

## Лабораторная работа

Цель: овладеть основными практическими навыками фильтрации трафика в сети с помощью списков контроля доступа (ACL), в программе Cisco Packet Tracer.

### Теоретические сведения

#### Введение

ACL (Access Control List) — это набор текстовых выражений, которые что-то разрешают, либо что-то запрещают. Обычно ACL разрешает или запрещает IP-пакеты, но помимо всего прочего он может заглядывать внутрь IP-пакета, просматривать тип пакета, TCP и UDP порты. Также ACL существует для различных сетевых протоколов (IP, IPX, AppleTalk и так далее). В основном применение списков доступа рассматривают с точки зрения пакетной фильтрации, то есть пакетная фильтрация необходима в тех ситуациях, когда у вас стоит оборудование на границе Интернет и вашей частной сети и нужно отфильтровать ненужный трафик.

Вы размещаете ACL на входящем направлении и блокируете избыточные виды трафика.

Итак, ACL (access control list) — это строго говоря, механизм для выбора из всего потока трафика какой-то части, по заданным критериям. Например, через маршрутизатор проходит множество пакетов, и вот такой ACL выбирает из множества только те пакеты, которые идут из подсети 192.168.1.0/24:

```
access-list 1 permit 192.168.1.0
```

Что дальше делать с этим трафиком — пока неизвестно. Например, трафик, попавший под ACL может заворачиваться в VPN тоннель, или, подвергаться трансляции адресов (NAT). В курсе CCNA рассматривается два способа использования ACL: основной — это фильтрация трафика, второй — использование ACL при настройке NAT. Важно следующее: не имеет значения, где и для каких целей мы будем использовать ACL, правила написания ACL от этого не меняются. Кроме того, если мы только создали ACL, то он пока ни на что не влияет. ACL — это просто несколько неработающих строчек в конфиге, пока мы его не применим, например, на интерфейс, для фильтрации трафика.

#### Типы ACL

ACL-и бывают двух видов: стандартные и расширенные. Стандартные позволяют отфильтровывать трафик только по одному критерию: адрес отправителя, в CCNA рассматривается конкретно только ip адрес отправителя. Можно, например, поставить на выходе из нашей сети такой ACL:

```
access-list 1 permit host 192.168.10.50
```

```
access-list 1 permit host 192.168.10.53
```

```
access-list 1 permit host 192.168.10.60
```

Этот ACL будет разрешать выход в интернет только с перечисленных в нём трёх ip адресов (для такой задачи, как вы видите, нам хватило стандартного ACL).

Расширенный ACL позволяет фильтровать трафик по большому количеству параметров:

1. Адрес отправителя
2. Адрес получателя
3. TCP/UDP порт отправителя
4. TCP/UDP порт получателя
5. Протоколу, завёрнутому в ip (отфильтровать только tcp, только udp, только icmp, только gre и т.п.)
6. Типу трафика для данного протокола (например, для icmp отфильтровать только icmp-reply).
7. Отделить TCP трафик, идущий в рамках установленной TCP сессии от TCP сегментов, которые только устанавливают соединение

Возможности расширенных ACL богаче стандартных, кроме того, они могут расширяться дополнительными технологиями:

- Dynamic ACL — ACL, в котором некоторые строчки до поры до времени не работают, но когда администратор подключается к маршрутизатору по telnet-у, эти строчки включаются, то есть администратор может оставить для себя «дыру» в безопасности для отладки или выхода в сеть.
- Reflexive ACL — зеркальные списки контроля доступа, позволяют запоминать, кто обращался из нашей сети наружу (с каких адресов, с каких портов, на какие адреса, на какие порты) и автоматически формировать зеркальный ACL, который будет пропускать обратный трафик извне вовнутрь только в том случае, если изнутри было обращение к данному ресурсу.
- TimeBased ACL, как видно из названия, это ACL, у которых некоторые строчки срабатывают только в какое-то время. Например, с помощью таких ACL легко настроить, чтобы в офисе доступ в интернет был только в рабочее время.

