**Содержание**

[НАСТРОЙКА СЕТЕВОЙ СВЯЗНОСТИ 3](#_Toc169086199)

[НАСТРОЙКА МЕХАНИЗМОВ ЦЕНТРАЛИЗОВАННОГО АДМИНИСТРИРОВАНИЯ 22](#_Toc169086200)

[НАСТРОЙКА ВЕБ-СЛУЖБ И СИСТЕМ ХРАНЕНИЯ ИНФОРМАЦИИ 38](#_Toc169086201)

[НАСТРОЙКА СЕТЕВЫХ И ОПЕРАЦИОННЫХ ОГРАНИЧЕНИЙ 67](#_Toc169086202)

[НАСТРОЙКА РАБОЧИХ МЕСТ ПО РЕКОМЕНДАЦИЯМ ФСТЭК 71](#_Toc169086203)

[КОМПРОМЕНТИРУЮЩИЕ ДАННЫЕ И СЕРВИСЫ 71](#_Toc169086204)

[АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ ПРОИЗВОДИТЕЛЯ 76](#_Toc169086205)

**Защита корпоративной ИТ-инфраструктуры.**

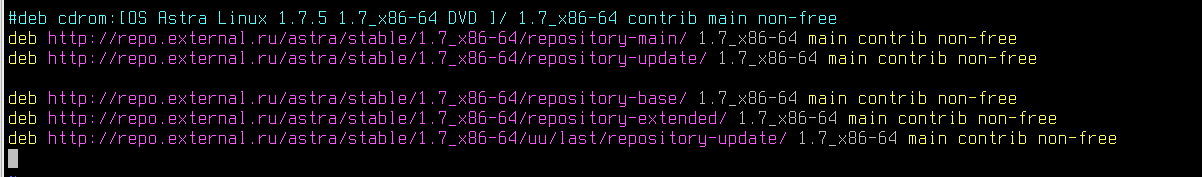
Доступ до виртуальных машин на базе Debian 12 – root:toor

Доступ до виртуальных машин на базе Astra Linux – astra:P@ssw0rd

DNS сервер – 77.88.8.1.

Репозиторий – repo.external.ru

Подключить репозиторий нужно так



## НАСТРОЙКА СЕТЕВОЙ СВЯЗНОСТИ

1. На всех устройствах создаём и настраиваем L3-интерфейсы.

**Что нужно делать?**

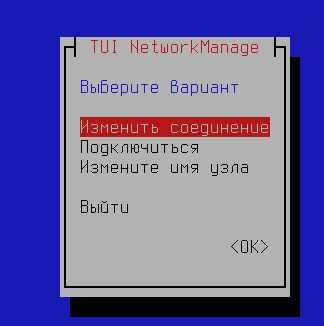
Необходимо на все устройства установить IP-адреса.

**Как это делать?**

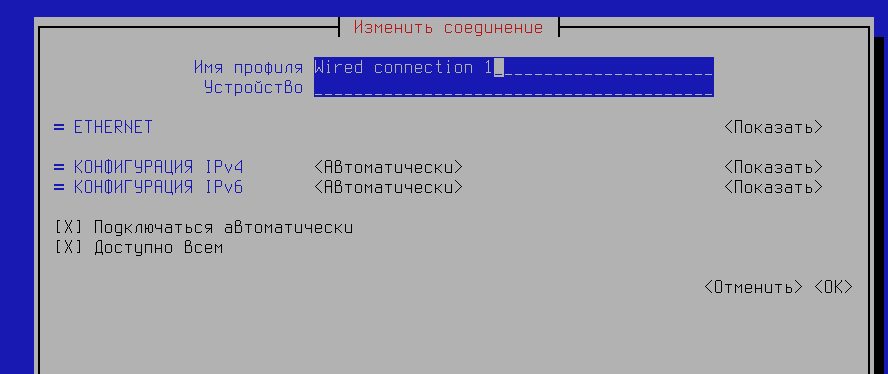
На Astra Linux –

Если есть графический интерфейс (например, на компьютерах клиентов), или NetworkManager. Воспользоваться можно утилитой **nmtui**

Введите её в консоли, откроется псевдографический интерфейс.



Выберем «Изменить соединение» 🡪 «Wired Connection 1»

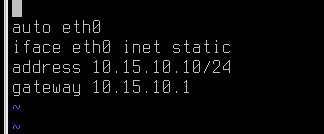


И здесь наблюдаем отсутствие каких-либо настроек.

**Это следствие того, что на компьютере изначально был выбран другой инструмент настройки.**

**Это важный момент, на компьютерах под управлением Linux (любой дистрибутив) нельзя использовать несколько инструментов работы с сетью – выбирайте только один, nmtui, /etc/network/interfaces или что-то другое.**

**Конкретно на этих хостах везде - /etc/network/interfaces основной инструмент управления сетью. Файл выглядит вот так**

****

Конструкция файла:

auto eth0 – укажет, что интерфейс будет добавлен в автозагрузку

iface eth0 inet static – указывает, что интерфейс настроен статическим IP-адресом

address 10.15.10.10/24 – указываем адрес и маску подсети

gateway 10.15.10.1 – указываем шлюз.

После настройки интерфейса, перезагрузите службу networking –

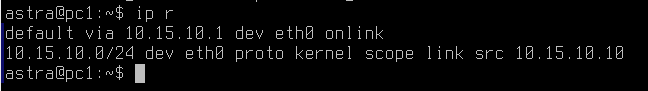
sudo systemctl restart networking

**Как проверить?**

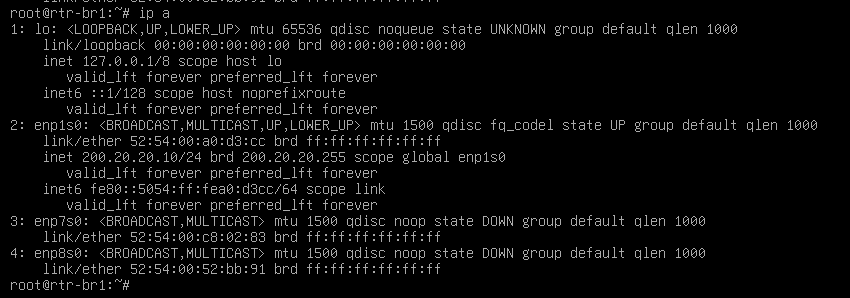
**Введите команду – ip a. Это покажет настроенный IP-адрес.**

****

**Введите команду – ip r – чтобы убедиться в настроенном шлюзе по умолчанию**

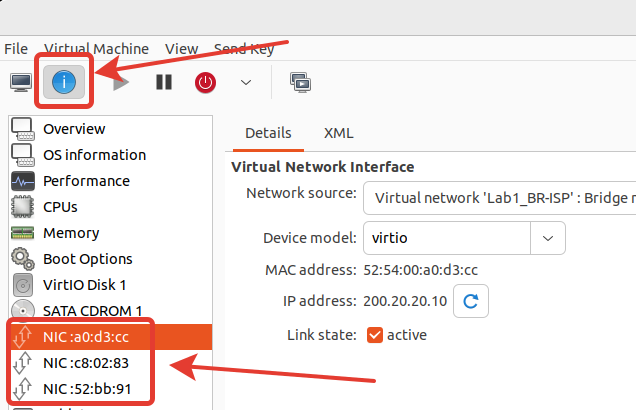
****

А что делать, если интерфейсов несколько?



И как понять какой интерфейс подключен в какой порт?

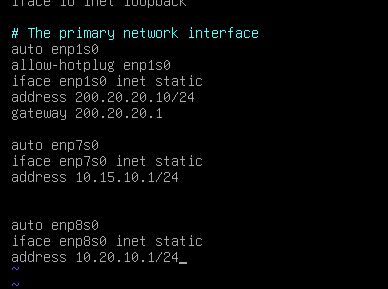
Перейдите в «Информацию»



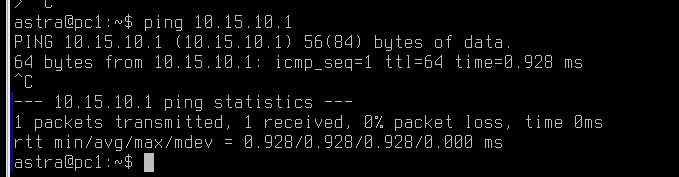
На ROUTER1, например, интерфейсов аж 3. Один из них уже настроен в сторону провайдера ISP.

В информации видно интерфейсы и их мак-адреса, на основе этой информации вы сможете легко понять куда и как нужно подключать и настраивать сетевые реквизиты.

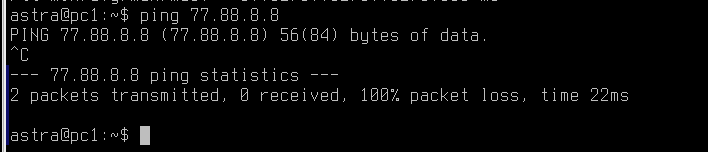
В случае ROUTER1 настройка была бы такой -



Проверим, что с PC1 пинг до ROUTER1



Отлично! Тогда и в интернет выход есть?



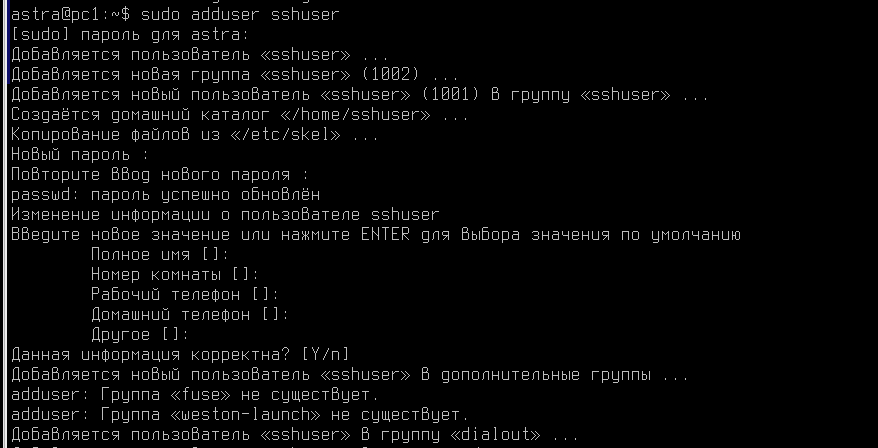
Но нет! А не работает, потому что на ROUTER1 не настроен [NAT](https://habr.com/ru/articles/583172/)! И у него есть разные вариации, об этом подробнее в п.6.

По аналогии настройте mIP-адреса на всех устройствах в сети.

1. Все устройства должны быть доступны по протоколу SSH.

Как делать?

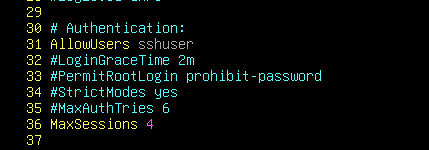
Создать пользователя можно командой – **sudo adduser sshuser**

****

Данной командой вы создаете пользователя, который будет автоматически добавлен во все стандартные группы пользователей, получит домашний каталог и пароль. Данной командой удобно пользоваться если нужно сделать пользователя быстро.

Далее необходимо настроить файл - **/etc/ssh/sshd\_config –** если такого файла нет, значит нужно установить пакет – **openssh-server (на PC1 его установить пока что не получится, ведь мы еще не дошли до настройки интернета)**

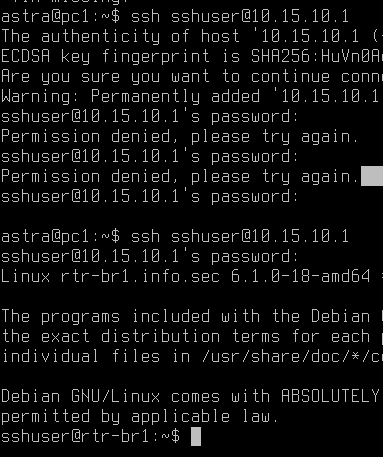
Для реализации пунктов заданий сессий и пользователей –



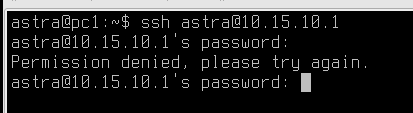
После настройки не забудьте – systemctl restart sshd

Как проверить?

Попробуйте подключиться под sshuser на ROUTER1 с PC1



При этом под другими пользователями вход должен быть запрещен.



Лимит сессий можно будет проверить далее, когда мы сможем подключиться с нескольких разных ПК.

1. Между ROUTER2 и ROUTER1 настроим удаленный защищенный туннель.

Как делать?

Начнем с того, какие варианты вообще можно было бы сделать – настроить IPSEC + GRE, настроить WireGuard, настроить OpenConnect и много другое. Рассмотрим два варианта – через IPSEC + GRE или через WireGuard.

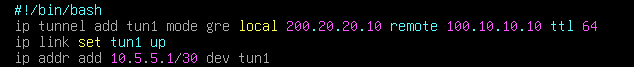
Начнем с IPSEC + GRE.

1) Установим IPSEC – **apt install strongswan**

**GRE туннель можно собрать несколькими инструментами, в этом решебнике разберем формат создания GRE туннеля через скрипт.**

**Создайте файл - /etc/gre.up**

**Со следующим содержимым, делаем на ROUTER1**

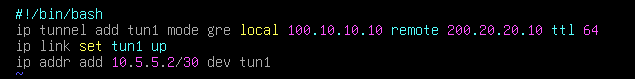
****

Данный скрипт выполняет создание скрипта, первой строкой мы описываем процесс создания туннеля, с указанием внешних адресов наших роутеров.

Второй строкой – включаем туннель

Третьей строкой – настраиваем адрес на интерфейсе. Выдайте ему права на выполнение – **chmod +x /etc/gre.up.**

**На ROUTER2 файл /etc/gre.up будет таким –**

****

Запустить скрипт можно просто указанием пути до него - /etc/gre.up

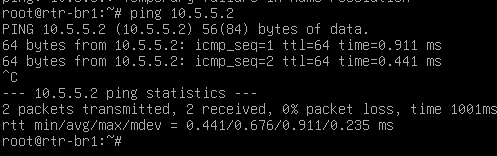
**Как проверить создание туннеля?**

Команда ip a покажет перечень настроенных интерфейсов





А также проверить пинг по туннельному интерфейсу



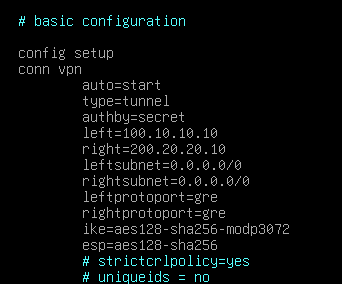
Подключение есть! Туннель GRE сформирован, но GRE – протокол небезопасный, он решает только задачу инкапсуляции и организует удаленное соединение (как VPN). Безопасным данный протокол сделает IPSEC.

