1. Установить SSH-сервер и настроить удалённое подключение по ключам, вместо пароля.

Список команд, используемых для установки и конфигурирования сервера:

sudo apt install openssh-server –y

sudo systemctl enable --now ssh

проверяем, что openssh сервер запустился

sudo systemctl status ssh

Выводим список всех правил iptables

sudo iptables –L

Если нет правила для порта SSH – то вводим команду, чтобы разрешить подключение

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

Подготавливаем папку и файл для сертификата openssh для текущего пользователя

mkdir ~/.ssh

chmod 0700 ~/.ssh

touch ~/.ssh/authorized\_keys

chmod 0644 ~/.ssh/authorized\_keys

С помощью ssh-keygen или puttygen генерируем сертификаты.

В файл ~/.ssh/authorized\_keys копируем публичный сертификат (в формате openssh)

Проверяем в файле /etc/ssh/sshd\_config параметр PubkeyAuthentication, нужно чтобы имел значение yes

sudo nano /etc/ssh/sshd\_config

Сервер настроен, проверяем подключение

Успешное подключение к серверу с использование public\_key. Сообщение можно посмотреть в логах /var/log/auth.log



1. Создать нового пользователя с домашней директорией и выдать ему возможность запускать следующие утилиты без требования пароля:

- /sbin/route, /sbin/iptables, /usr/bin/nmap, /usr/sbin/hping3

- usr/bin/systemctl

- sbin/ifup, /sbin/ifdown

Создаем пользователя skilluser командой:

sudo adduser skilluser

Добавляем в файл /etc/sudoers строки, позволяющие запускать утилиты без требования пароля:

skilluser ALL=(ALL) NOPASSWD: /sbin/route, /sbin/iptables, /usr/bin/nmap, /usr/sbin/hping3

skilluser ALL=(ALL) NOPASSWD: /usr/bin/systemctl

skilluser ALL=(ALL) NOPASSWD: /sbin/ifup, /sbin/ifdown

sudo nano /etc/sudoers

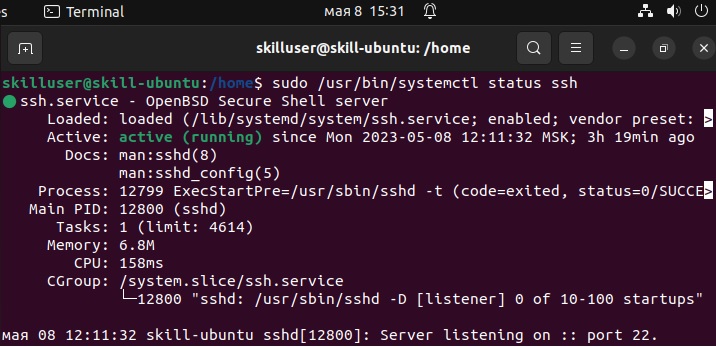
Для проверки заходим под пользователем skilluser

su skilluser

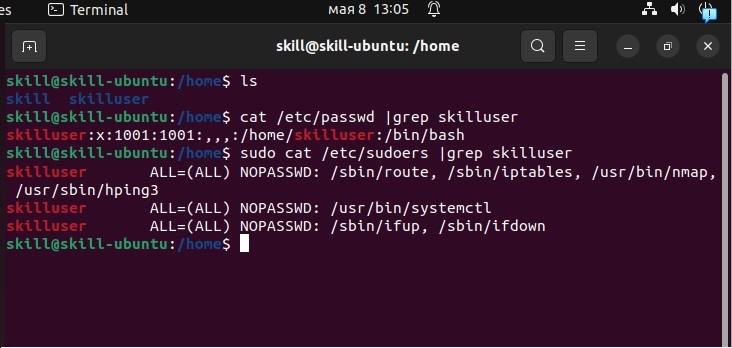
Запускаем утилиту /usr/bin/systemctl для проверки статуса SSH сервера.

sudo /usr/bin/systemctl status ssh

Как видно из скриншота, утилита запустилась без ввода пароля суперпользователя



Результат выполнения команды добавления пользователя и содержимое файла /etc/sudoers на скриншоте ниже:

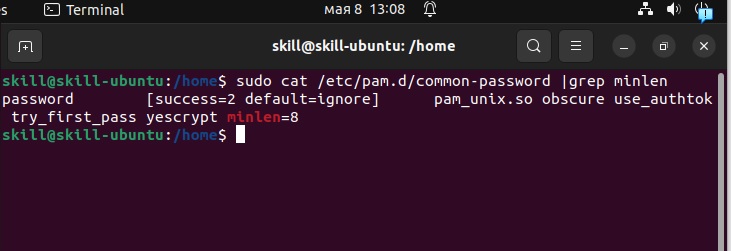


1. Установить минимальную длину пароля для пользователя в 8 символов.