Все ACL (и стандартные, и расширенные) можно задавать по-разному: именованным и нумерованным способом. Первый предпочтительнее, так как позволяет затем редактировать ACL, в случае же использования нумерованного способа, ACL можно только удалить целиком и заново создать, либо дописать очередную строчку в конец.

### **Порядок просмотра ACL**

Итак, что из себя представляет ACL и как трафик проверяется на соответствие.

ACL — это набор правил. Каждое правило состоит из действия (permit, deny) и критерия (для стандартных ACL — ip адрес отправителя, для расширенных — множество критериев). Рассмотрим такой пример стандартного нумерованного ACL:

```
access-list 1 permit host 192.168.1.1
access-list 1 deny 192.168.1.0
access-list 1 permit any
```

Этот ACL запрещает доступ для всей сети 192.168.1.0/24 кроме хоста 192.168.1.1 и разрешает доступ для всех остальных сетей. Как проверяется трафик на соответствие ACL? Построчно. То есть, приходит, например, пакет с адреса 192.168.2.2 на роутер, а на том интерфейсе через который он пришел стоит на вход указанный выше ACL, вот построчно Ip адрес отправителя сверяется с данным ACL, что важно — до первого совпадения. Как только пакет совпадёт с какой-то из строк, сработает действие (permit — пропустить пакет либо deny — уничтожить пакет) и дальше никаких проверок по оставшимся строчкам проводиться не будет. Если все строчки пройдены, а пакет так и не попал ни под одно из правил, то он по умолчанию уничтожается. В нашем случае, в примере выше любой пакет подходит под третью строчку, так как там вместо адреса стоит слово «any», означающее, что любой адрес подойдёт. Таким образом, приведённый ACL можно читать так:

- Если пакет пришёл с адреса 192.168.1.1, то его надо сразу же пропустить и не делать больше никаких проверок в этом ACL;
- В противном случае, если пакет пришел из сети 192.168.1.0 (кроме адреса 192.168.1.1, с которым мы уже разобрались строчкой выше), то пакет надо уничтожить и опять же, на этом закончить просмотр ACL, не переходить к следующему шагу;
- Если пакет не попал под первые два правила. То есть он не с адреса 192.168.1.1, да и вообще, не из сети 192.168.1.0, то он всегда попадает под правило permit any, то есть, пакет надо пропустить дальше — пусть идёт.

Очень важно понимать приведённый выше порядок просмотра строк в ACL, он един для всех типов ACL (не только для стандартного). Кроме того, из этого порядка следует очевидное правило: «В ACL должны идти наиболее специфичные, узкие, точные строчки вначале и наиболее абстрактные, общие — в конце».

### Применение ACL

ACL применяется для разных целей, но основная цель, для которой он используется в CCNA — фильтрация трафика на интерфейсе. Для этого надо сначала создать стандартный или расширенный ACL. Если ACL именованный, то у него есть имя, которое мы и укажем на интерфейсе, если нумерованный — то номер. Чтобы сделать это, заходим на интерфейс и пишем команду *ip access-group*, например, так:

```

R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface FastEthernet0/0
R1(config-if)#ip access-group MY_ACLS_NAME in