Перейдем в конфигурационный файл - **/etc/ipsec.conf**

**На роутере ROUTER1 файл /etc/ipsec.conf выглядит вот так**

****

**На роутере ROUTER2 файл /etc/ipsec.conf выглядит вот так**

****

**conn vpn -- создание соединения с именем vpn**

**auto=start -- запускать соединение автоматически при старте демона ipsec.**

**type=tunnel -- указывает ipsec работать в туннельном режиме. Туннельный**

**шифрует изначальный IP-пакет полностью и добавляет новый заголовок IP. Транспортный**

**же шифрует всё, что выше уровня IP, а заголовок IP оставляет без изменений.**

**Грубо говоря, туннельный режим вы используете для того, чтобы связать две приватные сети через публичную, обеспечив при этом шифрование (Что-то вроде безопасного GRE). Транспортный же актуален тогда, когда IP-связность уже достигнута,**

**но трафик между узлами нужно шифровать.**

**authby=secret -- указывает ipsec аутентифицироваться по ключу из файла**

**/etc/ipsec.secrets**

**left -- указывает локальный адрес (откуда подключаемся)**

**right -- указывает удаленный адрес (куда подключаемся)**

**leftsubnet -- локальные подсети, трафик из которых необходимо шифровать**

**rightsubnet -- удаленные подсети, трафик к которым необходимо шифровать**

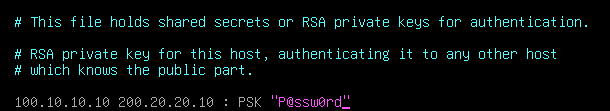
**leftprotoport -- локальный транспортный протокол для шифрования**

**rightprotoport -- удаленный транспортный протокол для шифрования**

**ike -- параметры первой фазы IPSEC**

**esp -- параметры второй фазы IPSEC.**

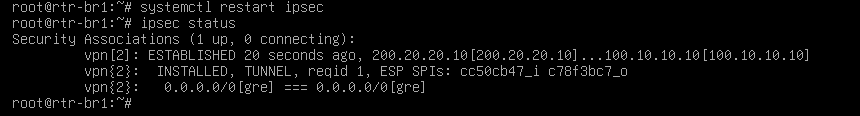
**Файл /etc/ipsec.secrets на обоих хостах одинаковый**

****

Теперь выполняем **– systemctl restart ipsec**

**Как проверить?**

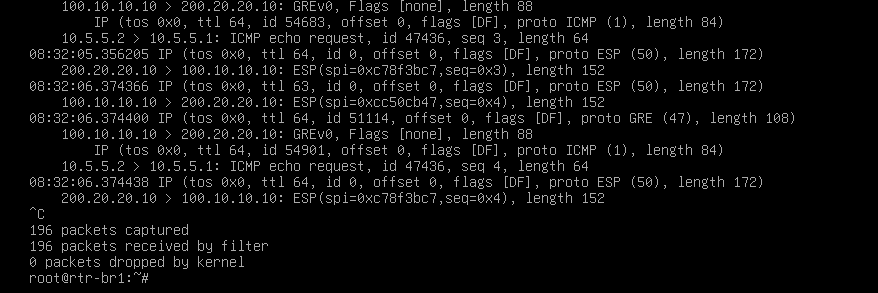
Первым делом на роутерах введите команду – **ipsec status**

****

Если вывод как на скриншоте выше – все работает

Но как это проверить по-настоящему? Берем tcpdump – консольный сниффер трафика.

Команда – **tcpdump -i enp1s0 -vvvvvvvvvvvv**



Она запустит процесс прослушивания интерфейса. В этот момент на ROUTER2 запустите пинг в ROUTER1. Если в выводе tcpdump вы увидите множество ESP сообщений – все сделано прекрасно и трафик точно шифруется.

На чемпионатах часто проверяют такую настройку через машину ISP. Чтобы точно убедиться в отсутствии MITM атак.

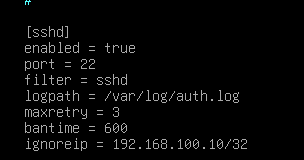
1. С помощью инструмента fail2ban ограничьте доступ по SSH. Используйте следующее правило:
   * Доступ к роутерам разрешен только с доменного контроллера.

**Как делать?**

**Fail2Ban** – программа для защиты серверов от атак методом грубой силы

Для начала установите fail2ban – **apt install fail2ban**

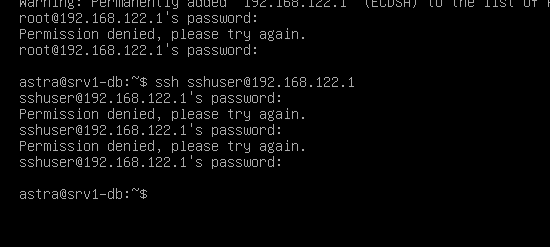
**На ROUTER1 и ROUTER2 настроить файл - /etc/fail2ban/jail.conf**

****

После этого **– systemctl restart fail2ban**

**Как проверить?**

Попробуйте подключиться до ROUTER1 или ROUTER2 с любого компьютера.



Выходит ошибка – это правильно!

Проверяем с доменного контроллера –



Все работает – как и планировалось по заданию!

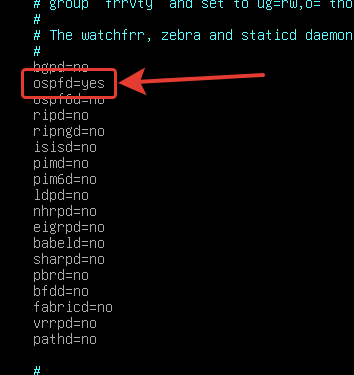
1. Для обеспечения сетевого взаимодействия настройте протокол динамической маршрутизации OSPF на роутерах
   * Обеспечьте работу hello-пакетов только на туннельном интерфейсе.

**Как делать?**

В рамках настроенного ранее туннеля настроим маршрутизацию OSPF. Используем для этого пакет FRR.

Установите его на обоих роутерах – **apt install frr**

Затем перейдите в конфигурационный файл - **/etc/frr/daemons**

**|** ****

Это потребуется для того, чтобы включить протокол OSPF.

После конфигурации перезагрузите службу FRR – **systemctl restart frr**

Далее вводим команды:

**vtysh -** для входа в систему FRR

**conf t –** зайдем в терминал

**router ospf -**  переходим в режим OSPF

**network 10.5.5.0/30 area 0 –** указываем подсети, которые планируем опубликовать. А опубликовать надо – подсеть в туннеле, и все подсети офиса. Кроме подсети в ИНТЕРНЕТ!

**network 10.15.10.0/24 area 0**

**network 10.20.10.0/24 area 0**

**passive-interface default**

**int tun1**

**no ip ospf passive**

**do wr**

****

**Таким образом мы выполнил настройку OSPF на ROUTER1**

Повторим настройку на ROUTER2

Команды:

**vtysh**

**conf t**

**router ospf**

**network 10.5.5.0/30 area 0**

**network 192.168.100.0/24 area 0**

**network 10.200.100.0/24 area 0**

**network 172.16.100.0/24 area 0**

**passive-interface default**

**int tun1**

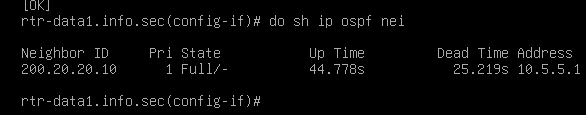
**no ip osp passive**

**do wr**

****

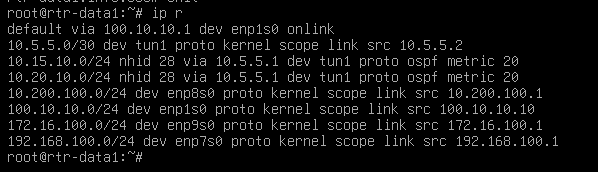
**Как проверить?**

**Команда – do sh ip ospf nei покажет соседей в протоколе OSPF**

****

А команда – ip r

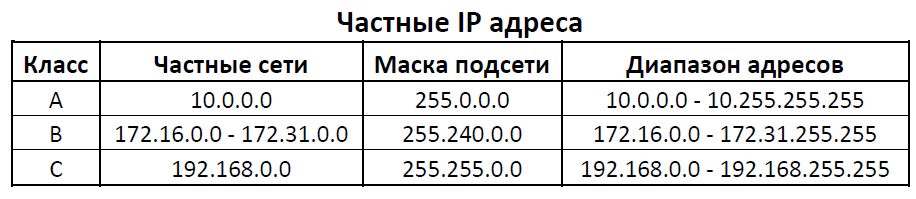
Покажет маршруты, которые кстати будут получены через OSPF (обратите на это внимание)



1. Для выхода в сеть «Интернет» используйте PAT, настроенный на ROUTER2 и ROUTER1-2 соответственно.

Как делать?

NAT – необходимая технология в IPv4 подсетях, учитывая общее крайне ограниченное количество IP-адресов. Он необходим для того, чтобы устройства с частными адресами могли выйти в интернет.

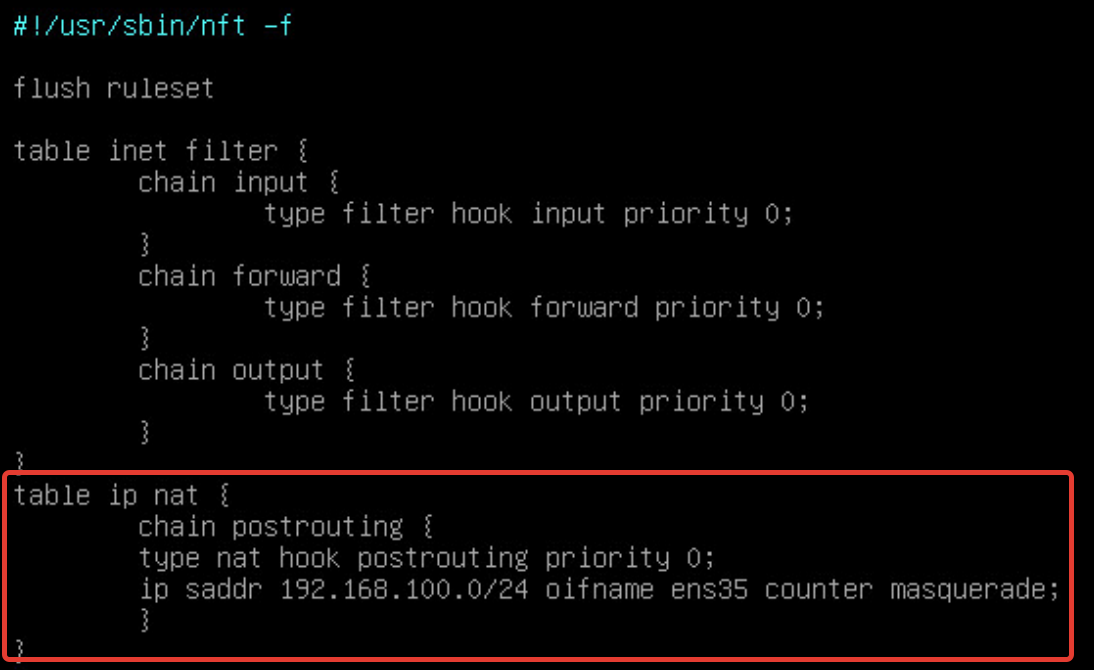


Для этого выполняется его трансляция в публичный адрес, который зачастую установлен на роутерах.

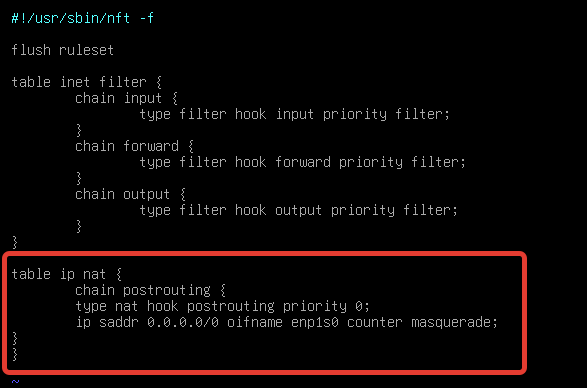
Трансляция сетевых адресов (Network Address Translation, NAT) — это подмена какого-либо адреса или порта в пакете. Она, как правило, требуется на границе между сетью компании и провайдером Интернет. Однако, это далеко не единственная задача. Настроить NAT можно с использованием разных пакетов. Сделаем это с использованием nftables. В Debian 11 пакет установлен по умолчанию, что позволяет сразу перейти к его настройке. Настройка производится при помощи описания конфигурации в файле **/etc/nftables**.

Для начала проведем пример не по заданию.

В этом случае, мы получили работающий NAT (PAT) из подсети 192.168.100.0\24, а имя интерфейса ens35 – из этого примера это интерфейс, настроенный в интернет-провайдера.

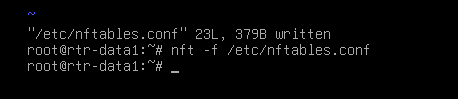


На нашем роутере ROUTER2 можно сделать вот так



Что за подсеть 0.0.0.0\0? Это указание всех подсетей, которые попадут под правило NAT, а интерфейс enp1s0 – то это адрес настроенный в ISP.

Применить настроенный NAT можно командой – **nft -f /еtc/nftables.conf**

****

В этом случае, мы выполнили применение настроек. Если в ответ команда ничего не вернула – сделано все правильно.

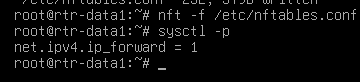
Чтобы добавить эти настройки активными после перезагрузки введите команду – **systemctl enable nftables**

Способ настройки выше подойдет, чтобы сделать NAT быстро и «грубо», для чемпионатов сработает, для реальной жизни – лучше всего указывать конкретные подсети доступные для выхода в Интернет.

Помимо этого, нужно включить на Linux поддержку маршрутизации и пересылки пакетов, для этого –

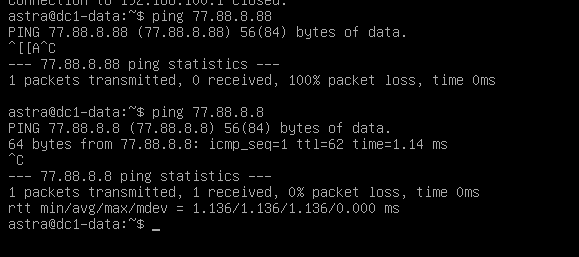
**echo net.ipv4.ip\_forward=1 > /etc/sysctl.conf**

**sysctl -p**

****

Проверить, что NAT работает можно если попытаться «пропинговать» что-то в «Интернете».

Как проверить?



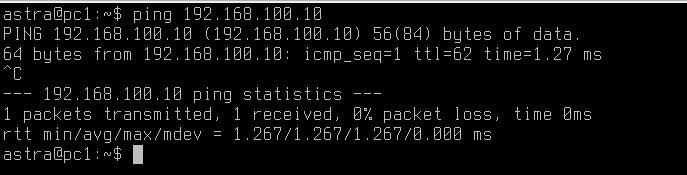
Повторите настройки на ROUTER1 и теперь у вас все сервера и клиенты имеют выход в интернет.

## НАСТРОЙКА МЕХАНИЗМОВ ЦЕНТРАЛИЗОВАННОГО АДМИНИСТРИРОВАНИЯ

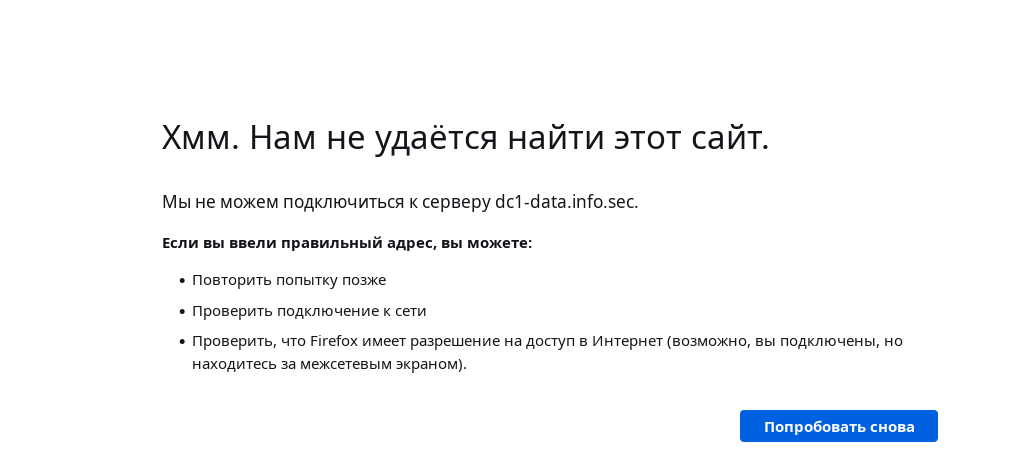
1. Убедитесь, что домен FreeIPA на SERVER-A развернут корректно:

Как проверить?

Для начала стоит проверить доступность с PC1 наш доменный контроллер, ведь FreeIPA намного удобнее администрировать через веб-интерфейс.