Использовалась статья: https://sysadmin78.ru/doku.php/how\_to:how\_to\_set\_password\_policies\_in\_linux

Для добавления опции минимальной длины пароля для пользователей, нужно отредактировать файл /etc/pam.d/common-password

sudo nano /etc/pam.d/common-password



1. Установить на сервер пакеты Java.

sudo apt update

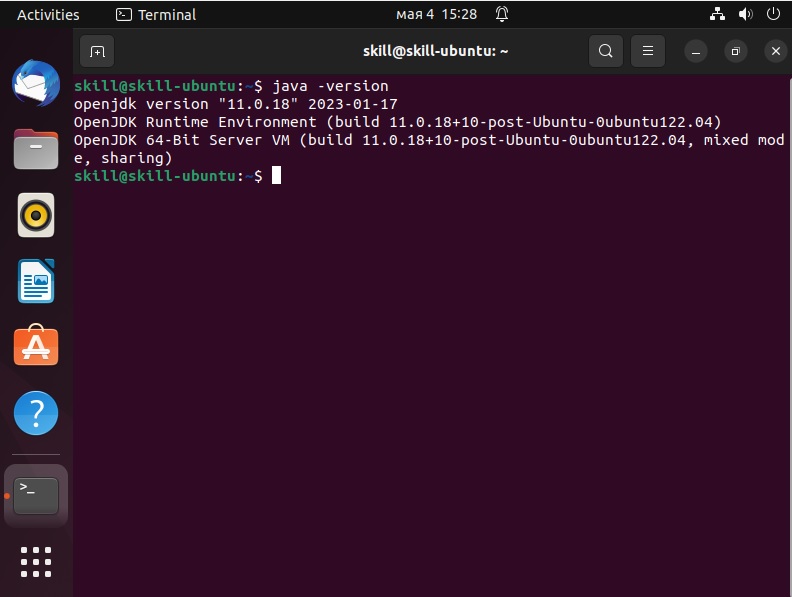
Устанавливаем пакеты JRE

sudo apt install default-jre –y

проверяем успешность установки пакетов java

java -version

использовался следующий источник: <https://ubuntu.com/tutorials/install-jre#2-installing-openjdk-jre>



1. Настроить автоматическое сканирование антивирусом всей ОС каждый понедельник в 4 утра. При этом раз в месяц должно происходить обновление базы данных антивирусов.

Для автоматического сканирования антивирусом clamav всей ОС каждый понедельник в 4 утра создаем задачу в CRON:

0 4 \* \* 1 clamscan --recursive --infected / --move=/tmp/clamscan --log=/var/log/clamscan.log

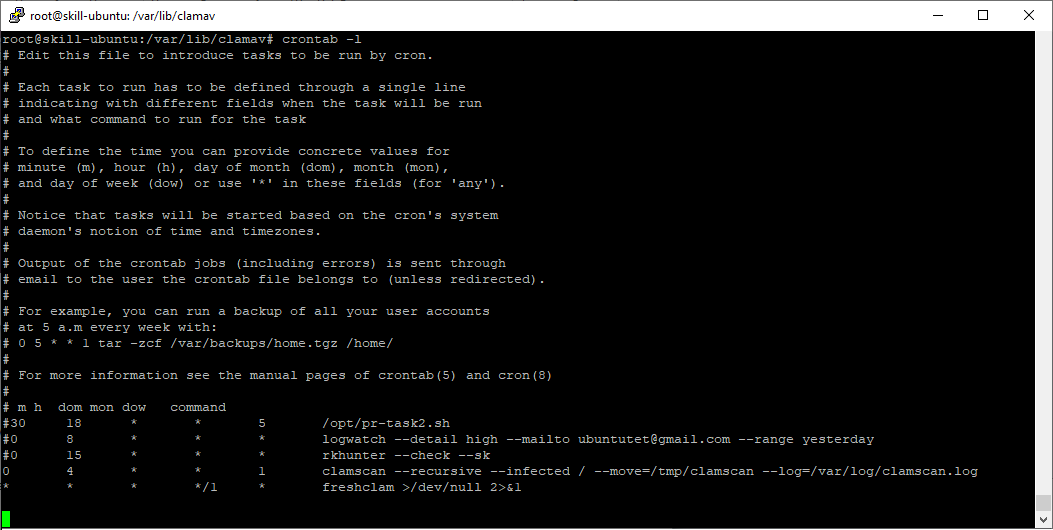
Антивирус clamav уже имеет задачу в CRON на ежедневное обновление. Чтобы увеличить интервал до 1 месяца нужно отключить автоматическое ежедневное обновление и добавить новую задачу в CRON:

sudo stop clamav-freshclam

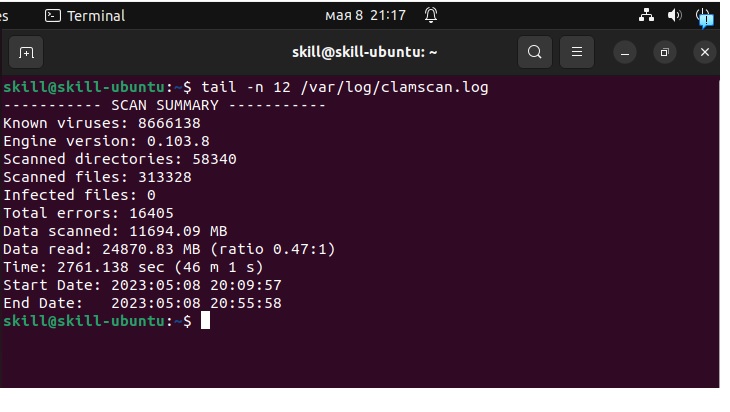
sudo update-rc.d clamav-freshclam disable

crontab –e

\* \* \* \*/1 \* freshclam >/dev/null 2>&1

Получаем следующий график в планировщике CRON

Результат сканирования командой clamscan --recursive --infected / --move=/tmp/clamscan --log=/var/log/clamscan.log



1. «Настроить firewall на блокирование всего входящего и выходящего трафика.»

Для блокировки всего входящего и исходящего трафика вводим команды:

iptables -P INPUT DROP

iptables -P OUTPUT DROP

Получаем следующую таблицу iptables. Исходящий трафик проверяем пингом.