```

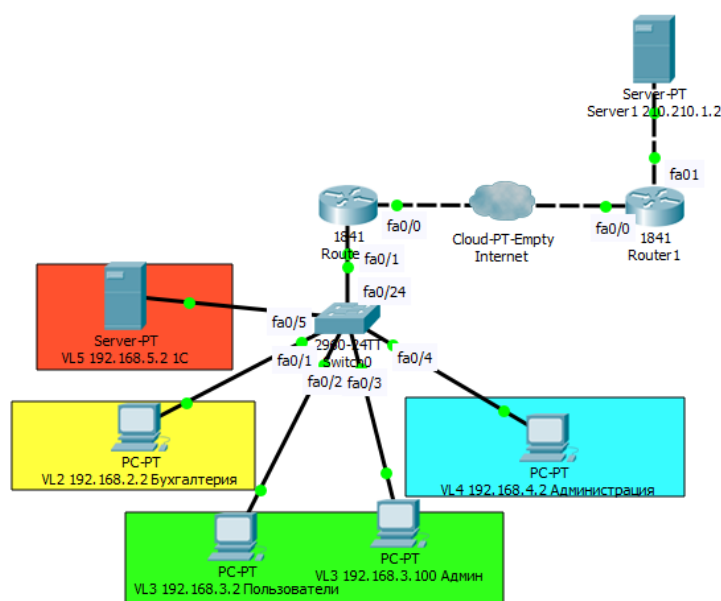
В этом примере мы применили ACL с именем MY\_ACLS\_NAME на интерфейсе Fa0/0 на весь входящий трафик (о чем говорит слово in) если бы мы не писали out — то фильтровался бы исходящий трафик.

Люди часто путаются с направлениями. Например, есть сеть, подключенная к маршрутизатору и стоит задача запретить входящий в эту сеть трафик. Так вот, в данном случае этот входящий трафик фильтруется применением ACL на out, то есть на выход. Всё просто, чтобы не запутаться, надо представить себя на месте маршрутизатора. Понятно, что если трафик входит в какую-то сеть, то он при том выходит из маршрутизатора и с точки зрения роутера, такой трафик исходящий.

### Практическое задание

Данная лабораторная работа может быть выполнена на реальном оборудовании или в Cisco Packet Tracer. Все необходимые действия указаны по порядку их выполнения. Для начала выполнения лабораторной работы необходимо открыть файл

Топология сети



На данной схеме у нас имеются 4 сегмента сети:

VL2 – сегмент бухгалтерии

VL3 – сегмент пользователей

VL4 – сегмент администрации

VL5 – сегмент сервера 1С

Так же изначально у нас уже установлена связь с нашими устройствами через Switch0.

Сперва настроим NAT. Для этого нам необходимо определить интерфейсы, которые будут являться внешними и внутренними для NATa.

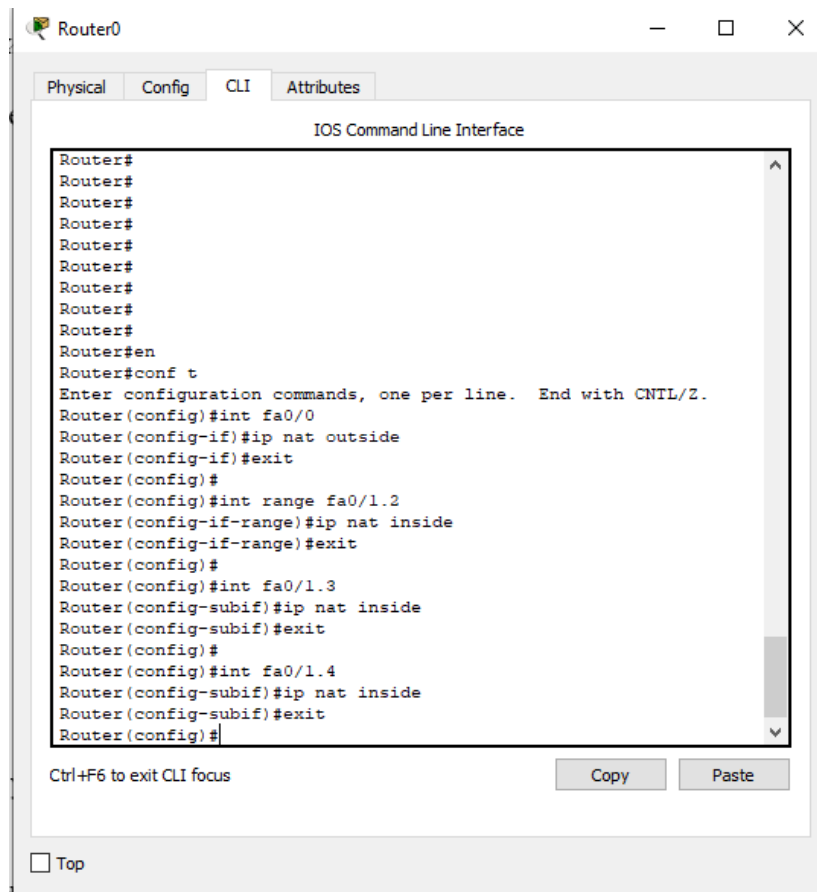
Зайдем на Router0 и во вкладке CLI пропишем следующие значения

```
en
conf t
int fa0/0
ip nat outside
exit

int range fa0/1.2
ip nat inside
exit

int fa0/1.3
ip nat inside
exit

int fa0/1.4
ip nat inside
exit
```