Доступ есть, через ранее созданный GRE-туннель. Теперь попробуем зайти на сайт.



Доступа нет! Но это не беда, а особенность Astra Linux – FreeIPA на данной платформе автоматически выполняет редирект на доменное имя. Исправить это можно так – либо добавить строчку в /etc/hosts или указать в качестве основного DNS-сервера именно DATA1, такой вариант будет проще и лучше для нас.

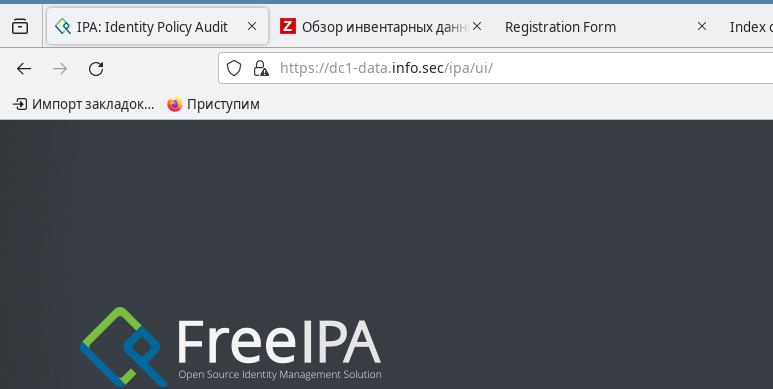
**sudo vim /etc/resolv.conf**

****

**Причем обратите внимание на файл /etc/resolv.conf, он указывает что файл подконтроллен NetworkManager. Давайте выключим его полностью, чтобы избежать ошибок в будущем –**

**systemctl disable --now NetworkManager**

**После этого можно зайти на сайт FreeIPA.**

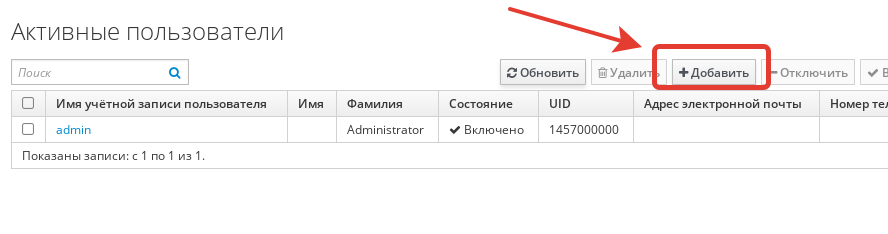
****

1. Создайте структуру пользователей, как в таблице:

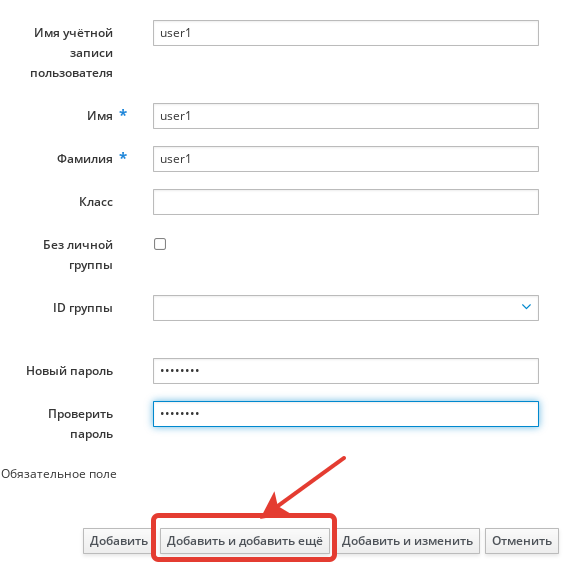
|  |  |  |
| --- | --- | --- |
| Логин | Группа | Пароль |
| User1 | Simple | P@ssw0rd |
| Ivanov | Admin | P@ssw0rd |
| Monitor | Monitoring | P@ssw0rd |

Как делать?

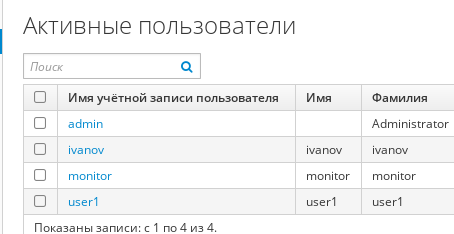
Заходим на FreeIPA



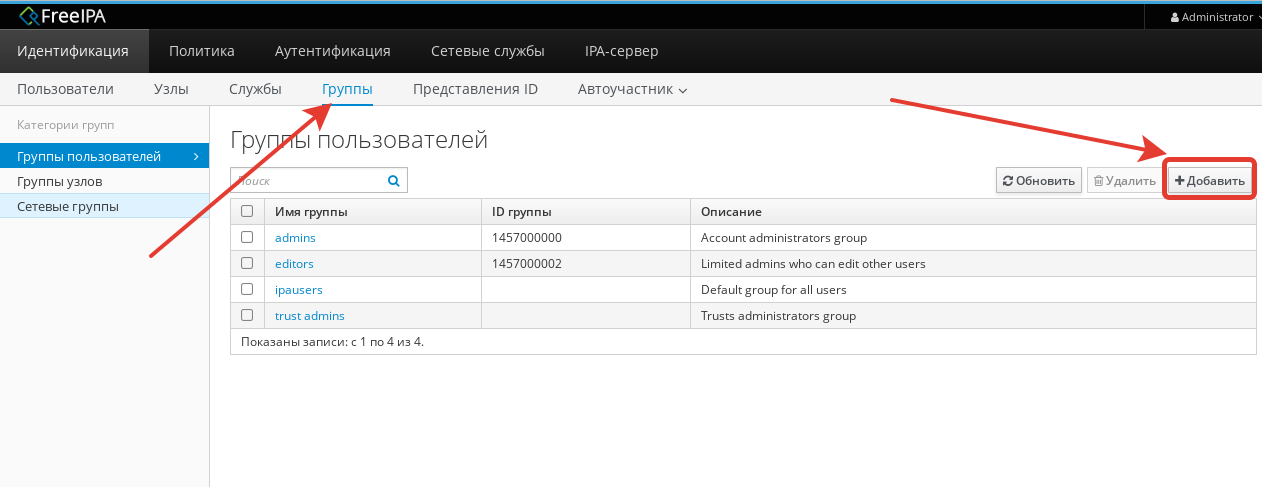
S

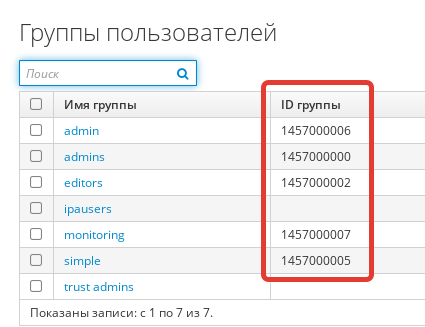


Вот все пользователи и появились

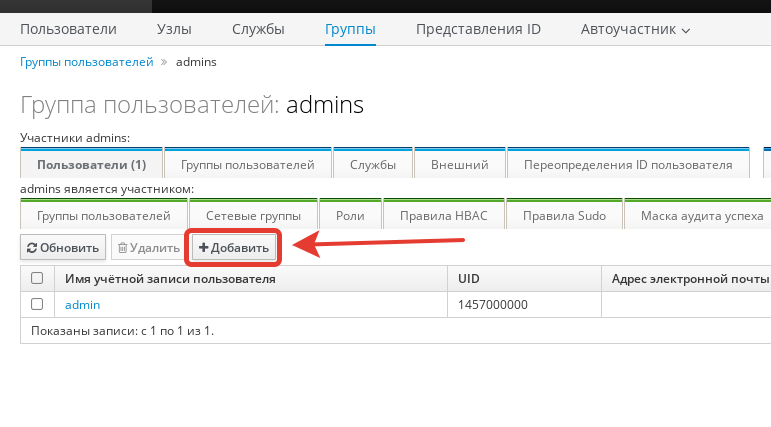


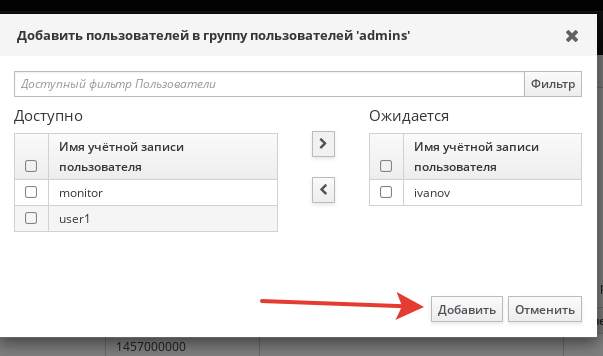
Далее надо создать группы, как требуется по заданию





А затем добавить пользователей в группу, как требуется по заданию. Кликните по группе, и выполните указания на скриншоте.





И далее по аналогии со всеми.

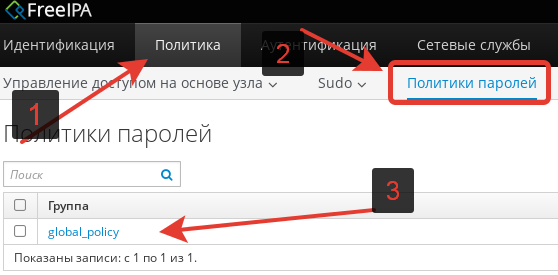
Как проверить?

Посмотреть, что пользователи входят в нужную группу на вкладке пользователей.

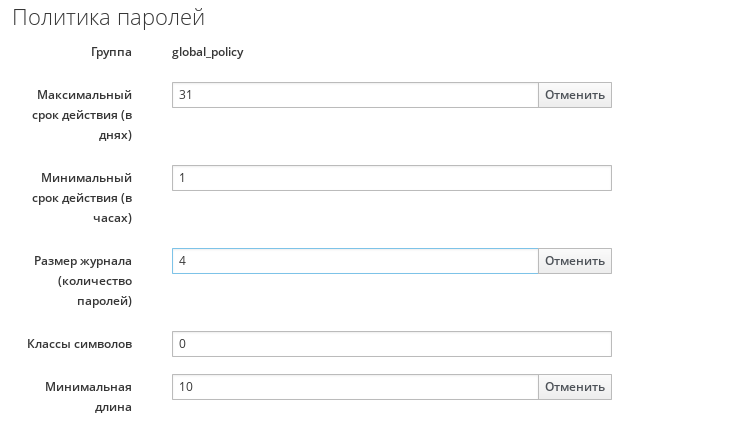
1. Настройте парольную политику таким образом:

* Минимальная длина пароля – 10;
* Журнал паролей должен учитывать не менее 4 раннее указанных паролей;
* Максимальный срок действия пароля – 31 день.

Как делать?

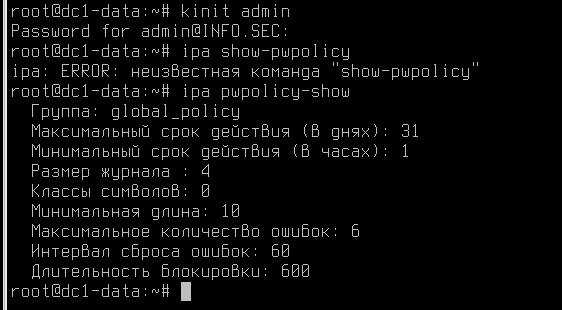


Далее парольная политика настраивается ровно как требуется по заданию.



Как проверить?

Сначала **kinit**, чтобы получить Керберос-ключ для аутентификации и авторизации. А затем **ipa show-pwpolicy**



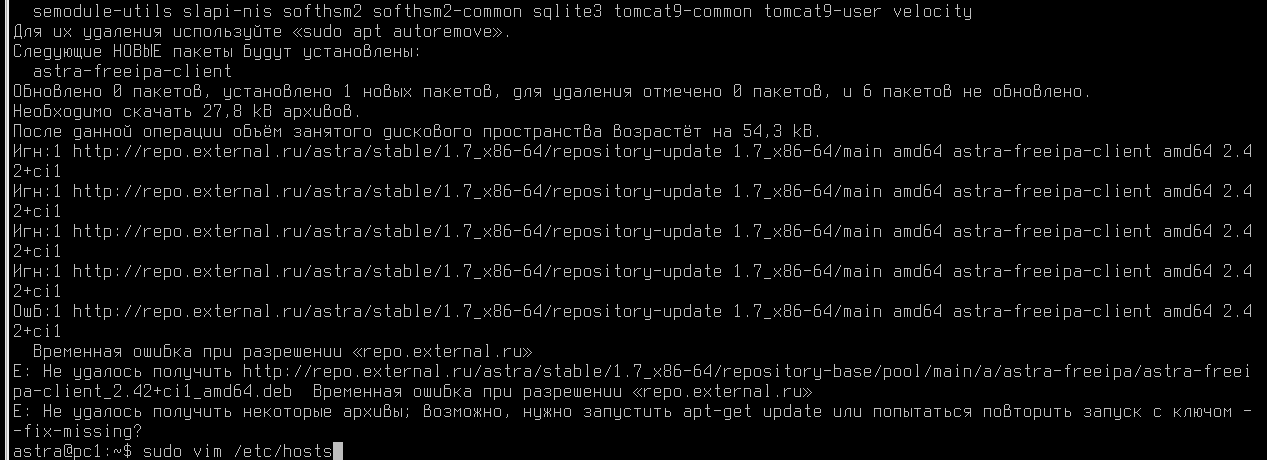
Это альтернативный вариант получить информацию о парольной политике.

1. Добавьте в домен FreeIPA два клиентских компьютера в подсети филиала.

Как делать?

1) Установить на PC1 и PC2 astra-freeipa-client

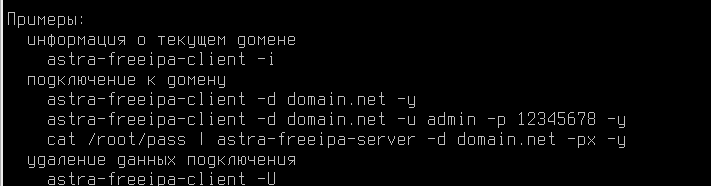
**apt install astra-freeipa-client -y**

****

**Если выйдет ошибка, значит система не смогла найти имя repo.external.ru, можно добавить либо второй DNS-сервер, либо вбить в /etc/hosts. Сделаем второй DNS.**

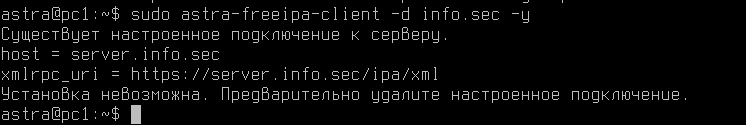
****

С помощью команды **– sudo astra-freeipa-client –help**

****

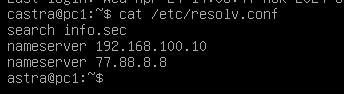
**Проверим, что вводится команда -**

**sudo astra-freeipa-client -d info.sec -y**

****

Исправить эту ошибку – было частью задания. Так что исправляем.

