, UFW активен, вход по ключу работает. Это лучшая проверка того, что конфигурация переживёт будущее обслуживание.

# Типичные ошибки

6. Создание ключа в /home/root/.ssh — у root домашняя директория /root, sshd никогда не прочитает ключ из /home/root.

7. Отключение паролей до проверки входа по ключу — потеря доступа; спасает только VNC-консоль провайдера.

8. Надежда на 99-hardening.conf при активном 50-cloud-init.conf — в sshd выигрывает первое значение, поэтому правьте самый ранний файл.

9. Смена порта правкой только sshd_config на Ubuntu 24.04 — порт не сменится без override для ssh.socket.

10. Включение ufw без правила для нового SSH-порта — мгновенная потеря доступа.

11. Разорванная при вставке строка ключа в authorized_keys — всегда проверяйте cat -A.

# Что дальше

Следующие рубежи, если захотите усилить защиту: fail2ban (автобан IP за перебор — полезен, если порты всё же открыты), отключение логов UFW для снижения шума, а также регулярное обновление системы: apt update && apt upgrade — на баннере свежей Ubuntu 24.04 обычно уже висит напоминание о доступных обновлениях.

---

Revision #2
Created 2026-10-02 20:50:33 UTC by im
Updated 2026-10-02 22:04:05 UTC by im