Мы специально не указали VL5, сеть сервера 1С, что бы у него, в целях безопасности, не было доступа в интернет.

Теперь можем создать Access List, для этого необходимо прописать следующие команды.

```
ip access-list standard FOR-NAT
```

```
permit 192.168.2.0 0.0.0.255
```

```
permit 192.168.3.0 0.0.0.255
```

```
permit 192.168.4.0 0.0.0.255
```

```
ip nat inside source list FOR-NAT interface fa0/0 overload
```

```
end
```

```
Router(config)#ip access-list standard FOR-NAT
Router(config-std-nacl)#permit 192.168.2.0 0.0.0.255
Router(config-std-nacl)#permit 192.168.3.0 0.0.0.255
Router(config-std-nacl)#permit 192.168.4.0 0.0.0.255
Router(config-std-nacl)#
Router(config-std-nacl)#ip nat inside source list FOR-NAT
interface fa0/0 overload
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

trl+F6 to exit CLI focus

Copy

Paste

Top

После настройки access list проверим ping с любого настроенного устройства. Пусть это будет компьютер бухгалтерии. Зайдем в него, во вкладку Desktop и выберем Command Prompt. Введем туда следующую команду

*ping 210.210.1.2*

Как можно увидеть на скриншоте ниже, сервер с интернетом успешно «пингуется», что говорит нам о правильной настройке

```
C:\>ping 210.210.1.2

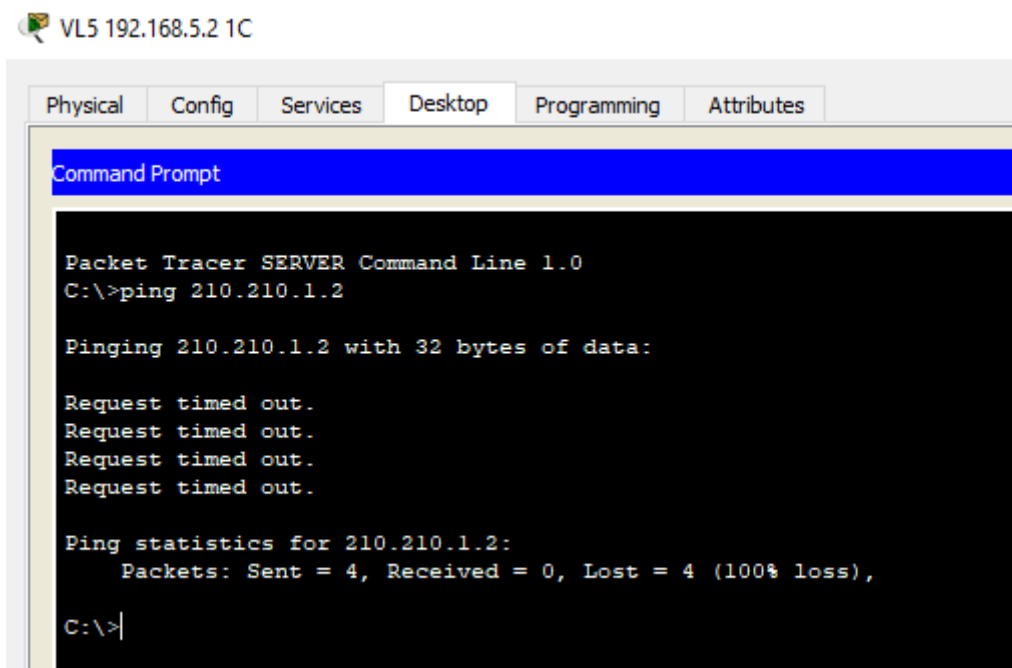
Pinging 210.210.1.2 with 32 bytes of data:

Reply from 210.210.1.2: bytes=32 time<1ms TTL=126
Reply from 210.210.1.2: bytes=32 time=1ms TTL=126
Reply from 210.210.1.2: bytes=32 time<1ms TTL=126
Reply from 210.210.1.2: bytes=32 time=1ms TTL=126

Ping statistics for 210.210.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>|
```

*Теперь проверим пинг с сервера 1С. Для этого выполним вышеописанные действия аналогично.*



Как видно, доступа нет, как раз потому что мы не включили данное устройство в Access List.

Далее, нам необходимо обеспечить безопасность нашей локальной сети от возможных атак из сети интернет. Для этого будем использовать расширенные Access List и применим их на входящий трафик, что бы не было лишней загрузки маршрутизатора. В качестве ip-адреса источника укажем любой, а в качестве получателя конкретно наши сети.

Для этого зайдём на Router0 и выполним следующие команды.

```
en
conf t
ip access-list extended FROM-OUTSIDE
deny ip any 192.168.2.0 0.0.0.255
deny ip any 192.168.3.0 0.0.0.255
deny ip any 192.168.4.0 0.0.0.255
deny ip any 192.168.5.0 0.0.0.255
exit

int fa0/0
ip access-group FROM-OUTSIDE in
end
```



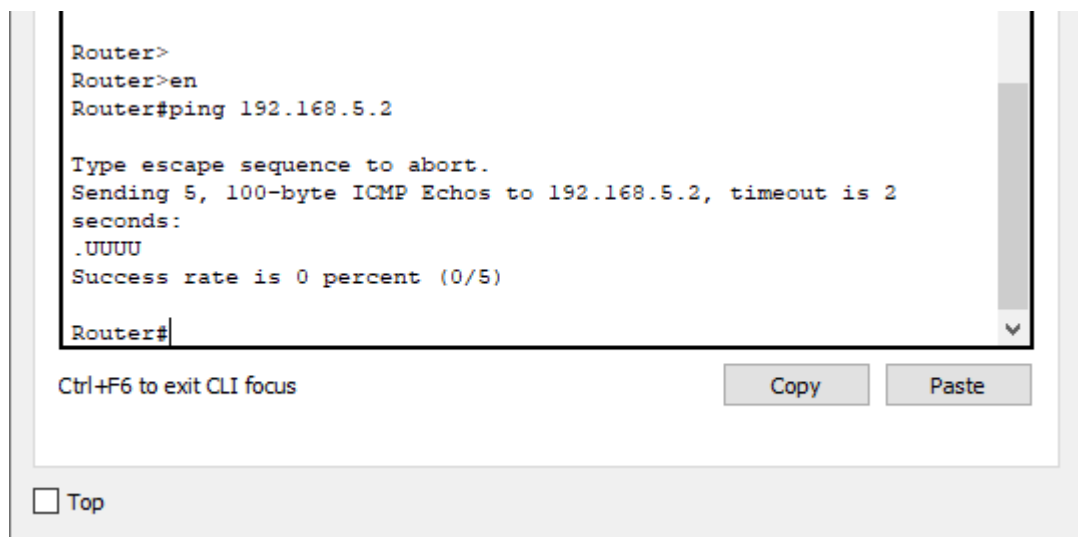
```

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list extended FROM-OUTSIDE
Router(config-ext-nacl)#deny ip any 192.168.2.0 0.0.0.255
Router(config-ext-nacl)#deny ip any 192.168.3.0 0.0.0.255
Router(config-ext-nacl)#deny ip any 192.168.4.0 0.0.0.255
Router(config-ext-nacl)#deny ip any 192.168.5.0 0.0.0.255
Router(config-ext-nacl)#exit
Router(config)#
Router(config)#int fa0/0
Router(config-if)#ip access-group FROM-OUTSIDE in
Router(config-if)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