1) Первым делом настроим /etc/resolv.conf



2) Далее удалим ранее развернутый некомпетентными подрядчиками FreeIPA-сервер:

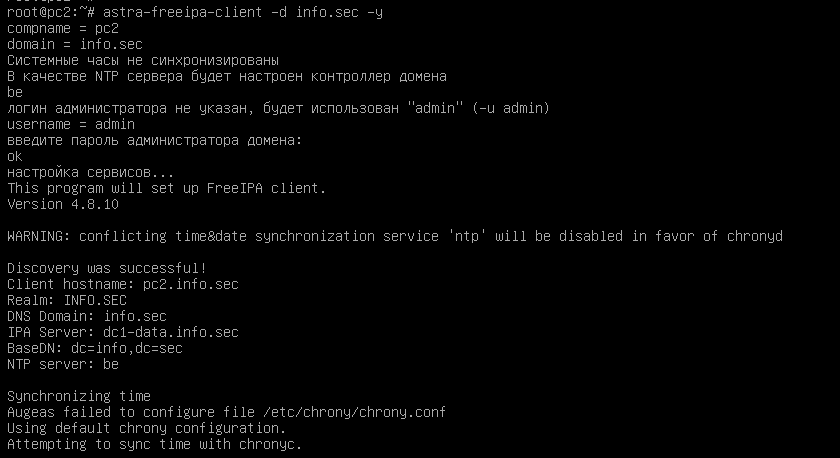
**ipa-server-install –uninstall**

**reboot – после удаления**

**3)**

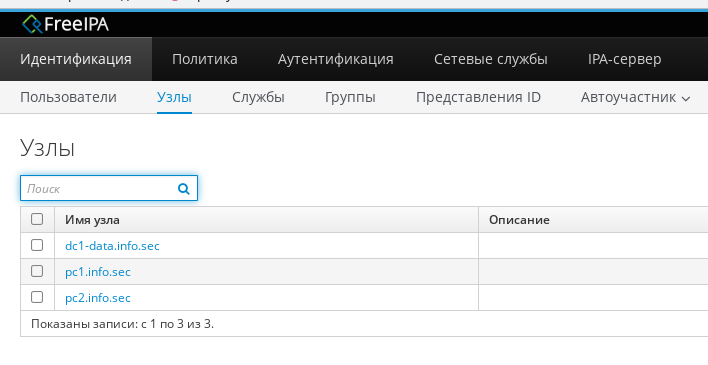
3) Затем введем в домен клиентский ПК –

**astra-freeipa-client -d info.sec -y**

****

Как проверить?

Проще всего будет перейти в интерфейс FreeIPA для проверки.



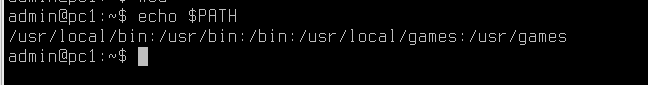
А также авторизоваться под доменным пользователем на компьютере клиента.

1. Настройте параметры аутентификации и работы пользователей:
   * Пользователям из группы Monitoring разрешите выполнять команды ls, head, tail; допускается, чтобы другие команды выполнялись через указание полного пути. Выполните настройку на PC1 и PC2.
   * Пользователям из группы Admin разрешите доступ к командам sudo. Выполните настройку на PC1 и PC2.

Как делать?

Это один из самых спорных пунктов задания, сделать его первую половину (про Monitoring) можно несколькими разными инструментами. Рассмотрим довольно костыльный вариант – через скрипт автовхода в систему.

В системе есть чудесная переменная – PATH, она указывает путь до директорий где система возьмет исполняемые команды для того чтобы сформировать перечень прав пользователя. Наш способ будет направлен на изменение этой переменной у всех в группе Monitoring.



Вот так PATH выглядит у нормального пользователя.

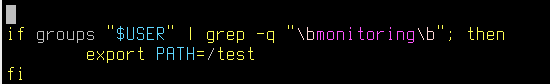
Далее скрипт, который бы сработал на вход пользователя, таких директорий и вариантов несколько:

/**etc/profile**

**/etc/bash.bashrc**

**/etc/bash\_profile**

Мы возьмем **/etc/bash.bashrc.**  Создайте там скрипт



Он проверит, находится ли пользователь в группе monitoring и если да, назначит ему переменную PATH=/test.

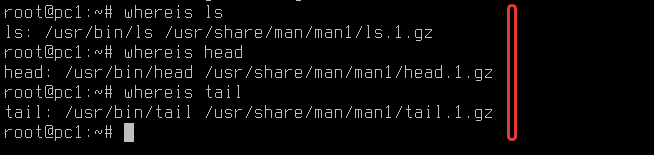
**Что за /test?**

Это папка, с которой мы сейчас начнем работу.

**mkdir /test**

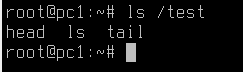
Папку сделали, теперь накопируем туда исполняемые файлы, что нужны нам по заданию

Найти исполняемый файл любой команды в Linux можно с помощью – **whereis**



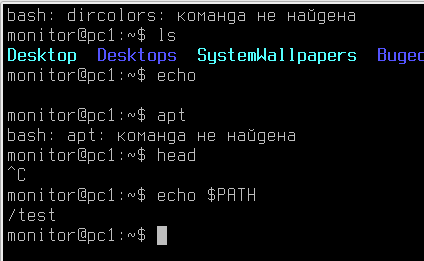
Накопируем все /usr/bin/\* файлы в директорию /test

**cp /usr/bin/ls /test**



Как проверить?

Войдем в систему под пользователем из группы monitoring



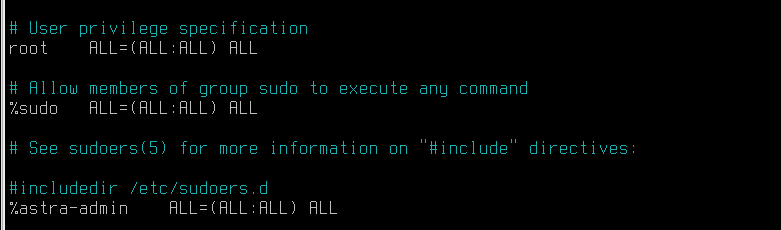
Все работает!

Вторая часть задания

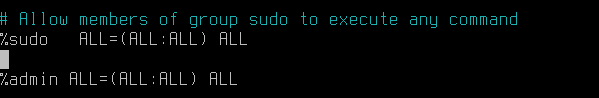
* + «Пользователям из группы Admin разрешите доступ к командам sudo. Выполните настройку на PC1 и PC2.»

Как делать?

Файл, который отвечает за то какие права sudo можно раздать называется - **/etc/sudoers.** Перейти к его редактированию можно командой – **visudo**.



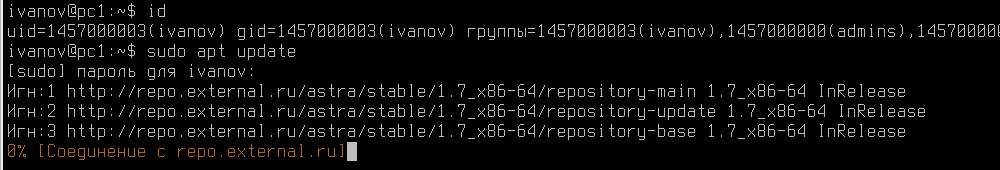
И здесь уже есть пример, например, для группы sudo, которая позволяет выполнять любые команды членам этой группы. Скопируем конструкцию, но даем права на доменную группу admin.



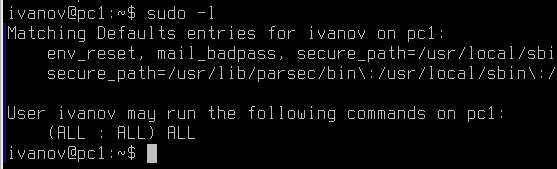
Получается так.

Как проверить?

Зайдем под пользователем из группы Admin FreeIPA, проверяем доступ к sudo



Также есть крутая команда – **sudo -l**

****

Тоже покажет список всех доступных судо-команд

1. В качестве инструмента групповых политик используйте \_\_\_\_\_\_\_\_\_\_.

Как делать?

Вариантов выполнения этого задания довольно много, можно было сделать через Ansible, Puppet, SaltStack, Chef и, может быть, другие разные инструменты.

Рассмотрим вариант с Ansible.

Где расположить скрипт?

Логичнее всего расположить на доменном контроллере.

1. Установим Ansible

**apt install ansible -y**

2. После этого переходим в каталог – **cd /etc/ansible**

3. Открываем файл – **vim /etc/ansible/ansible.cfg**  - это основной конфигурационный файл Ansible.

Конфигурационный файл Ansible может храниться в разных местах (файлы перечислены в порядке уменьшения приоритета):

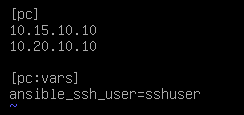
ANSIBLE\_CONFIG (переменная окружения)

* ansible.cfg (в текущем каталоге)
* ~/.ansible.cfg (в домашнем каталоге пользователя)
* /etc/ansible/ansible.cfg

В этом файле найдите строку – host\_key\_checking и приведите её к виду



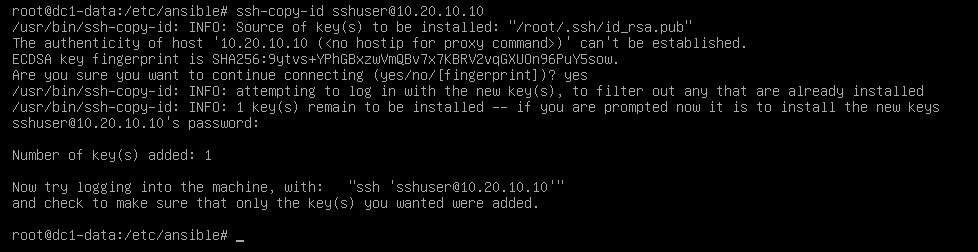
4. Далее про файл hosts – в этом файле хранится информация о хостах, которые будут добавлены к администрированию.



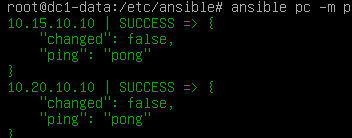
Для обеспечения безопасного доступа к хостам, настроим аутентификацию по ключам. Ansible по умолчанию выполняет именно такой формат подключения.

Помним, что все хосты доступны только под пользователем sshuser, именно его мы и указали в переменной ansible\_ssh\_user. Потому что Ansible, по умолчанию, пытается подключиться под логином того пользователя, кто запускает плейбук на сервере.

Через **ssh-keygen** – создайте пару ключей, а через **ssh-copy-id** передайте её на сервер



Проверить, что все работает хорошо и Ansible может подключаться к хостам можно через команду: **ansible pc -m ping**

****

Отлично, мы наладили безопасное соединение. Можем писать скрипт.

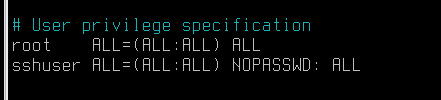
Вообще, Ansible имеет такое понятие как модули – они позволяют делать настройки быстро, надежно и удобно. Данные модули разрабатываются вендором или сообществом, узнать о всех модулях можно на сайте [ansible-galaxy](https://galaxy.ansible.com/)

Для выполнения задачи выше можно использовать обычный модуль shell или command, который просто выполнить команду на удаленном устройстве. Крайне нежелательно использовать модуль shell в работе всегда, ведь он имеет ряд недостатков. Для выполнения пункта задания выше – подойдет.

Итак, плейбук можно создать командой – **vim /etc/ansible/runme.yml**

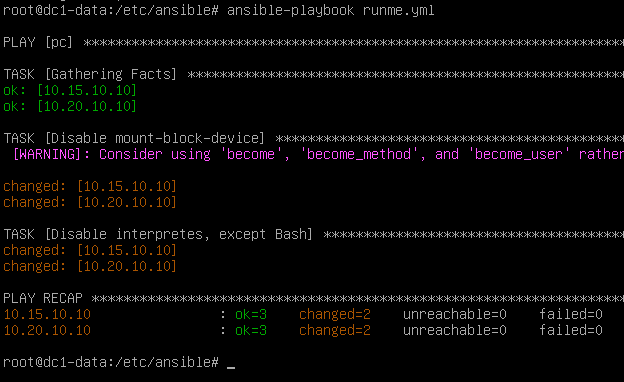


**Наблюдательный читатель заметит здесь sudo, и задумается – а как же пользователь sshuser выполнит такую команду без доступа к sudo? Настроим данному пользователю беспарольный доступ к такой команде вручную, на PC1 и PC2 в файле /etc/sudoers сделайте так:**

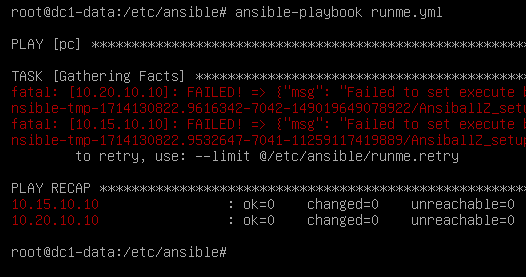
****

**Текстом задания не запрещено вносить правки в visudo, поэтому делаем так.**

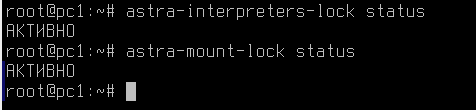
Пробуем запускать.



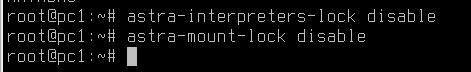
Важно уточнить, в специфике задания после этих правок Playbook больше не заработает, на всех машинах будет заблокирован Python.



Как проверить?



Если захочется еще раз плейбук запустить, выключите сделанные ранее настройки через команды



После этого плейбук можно снова запустить, и убедиться что он включит обратно все ограничения в ОС.

1. Используйте любой инструмент сертификации. В случае, если это будет не FreeIPA, директория для сертификатов - **/etc/ca. Все службы, что требуют HTTPS используют этот ЦА!**

Как делать?

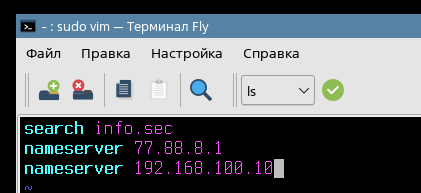
Можно использовать центр сертификации от FreeIPA, можно использовать решения по типу – easy-rsa или openssl.

В тексте задания мы будем работать с FreeIPA, в дальнейшем при выполнении пунктов задания по HTTPS мы подробнее разберем этот функционал.

1. В качестве DNS-сервера используйте FreeIPA.

Как делать?

В /etc/resolv.conf



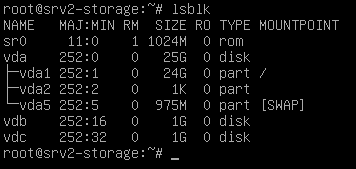
Убедитесь, что такие настройки выполнены везде.

## НАСТРОЙКА ВЕБ-СЛУЖБ И СИСТЕМ ХРАНЕНИЯ ИНФОРМАЦИИ

1. На SERVER-2 сконфигурируйте LVM том.

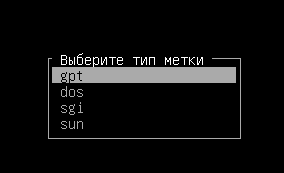
Как делать?

Для начала, давайте проверим что у нас есть подключенные диски командой lsblk.



Наблюдаем три диска – vda, который уже размечен на три раздела – vda1,vda2,vda5 эти разделы системные (так как смонтированы в /). Их трогать не надо. А вот остальные диски – vdb, vdc. С ними и работаем, для начала создать на них разделы, удобнее всего это сделать через cfdisk.

**cfdisk /dev/vdb**

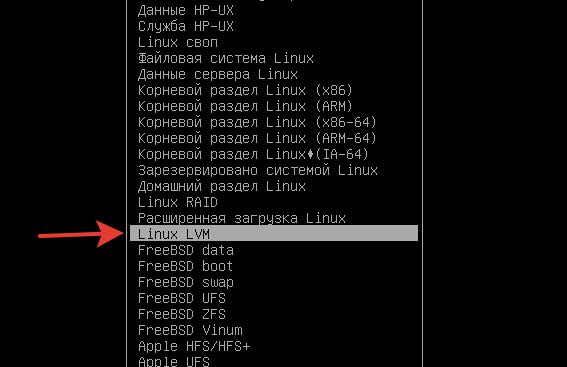


Создаем новый раздел

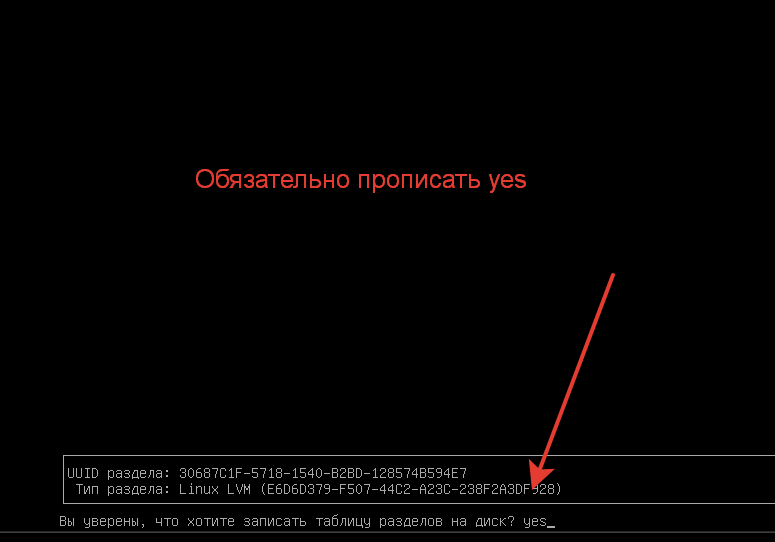


**Размер раздела: 1023М**

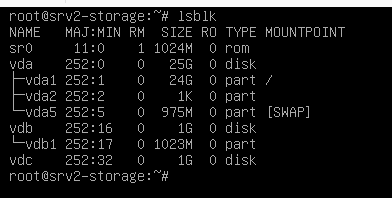
****





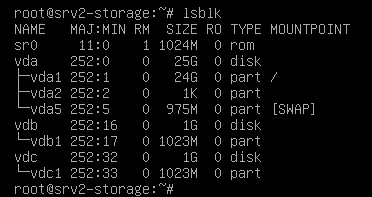


Затем выход из программки, проверить что все ок можно через – **lsblk**



Как мы можем заметить, теперь диск **/dev/vdb имеет раздел /dev/vdb1.** Напомню, что раздел – это уже логическое пространство данных, куда мы можем складывать наши данные, создавать папки и файлы.

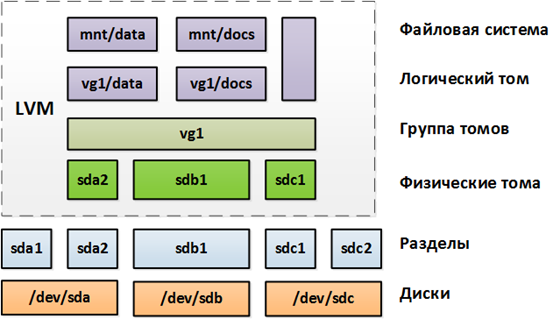
Повторите аналогичную процедуру с диском **/dev/vdc**



**Отлично, диски подготовлены, теперь мы можем укомплектовать наш LVM и зашифровать его через dm-crypt.**

Установим необходимые пакеты**:**

**apt install lvm2 cryptsetup**

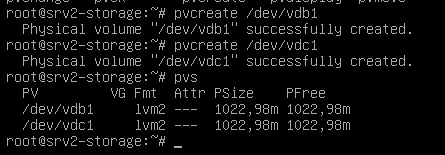
****

Немного теории о LVM – он выглядит как слоеный пирог, где каждый слой идет поверх другого.

Первый слой - физические тома, это те же самые разделы нашего диска, который мы сделали выше, но при этом – эти диски прошли инициализацию в LVM.

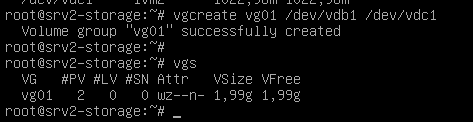
**Как это сделать?**

**pvcreate /dev/vdb1 /dev/vdc1**

****

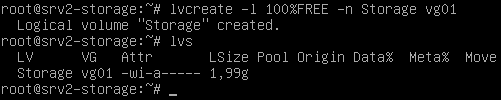
**Следующий слой – группа томов.**

**vgcreate vg01 /dev/vdb1 /dev/vdc1**

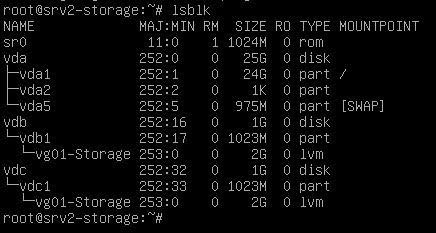
****

**И финальный слой - логический том.**

**lvcreate -l 100%FREE -n Storage vg0**

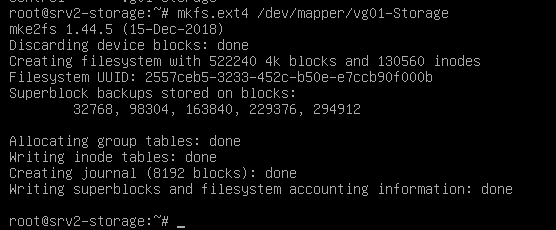
****

**Проверим, как изменился вывод – lsblk**

****

**После этого на логическом разделе нужно создать файловую систему**

**mkfs.ext4 /dev/mapper/vg01-Storage**

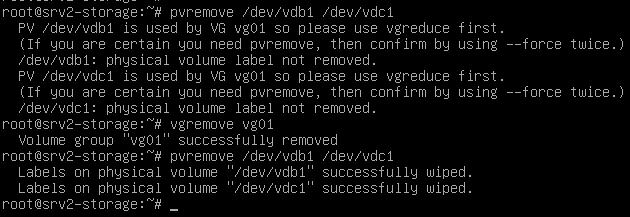
****

**Отлично, это все сборка LVM2 без шифрования. Будет полезно знать как это делать и без шифрования.**

**Теперь шифруем!**

**Да, попробуем разобрать нашу текущую инфраструктуру на crypsetup.**

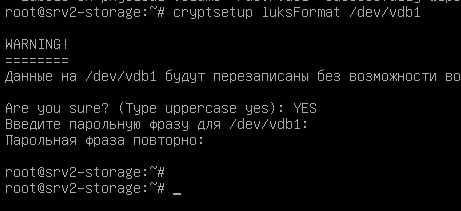
**Для начала, удалим все что было сделано выше.**

****

**А теперь наши разделы зашифруем –**

**cryptsetup luksFormat /dev/vdb1**

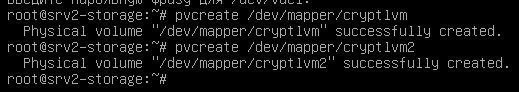
**cryptsetup luksFormat /dev/vdc1**

****

**crypsetup open /dev/sdb cryptlvm**

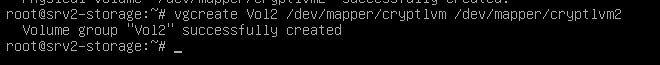
**crypsetup open /dev/sdc cryptlvm2**





Объединим в одно группу два наших зашифрованных

**vgcreate Vol2 /dev/mapper/cryptlvm /dev/mapper/cryptlvm2**

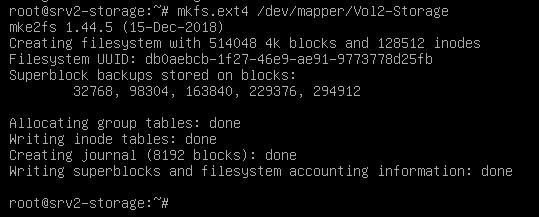


А теперь LVM соберем –

**lvcreate -l 100%FREE Storage Vol2**



**mkfs.ext4 /dev/mapper/Vol2-stripe\_vol**



**Как проверить, что все хорошо?**

**lsblk -f**

****

**Важно заметить – слова crypto\_LUKS повсюду – значит все правильно.**

Но что теперь? Да, раздел собран и он зашифрован.

**Но!** Любое обращение к диску – через пароль, хранение данных или авто монтирование также через пароль. Как это оптимизировать?

1. Сделать ключ

**dd if=/dev/urandom of=secretkey bs=512 count=4**

2. Скопируем в /etc/

**cp secretkey /etc/**

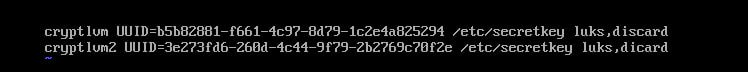
**3.** Добавим ключ в парольную фразу LUKS

**cryptsetup luksAddKey /dev/vdb1 /etc/secretkey**

**cryptsetup luksAddKey /dev/vdc1 /etc/secretkey**

**4.** Теперь доступ до ресурсов возможен как через пароль, так и через ключ.

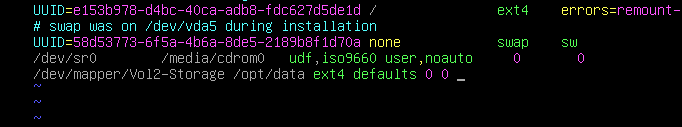
Идем в файл **/etc/crypttab**



Настраиваем его как на скриншоте, UUID легко получить через команду blkid, например –

**blkid >> /etc/crypttab**

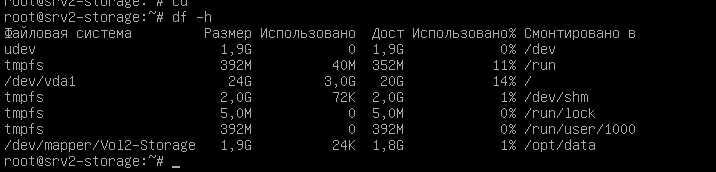
**В /etc/fstab при этом монтирование как обычно**

****

**Далее примонтировать все, что было описано в /etc/fstab можно через команду –**

**mount -a**

**Проверить, что все смонтировалось корректно через - df -h**

****

1. На сервере SERVER-2 реализуем веб-сервер в режиме файлового сервера:
   * В качестве хранилища используем /opt/data;
   * Доступ только для авторизованных пользователей – user:P@ssw0rd;
   * Файлы на веб-сервере индексируются и позволяют скачать их только после авторизации на сервере;
   * Сервер доступен по имени – [www.info.sec](http://www.info.sec);
   * Реализуем протокол HTTPS.

Как сделать?

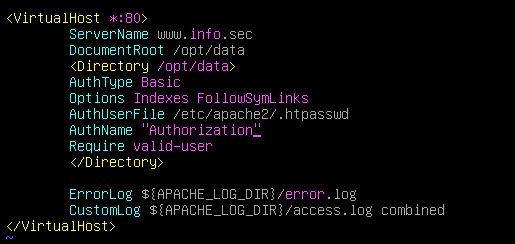
Это можно сделать как в Apache2, так и в NGINX.

Мы рассмотрим вариант через Apache2.

Для начала подготовим конфигурацию через протокол HTTP.

Идем в файл - **/etc/apache2/sites-enabled/000-default.conf**

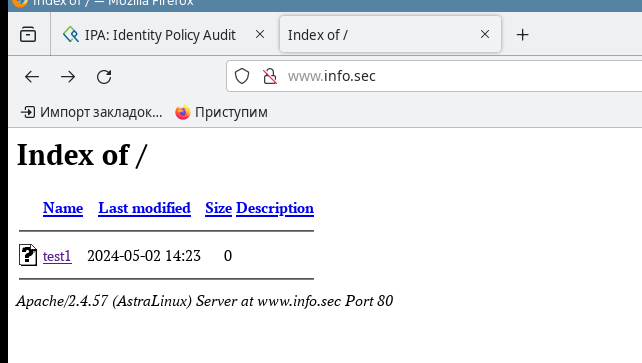
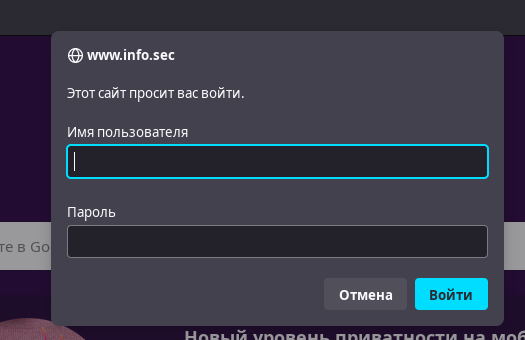
Удаляем все лишнее, и приводим файл к виду:



Затем создаем файл с хешем пароля и логином нашего единственного пользователя по заданию – user:P@ssw0rd

**htpasswd -c /etc/apache2/.htpasswd user**

Готово! Теперь можно сайт проверить, откуда-нибудь с PC1.

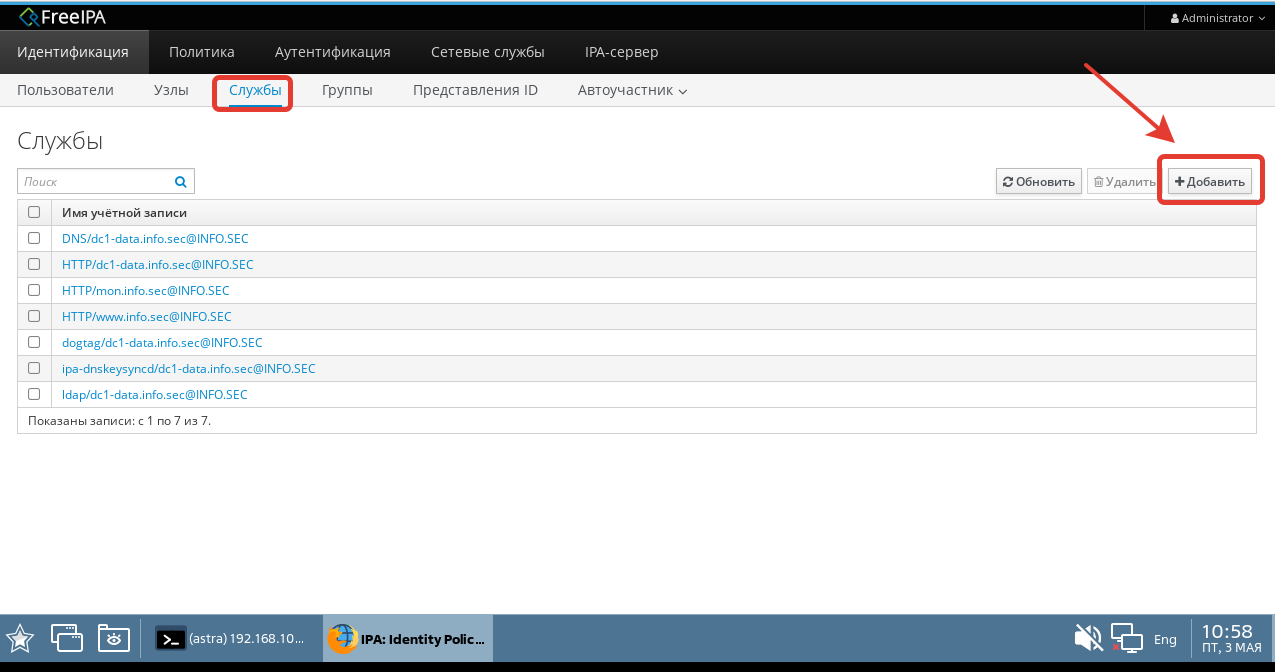
 

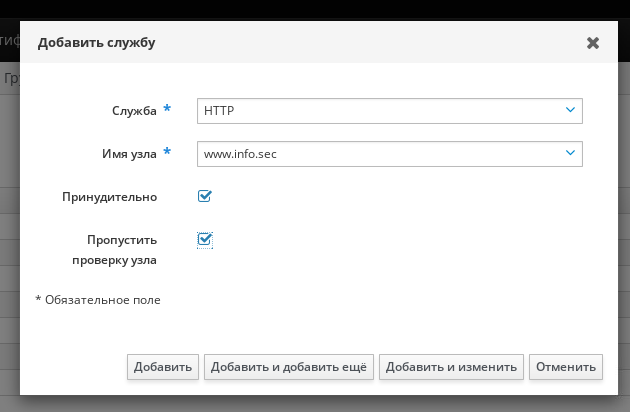
Отлично, HTTP готов, теперь пора прикрутить к нему сертификат.

Будем пользоваться решением от FreeIPA.