Проверим пинг с Router1 в локальную сеть командой. Видно, что не один пакет не прошёл

*ping 192.168.5.2*



```

Router>
Router>en
Router#ping 192.168.5.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2
seconds:
.UUUU
Success rate is 0 percent (0/5)

Router#

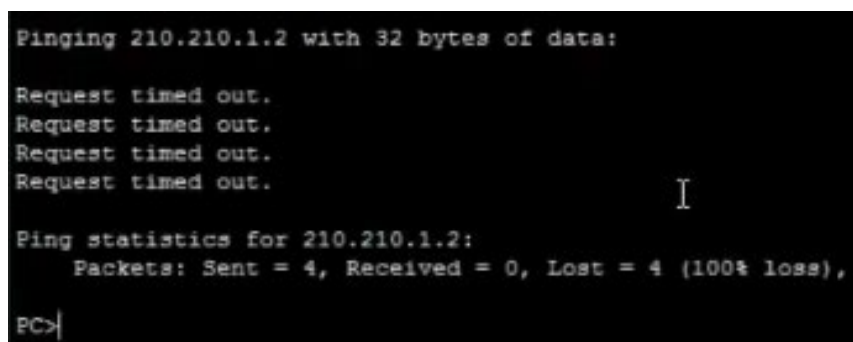
```

Ctrl+F6 to exit CLI focus

Copy Paste

☐ Top

Но если мы проверим доступ к интернету с любой сети, то увидим что он тоже пропал.



```

Pinging 210.210.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 210.210.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>

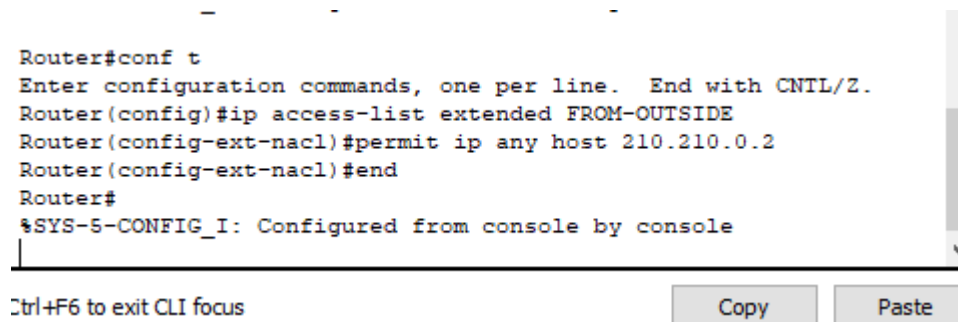
```

Для восстановление доступа к интернету, необходимо написать разрешающее правило, которое допустит исходящий трафик на внешний интерфейс.

Для этого в Router0 пропишем следующие команды.

```
en
conf t
ip access-list extended FROM-OUTSIDE
permit ip any host 210.210.0.2
end

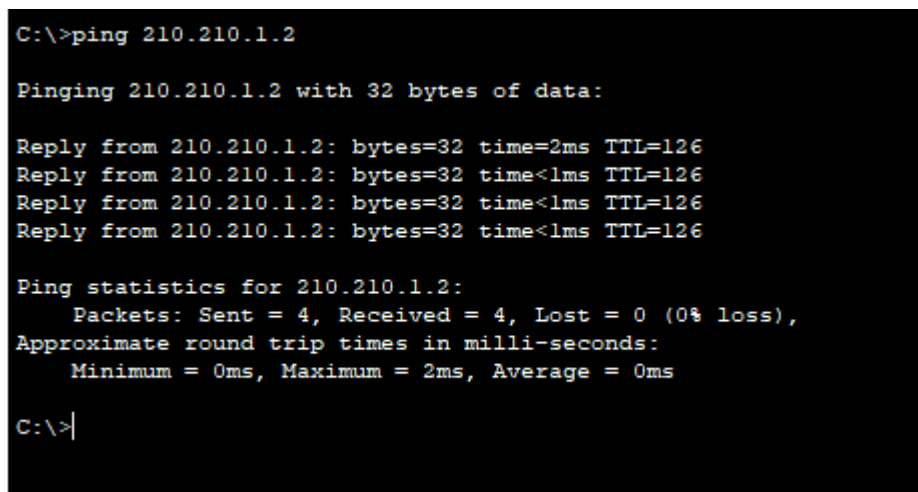
wr mem
```



```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list extended FROM-OUTSIDE
Router(config-ext-nacl)#permit ip any host 210.210.0.2
Router(config-ext-nacl)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Ctrl+F6 to exit CLI focus
```