Как выпустить сертификат во FreeIPA?

1. Создаем службу в веб-интерфейсе FreeIPA

Заполните форму, как на примере:



После этого, удобнее всего будет подключиться по SSH к SERVER-A с PC1, и получить керберос-ключ для администратора домена.



Далее, необходимо привязать службу HTTP к хосту в домене, лучше всего в нашем случае будет привязать её к доменному контроллеру командой –

**ipa service-add-host –hosts=SERVER-A.info.sec HTTP/www.info.sec**

*На примере ниже привязка идет к mon.info.sec, не отвлекае***мся 😊**



А затем выпускаем сертификат командой –

**ipa-getcert request -r -f /opt/cert1.crt -k /opt/cert1.key -N CN=www.info.sec -D** [**www.info.sec**](http://www.info.sec) **-K HTTP/www.info.sec**

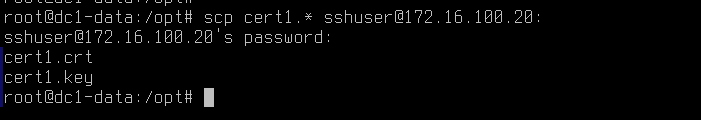
****

**Проверьте, чтобы в /opt появились ваши сертификаты!**

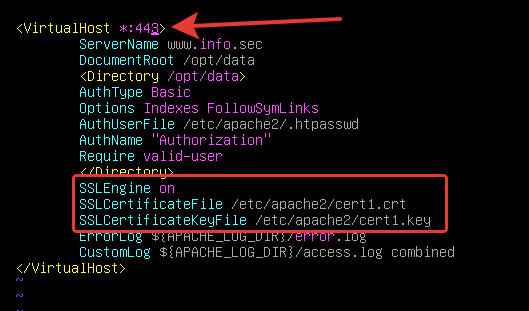
**А также, в веб-интерфейсе FreeIPA**

****

**Далее передаем свежие сертификаты на сервер SERVER-2**

****

**После этого, вернемся на SERVER-2, перенесите сертификаты из /home/sshuser в /etc/apache2/. А затем, отредактируйте конфигурационный файл вашего веб-сервера (на скриншоте выделены моменты, что мы поменяли)**

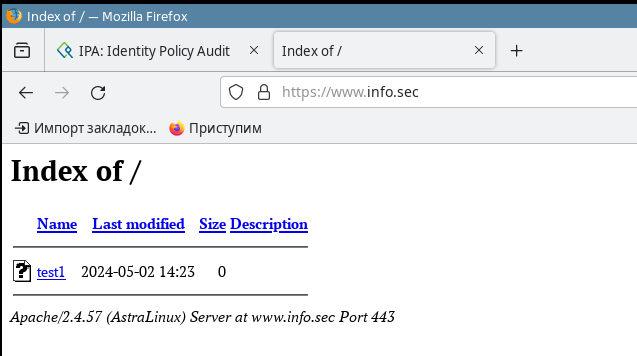
****

**Далее вводим команды:**

**a2enmod ssl**

**systemctl restart apache**

**После этого, перейдем на веб-сервер с PC1**

****

**Соединение успешно защищено! Правда, у вас будет сейчас ошибка!**

**Ошибка вызвана тем, что Firefox при добавлении сертификатов не подключает их автоматически.**

**Как только мы ввели в домен PC1 и PC2, они автоматически настроились на доверие к сертификатам от домена FreeIPA.**

**Но в Firefox это нужно доработать.**

**Для этого:**

**1) Удалите старую библиотеку из Firefox –**

**rm -rf /usr/lib/firefox/libnssckbi.so**

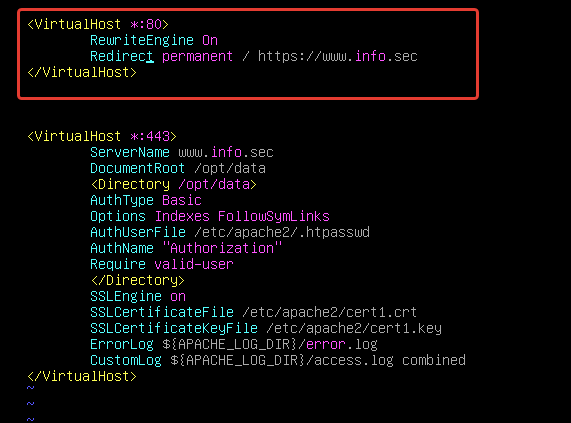
**2) Подменить её на другую –**

** ln -s /usr/lib/x86\_64-linux-gnu/pkcs11/p11-kit-trust.so /usr/lib/firefox/libnssckbi.so**

**После этого, доступ до сайта будет корректно работать, только по протоколу HTTPS.**

Доработаем конфигурацию.

Хоть и в задании это не требуется, но мы для общего развития доработаем конфигурацию, а именно - настроим автоматический редирект с HTTP до HTTPS.



Далее:

**a2enmod rewrite**

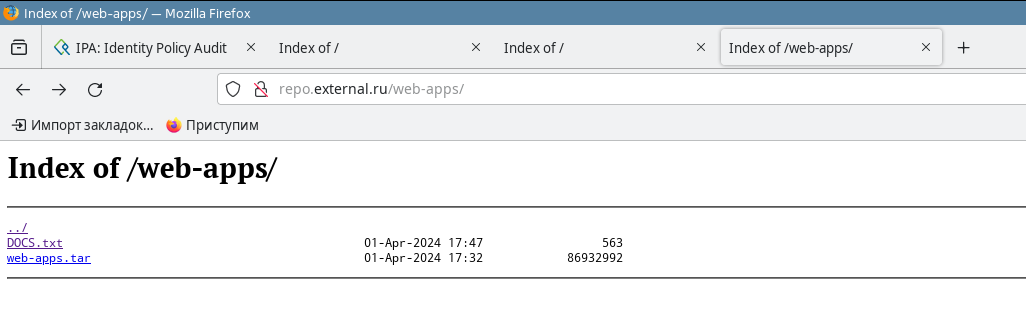
**systemctl restart apache2**

Теперь у нас сайт доступен всегда по протоколу HTTPS, даже если пользователь в браузере вручную введет <http://www.info.sec>

1. Обеспечим корректную работу веб-приложения на сервере SERVER-1. Документация к ПО доступна на сайте – repo.external.ru/web-apps.

Как делать?

Для начала, конечно, перейдем по указанному в задании URL – http://repo.external.ru/web-apps



Имеется два файла - текстовый и архив. Начнем с текстового файла.

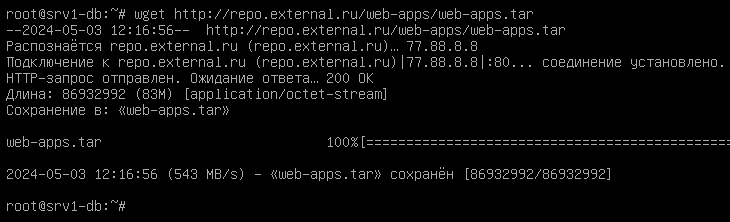
Открыв его, мы можем кратко понять, что это запакованная в докер-контейнер программа. Эта программа является бета версией будущего корпоративного портала. Дана инструкция, как развернуть и запустить приложение, а также указаны веб-маршруты:

/info

/

/register

Скачаем web-apps.tar сразу на SERVER-1



Если команды wget нет – **apt install wget**

Также, на SERVER-1 установите Docker –

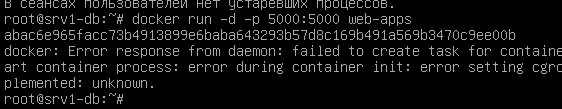
**apt install docker docker.io**

Докер установлен, выполняем 1 пункт инструкции к приложению **–**

**docker load < web-apps.tar**

****

**А теперь пробуем запустить, но вот незадача – не запустится контейнер!**

****

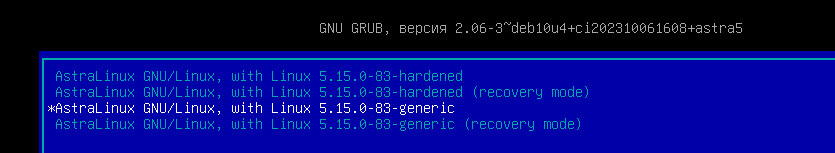
**Причина этому проста, у нас ядро hardened – а оно docker не поддерживает.**

****

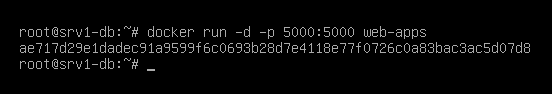
**Скачаем generic ядро –**

**apt install linux-image-5.15.0-83-generic**

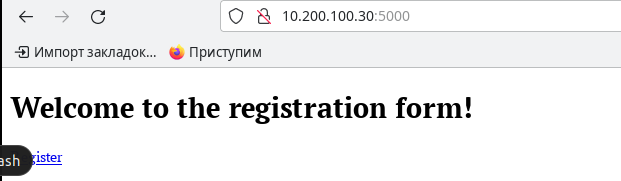
**После установки generic ядра, перезапускаем компьютер и грузимся с нового ядра**

****

**После этого запустится все прекрасно**

****

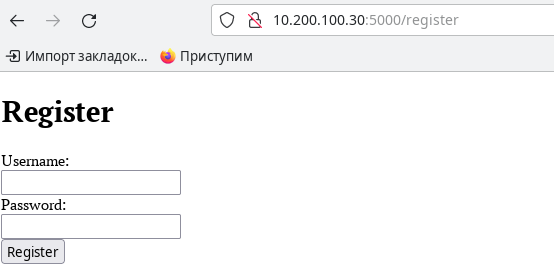
**Проверяем работу приложения –**

****

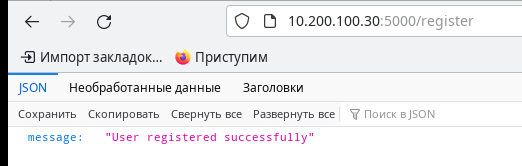
**Работает! Теперь проверяем работу приложения и даем свой вердикт, как специалисты по ИБ.**

**Путь / - приводит нас на главную страничку, тут интересного ничего нет.**

**Путь /register – форма регистрации**

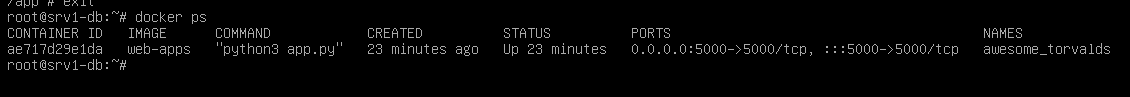
****

**Попробуем зарегистрироваться**

****

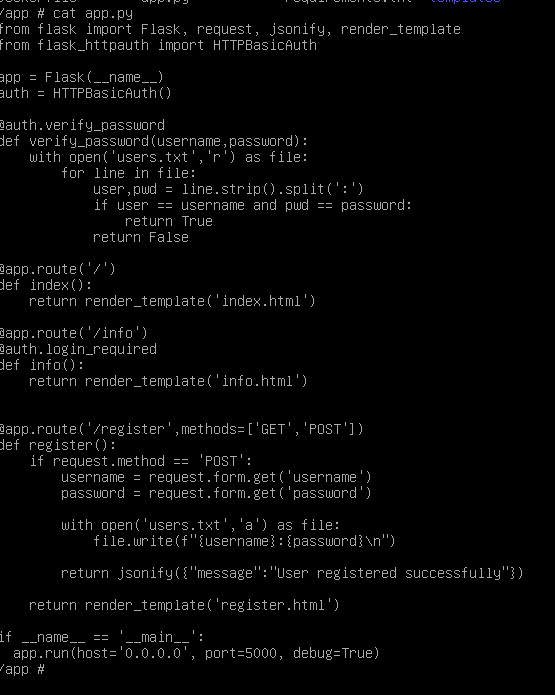
**/info – работает нестабильно, об этом нас уведомляет разработчик.**

Анализ HTML страниц ни к чему не приводит, пойдем проверять сам контейнер?

****Через **docker ps**

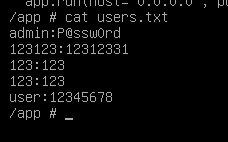
**Находим имя нашего контейнера в столбе NAMES.**

****

**Попали в контейнер! Посмотрим, что делает app.py – кажется это основной код нашего приложения**

**Самый большой интерес вызывает поле /register, а именно формат хранения пользователей в файле users.txt.**

**Проверяем?**

****

**О как! Все пароли в открытом виде – непорядок, таким пользоваться нельзя!**

**Пишем это в аргументацию к заданию и получаем баллы. Больше проблем с приложением нет (*наверное, как минимум я не задумывал*).**

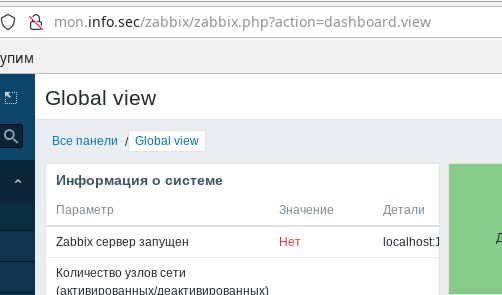
**Но, если нашли бы еще что-то – смело пишите об этом в отчете. Тут лучше работать по правилу, чем больше, тем лучше. Зачастую самые полноценные и хорошо документируемые отчеты получают много баллов.**

4. Zabbix-сервер: Обеспечение его безопасности.

Как делать?

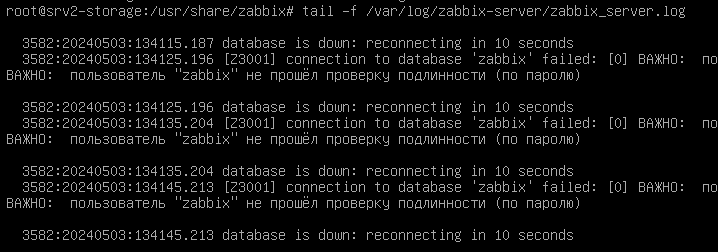
0) Добавить в DNS запись – mon.info.sec

Для начала починить Zabbix, ведь подрядчик его не доделал нормально



Выходит ошибка, что якобы сервер неактивен. Исправляем это.

Для начала, понять бы что случилось, посмотрим логи



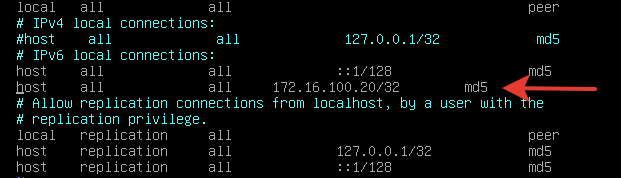
Ага, к базе данных не подключается. Посмотрим, что на SERVER-1?

Идем в файл - **/etc/postrgresql/11/main/pg\_hba.conf**

Наблюдаем две странных настройки



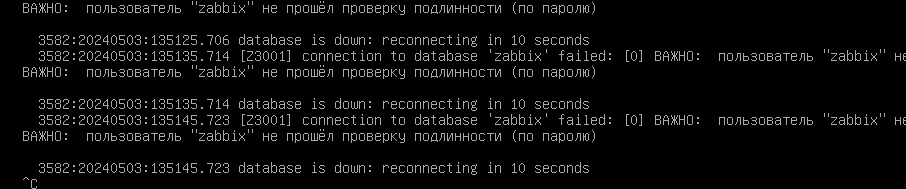
Данные настройки позволяют кому угодно, и откуда угодно подключаться к нашей БД без пароля – это неправильно, исправляем



Перезагружаем службу postgresql –

**systemctl restart postgresql**

Но ошибки Zabbix это не исправило.

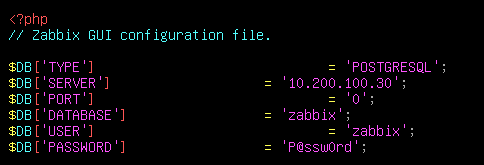


Пройдем по конфигурационным файлам Zabbix. Их всего два –

1) **/etc/zabbix/zabbix.conf.php**

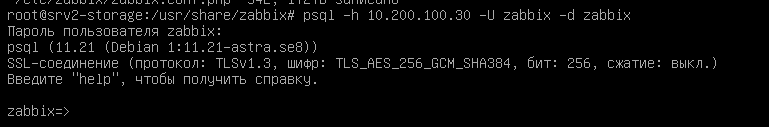
2) **/etc/zabbix/zabbix\_server.conf**

Первый файл посмотрим, и там такое –

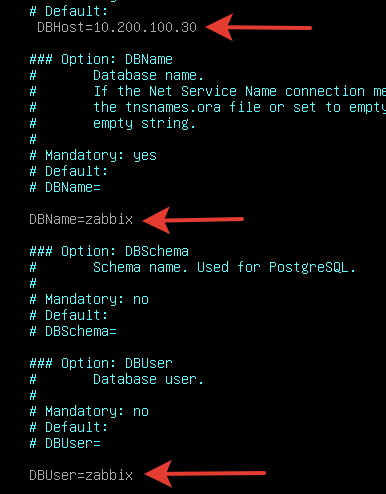


Настройки выглядят правильными, но нужно убедиться что именно так мы можем подключиться к БД. –

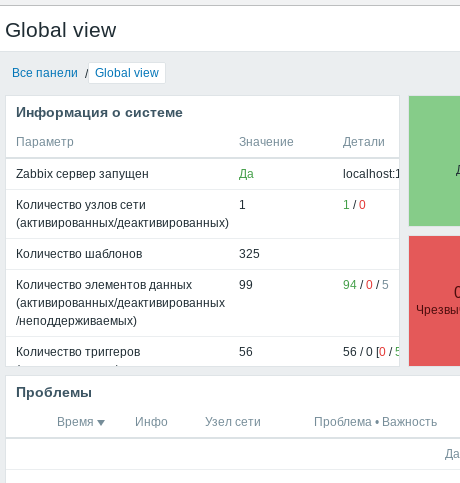
**psql -h 10.200.100.30 -U zabbix -d zabbix**

****

**Подключается, значит в этом файле конструкция правильная. Проверяем второй. Здесь если параметры DBHost, DBName, DBUser – они отвечают за сведения о подключениях к БД. Тут то и беда. Исправьте их на верные, как тут -**

****

**После этого, ребутнем zabbix-server и посмотрим что получится.**

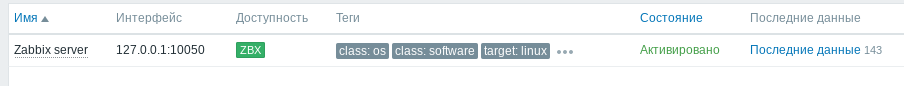
****

**Теперь все работает, да еще и ошибки небезопасной конфигурации исправили!**

Осталось добавить хосты для мониторинга.

По условию задания, добавить в мониторинг надо все роутеры и сервера.

Начнем по порядку, с нашего же SERVER-2. Для того чтобы добавить Zabbix-сервер в мониторинг «самого себя», достаточно просто установить и включить zabbix-agent.

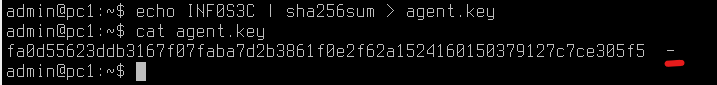


Вот если вы видите все как выше - значит все круто.

Но! Это мы добавили агент без шифрования, а по условиям задание нам надо зашифровать соединение.

Для выполнения этой части задания потребуется первым делом сгенерировать ключ:

echo INF0S3C | sha256sum > agent.key



Причем обратите внимание на «черточку» в конце,она не нужна в итоговом файле, так что после выполнения команды отредактируйте файл, удалив этот символ.

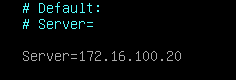
Этот ключ нужно передать на все клиенты, кто будет подключаться к Zabbix.

Так как мы работаем с SERVER-2, в качестве примера, файл отправим туда. Но распространить, ровно, как и настроить, нужно будет на всех серверах и роутерах.

Далее, на SERVER-2, файл **/etc/zabbix/zabbix\_agentd.conf**:

Для удобства, все указанные ниже параметры актуальны для любого хоста на стенде:

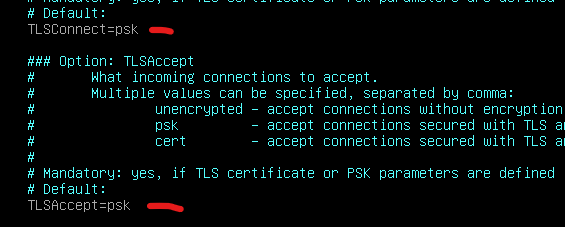
1. Корректируем параметр сервер –



2. Далее уже важные вещи, такие как конфигурация TLS PSK

**TLSConnect=psk**

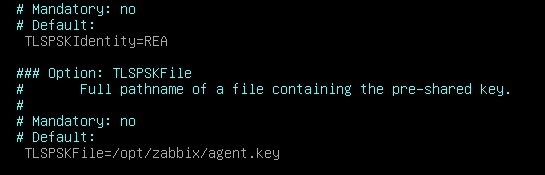
**TLSAccept=psk**



3. И замыкаем настройки

**TLSPSKIdentity=REA**

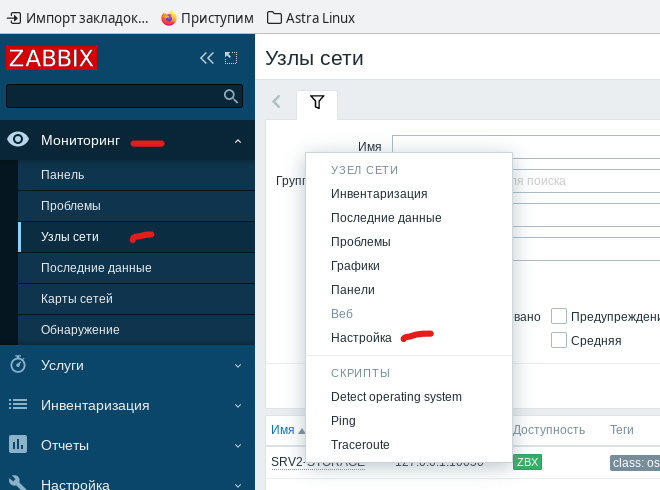
**TLSPSKFile=/opt/zabbix/agent.key**



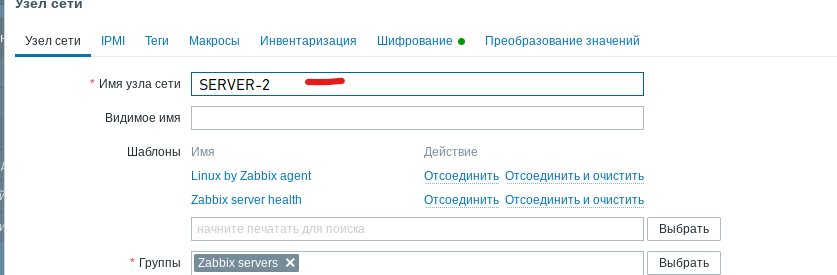
Далее разместить ключ по тем директориям, что планировалось в конфигурации и выполняем перезагрузку.

На стороне сервера, в веб-интерфейсе делаем так:

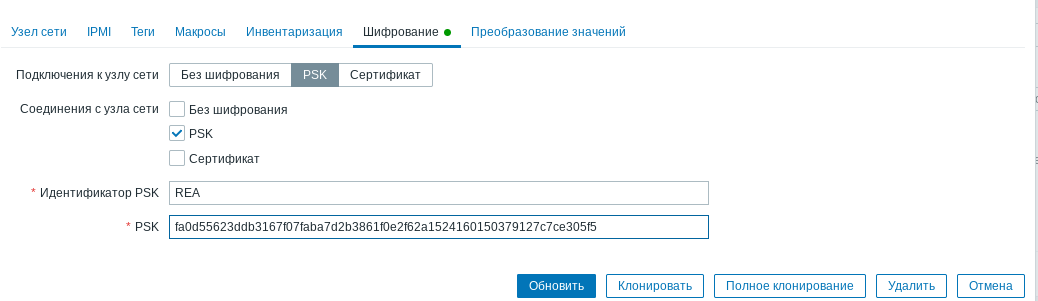
Переходим в Узлы сети



**Поменяем имя, для красоты**

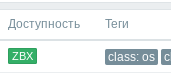


Дальше параметр Шифрование, и делаем как на скриншоте



Все готово, шифрование корректно настроено!

Помните, что когда вот так:



Значит все хорошо, а еще на первой вкладке, где мы меняли имя обратите внимание на шаблоны там есть - Linux Template, именно его нужно будет настроить все всех подключаемых клиентах.

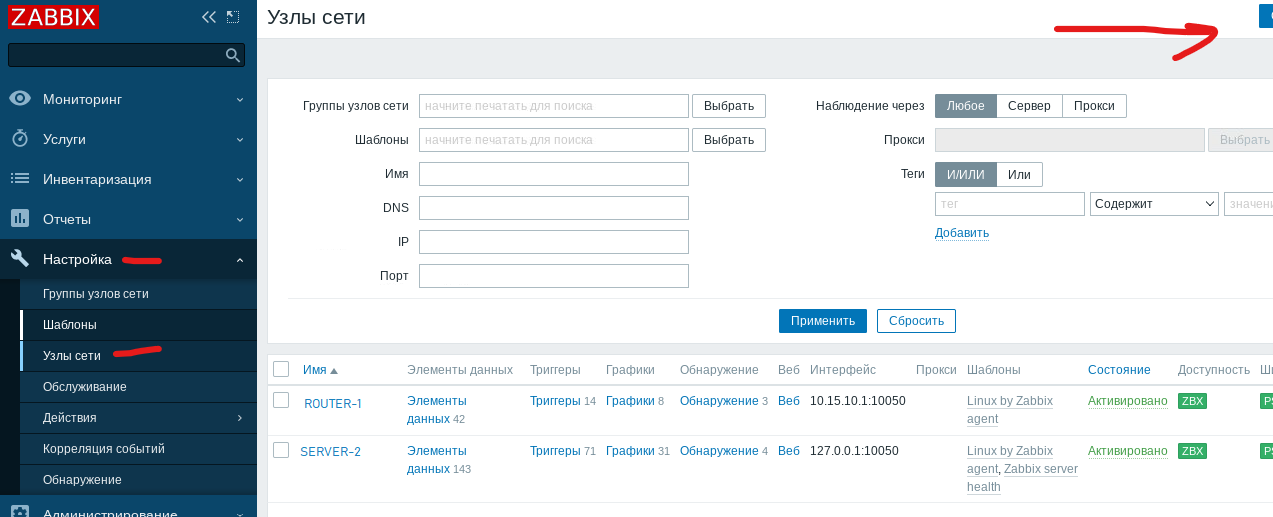
Закрепим успех и подключим, например, роутер – ROUTER1:

1. **apt install zabbix-agent -y**

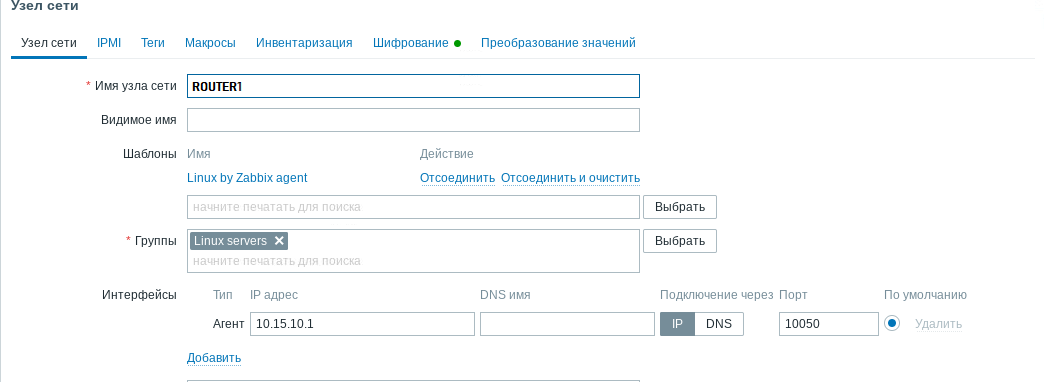
**2. Затем, передать ключ, настроить все как по аналогии выше.**

**В веб-интерфейсе идем:**

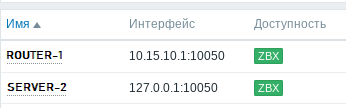
**Настройка – Узлы сети, в правом верхнем углу будет «Создать узел сети»**



Заполняем под ROUTER1



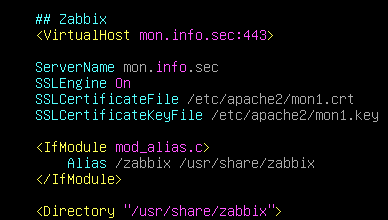
Не забываем про шифрование. Готово! Теперь ребут zabbix-agent на стороне клиента, ждем пару минут и в интерфейсе видим:

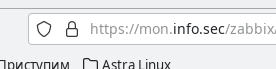


По аналогии добавляем всех!

**Остается финальный штрих – HTTPS.**

**В шапке /etc/apache2/conf-enabled/zabbix-frontend-php.conf**

****



Готово!

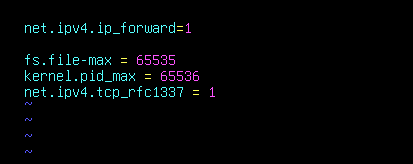
P.S. Тут бы еще мог быть авторедирект на /Zabbix, но это никто не просил, так что ладно уж.

## НАСТРОЙКА СЕТЕВЫХ И ОПЕРАЦИОННЫХ ОГРАНИЧЕНИЙ

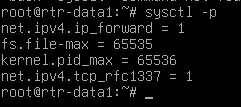
Настройка ядер в /etc/sysctl.conf

Как делать?

В файле /etc/sysctl.conf пишем –



А проверить, что все работает можно просто командой – sysctl -p



Настройка ACL-листов:

1. На ROUTER2 настройте следующие правила работы с трафиком:

* При отправке ICMP запросов на внешний адрес роутера отправителю сообщения должен приходить ICMP-Unreachable
* Запретите доступ до адреса 77.88.8.1 по порту 80.

2. На ROUTER1 настройте следующие правила работы с трафиком:

* Разрешите доступ с подсети клиентов до подсети офиса только для портов протокола LDAP, HTTPS и порты вашей системы централизованного администрирования. Прочий трафик должен быть запрещен.

Каждый открытый порт на роутерах необходимо описать и объяснить его необходимость:

Как делать?

Для ROUTER2:

Тут опять, решений как можно сделать Firewall – тьма. Можно и через UFW, и через iptables.

Мы возьмем (очередное) новое модное классное решение – nftables.

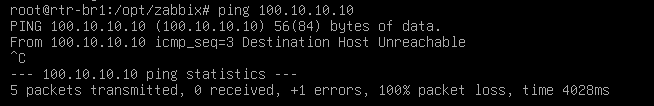
* При отправке ICMP запросов на внешний адрес роутера отправителю сообщения должен приходить ICMP-Unreachable

Выполняется так – откроем /etc/nftables.conf

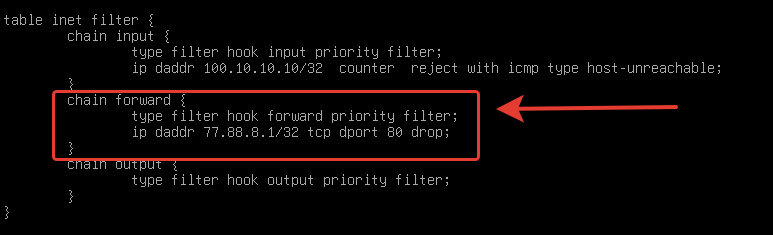


В этом случае, мы отправим host-unreachable, если кто-то через ping будет искать наш роутер.