Теперь выполнив проверку, мы увидим, что интернет на устройствах у нас снова появился



```
C:\>ping 210.210.1.2

Pinging 210.210.1.2 with 32 bytes of data:

Reply from 210.210.1.2: bytes=32 time=2ms TTL=126
Reply from 210.210.1.2: bytes=32 time<1ms TTL=126
Reply from 210.210.1.2: bytes=32 time<1ms TTL=126
Reply from 210.210.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 210.210.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>|
```

Далее настроим на Router0 доступ по telnet. Для этого выполним следующие команды.

```
en
conf t
username admin privilege 15 password cisco
enable password cisco
line vty 0 4
login LOCAL
end

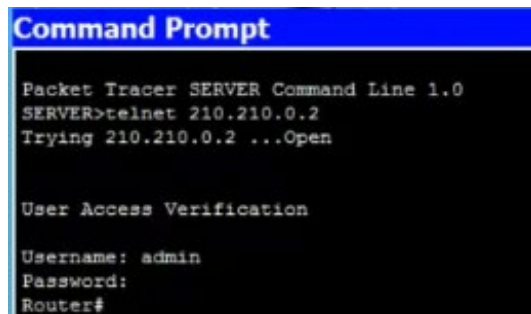
wr mem
```

```

Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#username admin privilege 15 password cisco
Router(config)#enable password cisco
Router(config)#line vty 0 4
Router(config-line)#login LOCAL
Router(config-line)#

```

Проверив доступ к telnet с публичного сервера, то мы увидим, что доступ разрешен, хоть защищен паролем, но злоумышленник может воспользоваться программой для перебора паролей, что сведет защиту к нулю. Поэтому воспользуемся access list для ограничения доступа по telnet из внешней сети



```

Command Prompt

Packet Tracer SERVER Command Line 1.0
SERVER>telnet 210.210.0.2
Trying 210.210.0.2 ...Open

User Access Verification

Username: admin
Password:
Router#

```

Выполним на Router0 следующие команды.

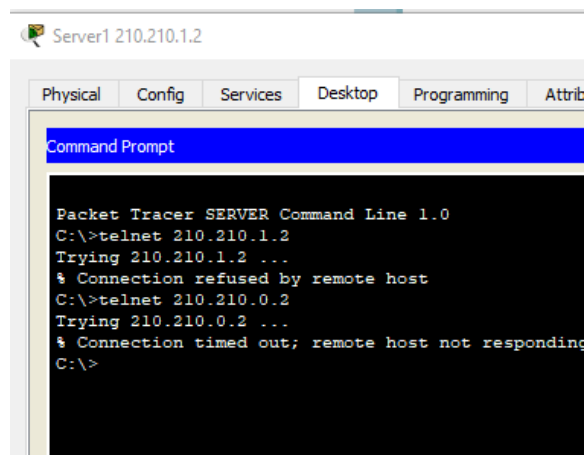
```

conf t
no ip access-list extended FROM-OUTSIDE
ip access-list extended FROM-OUTSIDE
deny tcp any host 210.210.0.2 eq telnet
permit ip any host 210.210.0.2

```

Здесь мы сперва удалили все Access list, и создали заново, но изначально ввели команду ограничивающую доступ к роутеру через *telnet*, а уже потом задали основное правило. Сделано это было для того, потому что в Access List работают сверху-вниз и трафик попадающий на маршрутизатор проверяется сначала первым правилом, потом вторым и так далее, в нашем же случае. У нас был разрешен любой внешний трафик и поэтому 2 правило даже не проверялось.