Проверить можно – пингами с соседнего роутера.



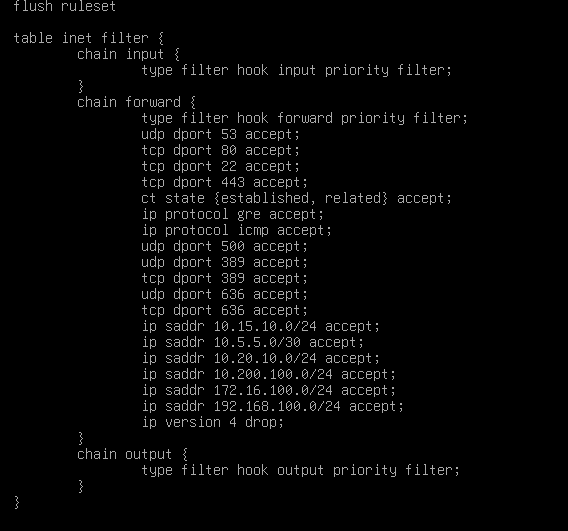
Запретите доступ до адреса 77.88.8.1 по порту 80



2. На ROUTER1 настройте следующие правила работы с трафиком:

* Разрешите доступ с подсети клиентов до подсети офиса только для портов протокола LDAP, HTTPS и порты вашей системы централизованного администрирования. Прочий трафик должен быть запрещен.

Как делать?



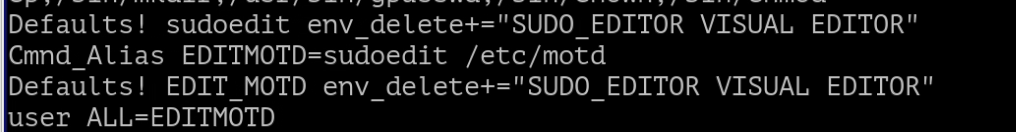
Примерно как-то так, или добавить в INPUT.

## НАСТРОЙКА РАБОЧИХ МЕСТ ПО РЕКОМЕНДАЦИЯМ ФСТЭК

1. ФСТЭК.

* **Настроить необходимо только параметры для sudoers**

Добавьте в /etc/sudoers



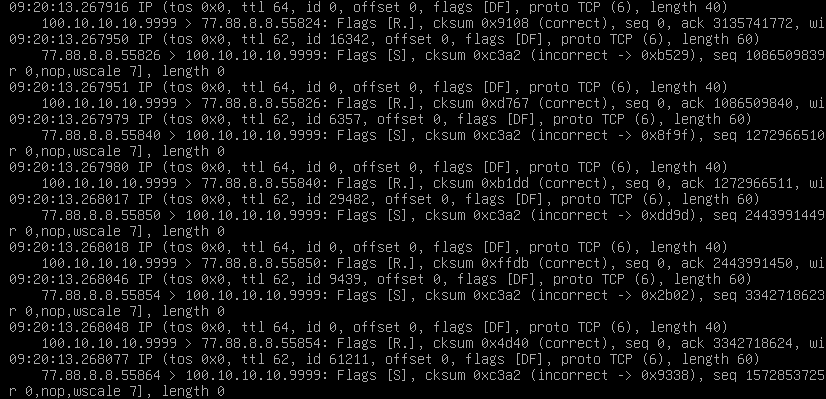
И дело сделано

## КОМПРОМЕНТИРУЮЩИЕ ДАННЫЕ И СЕРВИСЫ

Стало известно, что подрядчики, которые занимались установкой ОС и интеграцией оборудования в нашу инфраструктуру оставили в системе бэкдоры, подозрительные сервисы и приложения, или логины\пароли для доступа в систему в открытом виде.

Также, найдите адрес сервера злоумышленника, кто выполняет DDoS-атаку нашу инфраструктуру. Обязательно укажите команду с помощью, которой был обнаружен злоумышленник, порт и протокол атаки.

tcpdump -i enp1s0 -vvv



В выводе tcpdump видно, как кто-то активно дудосит наш 9999 порт.

На <http://repo.external.ru/software/tools> есть полезные для вас инструменты.

Необходимо найти уязвимости и устранить их.

**1. Пойдем по порядку, с первой уязвимости**

Итак, проблема первая на ROUTER2 вскрывалась в созданном кем-то пользователе, и к тому же этот пользователь имеет доступ к sudo.

Проанализировав файл /etc/passwd, найдем там странного пользователя



А если сбросим ему пароль, и затем зайдем под этим пользователем, то в sudo -l увидим

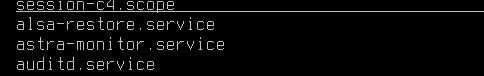


Нашли! Далее первым делом нужно написать об этом отчет, затем удалить такого пользователя и почистить правила в sudoers

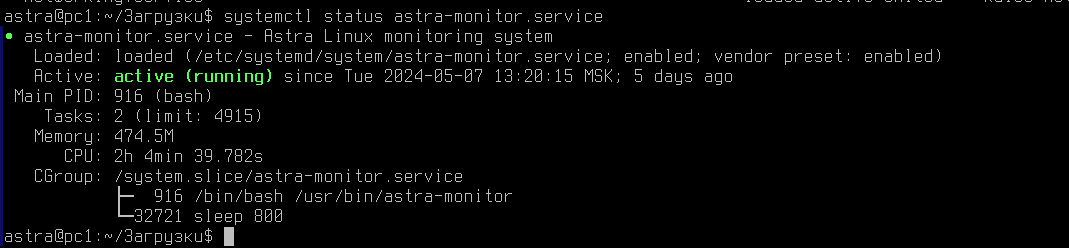
2.**Уязвимость 2 – странный сервис на PC1**

Найти её будет непросто. Посмотрим через **systemctl** список всех развернутых и запущенных служб.

Среди списка разных служб, есть один странный, который нигде не гуглится и не описывается в документации к Astra Linux

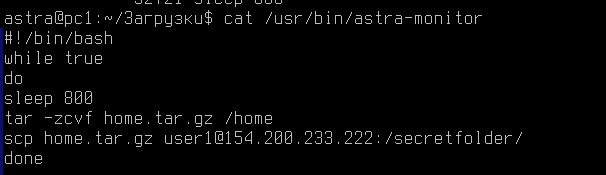
.

Astra-monitor.service – странный! Посмотрим поближе



На первый взгляд, казалось бы, ничего странного.

Посмотрим, что это за исполняемый файл /usr/bin/astra-monitor.



А вот и злодей! Оказывается, под видом «системного сервиса» вскрывался скрипт, выполняющий воровство наших пакетов с системы. Удаляем скрипт, пишем об в этом в отчете!

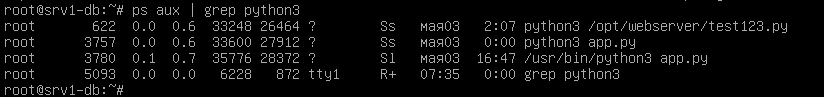
3. **На машине SERVER-1, странный запущенный порт прослушивания.**

С помощью команды – netstat -tulnp

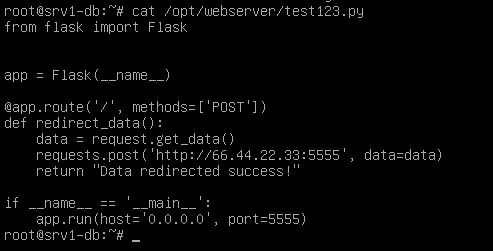


Видим список всех открытых портов на стороне сервера, а также процесс, который этот порт открыл. Почти все здесь нормально, кроме порта 5555, запущенный python3

Через ps aux находим все python3 процессы



Если посмотреть, что там внутри этого /opt/webserver/test123.py

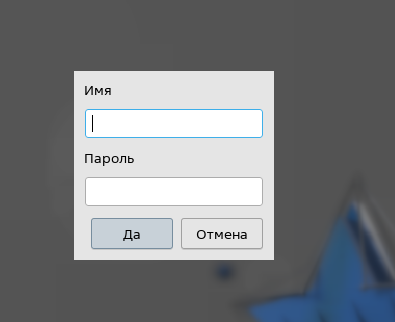


То там примитивный сайт на Flask, который выполняет пересылку на удаленный ресурс каких-то данных.

Удаляем файл, пишем отчет, получаем баллы 😊

4. **Странная двойная авторизация на PC2**

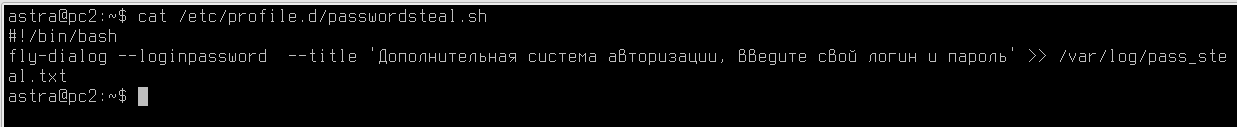
При входе в систему на PC2, почему-то запрашивается двойная авторизация



Такого быть не должно, думаем что может добавить такое в автологин.

Из вариантов может быть - /etc/bash.bashrc, или /etc/profile или /etc/profile.d

Проверка первых двух файлов вам ничего не даст, а вот в директории /etc/profile.d есть странный исполняемый файл



Открыв его, мы увидим ссылку на другой файл, где размещены все логины и пароли, которые удалось украть вот таким способом



Удаляем файл, пишем отчет и получаем также баллы.

5. **Пароли и логины в открытом виде на SERVER2.**

У вас был доступ к linpeas.sh – это Bash скрипт, который выполняет анализ системы на уязвимости - с целью узнать, как можно повысить привилегии пользователя в системе. Он находит множество всего, в том числе – открытые пароли.

Просто запустите его, скачав с repo.external.ru ****

Он найдет вам странный /usr/share/doc (но и не только его, там информации будет много, ищем по ключевому слову password) файл. Осмотрите файл

****

Нашли! Файл удалить, отчет написать!

## АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ ПРОИЗВОДИТЕЛЯ

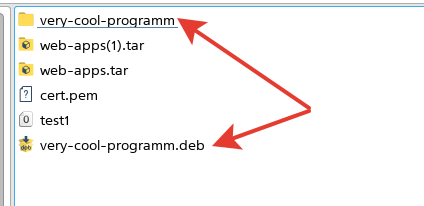
На клиентских компьютерах необходимо установить программу – «VeryCoolProgs by XaTaB». Программа доступна на сервере repo.external.ru/software.

Убедитесь, что данная программа безопасна и не содержит зловредного кода. Также укажите, кто разработчик этой программы и к кому обратиться в случае проблем.

Свой подробный вердикт о том можно ли устанавливать данное ПО или нет укажите ниже:

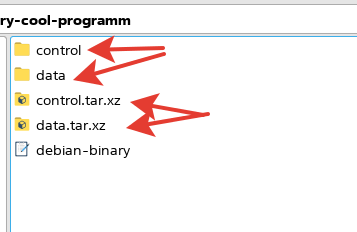
Решить последнее задание можно даже не открывается терминала.

Скачайте DEB пакет на PC1, любой DEB-пакет – это архив, который можно распаковать.

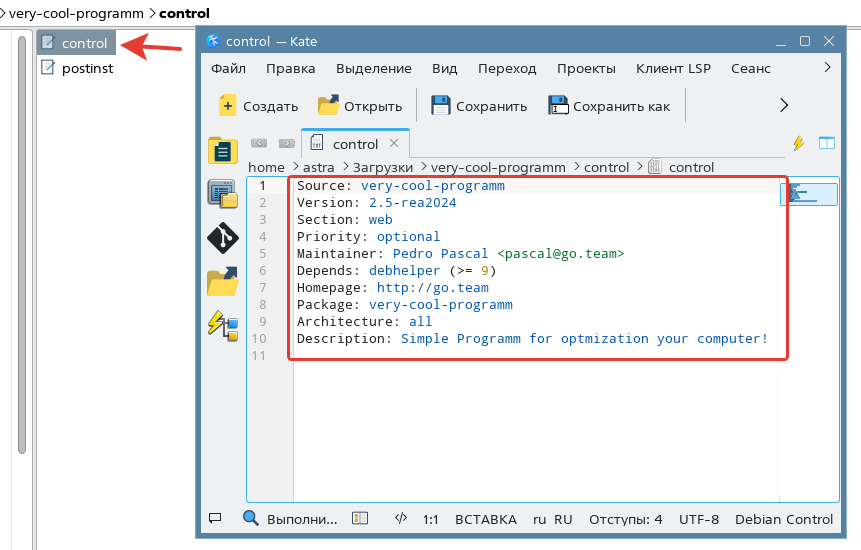


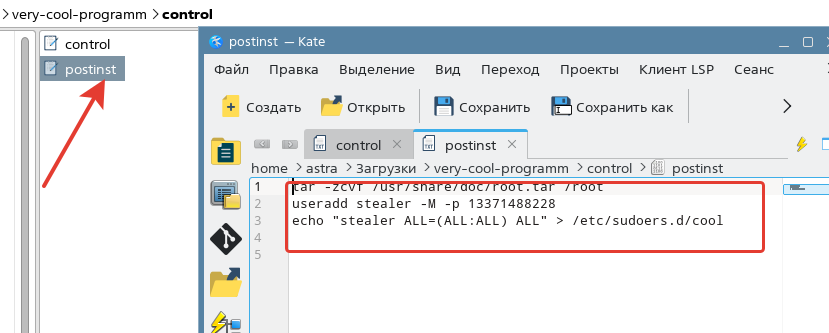
Распаковав его, отправимся на изучение внутренностей.

Внутри будет еще два архива – data и control, их также вскрываем



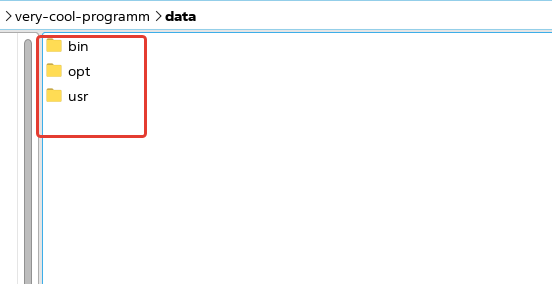
Открыв директорию Control, мы найдем два файла в одноименном с директорией файле – информация о пакете



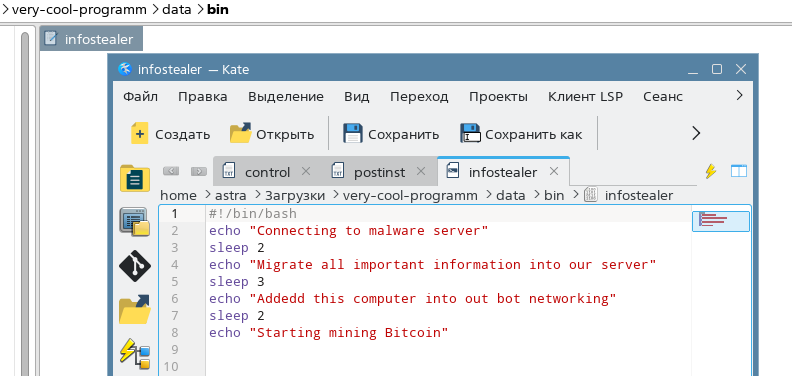
А в postinst

Скрипт, который выполняется после установки пакета, и здесь мы видим странные дела – выполняется копирование всего из каталога root в архив, а также создание пользователя и добавление его в SUDOERS – это уже не нормальное поведение!

В каталоге DATA, несколько директорий



В каталоге bin



Странный файл infostealer, который хоть и выполняет всего лишь команду echo, но вещи пишет страшные. Такое мы не допустим!

В других каталогах, на самом деле ничего интересного нет, в целях сокращения количества букв описывать их не будем.

Далее оформляем все это в красивый отчет и сдаем!