Проверив снова, мы убедимся, что доступ через telnet из внешней сети – запрещен.



```

Server1 210.210.1.2

Physical Config Services Desktop Programming Attributes

Command Prompt

Packet Tracer SERVER Command Line 1.0
C:\>telnet 210.210.1.2
Trying 210.210.1.2 ...
% Connection refused by remote host
C:\>telnet 210.210.0.2
Trying 210.210.0.2 ...
% Connection timed out; remote host not responding
C:\>

```

Теперь давай рассмотрим стандартные Access List. У нас имеется сервер 1С, очевидно, что к нему должен иметь доступ только сегмент сети «Бухгалтерия», а остальные не должны.

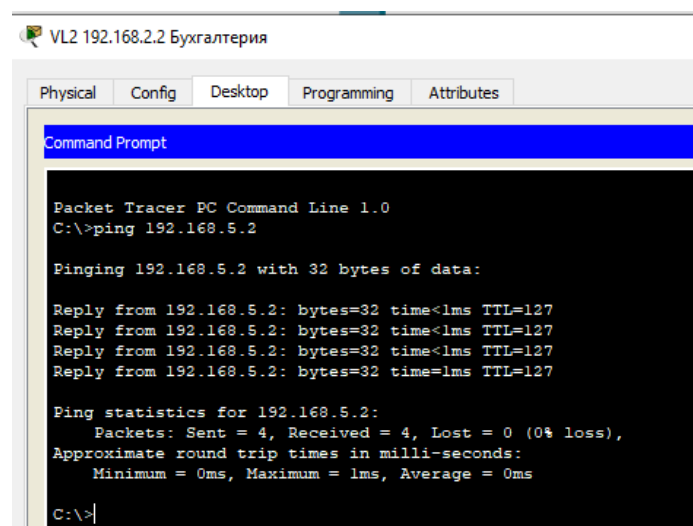
Для этого на Router0 пропишем следующие команды

```
conf t
ip access-list standart TO-1C
permit 192.168.2.0. 0.0.0.255
exit
```

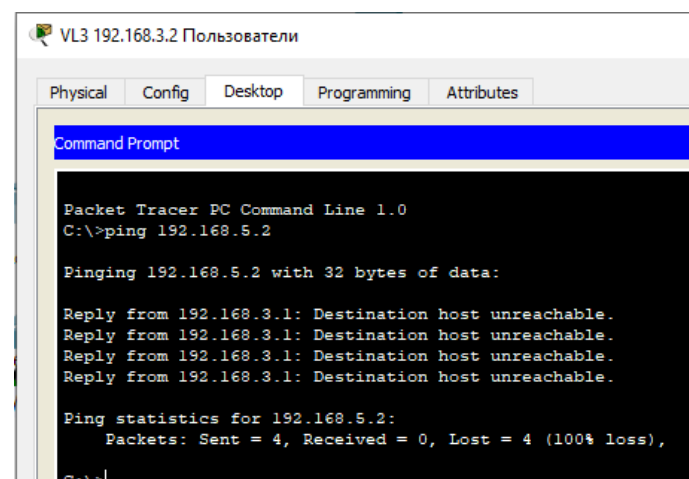
```
int fa0/1.5
ip access-group TO-1C out
end
```

```
wr mem
```

Выполним проверку с компьютера бухгалтерии командой ping, видно что ping проходит.



С любого другого компьютера не будет доступа



Вывод: в данной лабораторной работе, мы познакомились с Access List и настроили контроль доступа трафика в нашей сети.

### **Контрольные вопросы**

1. Сформируйте определение ACL.
2. Что такое NAT?
3. По каким параметрам ACL может фильтровать трафик?