# ACLcheck — утилита для анализа списков доступа сетевого оборудования Cisco

# Руководство пользователя

**Оглавление**



## Описание функционала

Если Вы не раз сталкивались с большими списками доступа или входящими в них object-группами, то наверняка задавались вопросом, существует ли инструмент, позволяющий определить, пропустит ли access-лист определённый пакет и какие строки сработают.

Конечно, такие инструменты существуют и полностью или частично решают перечисленные задачи. Однако, они, как правило, являются частью мощных «комбайнов» управления сетью, 90% функционала которых обычно не используется.

Безусловно, никто не запрещает использовать регулярные выражения для поиска определённых строк списка доступа прямо с консоли сетевого устройства. Но данный метод предоставит очень поверхностный результат. Например, он не отобразит доступ хоста, попадающего в сетевую маску или порт, попадающий под диапазон. Тем более, таким образом нельзя отобразить все существующие доступы между двумя заданными узлами или сетями. Опытный сетевой администратор осведомлён о безрезультативности метода простого парсинга access-листа для таких задач.

Рассматриваемая небольшая утилита создана именно для этого — найти строки access-листа, разрешающие или запрещающие определённый сетевой трафик, и даже более — выявить все строки, имеющие отношение к доступам между между заданными точками.

Идея простая: программа находит строки access-листа, удовлетворяющие заданному критерию. Сам критерий выглядит как строка access-листа, но без использования оператора «permit» или «deny».

Если регулярно добавлять сетевые правила в access-лист без проверки их существования, то списки доступа станут содержать большое количество избыточных правил. Для решения этой проблемы в программе реализован функционал анализа списка доступа на избыточность. С его помощью можно выявить лишние правила и высвободить ресурсы оборудования.

При использовании в ACL object-групп программе необходимо передать (скопировать) их состав. В итоговом ACL такие группы будут представлены элементарными правилами.

## Интерфейс программы


Рис.1 Главное окно

На рисунке 1 представлено главное окно программы со следующими элементами:

1 — поле ввода access-листа

2 — поле ввода object-групп

3 — запуск распознавания access-листа

4 — поле вывода распознанного access-листа

5 — однострочное поле ввода условия

6 — многострочный список ввода условий

7 — прямая проверка условия

8 — обратная проверка условия

9 — поле результата проверки

10 — шкала позиционирования сработавших строк ACL

11 — поле просмотра деталей сработавших строк ACL

12 — анализ ACL на конфликты и избыточность

13 — упорядочивание строк ACL по различным критериям

14 — указатель текущего активного условия в многострочном списке (6)

15 — поле краткого обозначения результатов проверки списка условий (6)

16 — переключатель типа маски

17 — переключатель режима проверки адресов источника и назначения

18 — игнорирование строк ACL с ICMP в режиме частичного совпадения адресов

19 — выбор вариантов использования имени ACL в командах CLI

20 — вывод object-групп, используемых в распознанном ACL

21 — вывод поддерживаемых протоколов, а также типов и кодов ICMP

22 — поле вывода ошибок распознавания ACL

23 — переключатель анализа отдельных портов из набора

24 — переключатель анализа частично перекрывающихся диапазонов портов

25 — переключатель режима проверки по неполному набору портов

26 — вывод зеркального ACL (source и destination меняются местами)

# Основные шаги

Исходный access-list необходимо скопировать в поле 1. Если он содержит object-группы, то их состав необходимо скопировать в поле 2. ACL и object-группы можно копировать как с конфигурации устройства (“show running-config”, “show startup-config”), так и с результата команд “show access-lists”, “show object”.

Ниже приведён пример результата команды “show running-config”, допустимого для использования в поле 1:

```
ip access-list extended ACL
permit icmp host 172.16.0.6 host 172.21.0.6
permit ip host 172.16.0.6 host 172.21.0.1
permit tcp host 192.168.8.15 range 1024 65534 host 192.168.66.47
permit tcp 192.168.8.0 0.0.0.255 eq 22 1521 3389 addrgroup ADMIN_BSD
permit tcp host 192.168.8.12 eq 1521 192.168.83.20 0.0.0.1
```

Тот же access-list по команде “show access-lists”:

```
Extended IP access list ACL
10 permit icmp host 172.16.0.6 host 172.21.0.6
20 permit ip host 172.16.0.6 host 172.21.0.1 (32 matches)
30 permit tcp host 192.168.8.15 range 1024 65534 host 192.168.66.47
40 permit tcp 192.168.8.0 0.0.0.255 eq 22 1521 3389 addrgroup ADMIN_BSD (1 match)
50 permit tcp host 192.168.8.12 eq 1521 192.168.83.20 0.0.0.1
```

Поддерживаются также маски, выраженные длиной префикса (CIDR записи):

```
ip access-list ACL
permit icmp 172.16.0.6/32 172.21.0.6/32
permit ip 172.16.0.6/32 172.21.0.1/32
permit tcp 192.168.8.15/32 range 1024 65534 192.168.66.47/32
permit tcp 192.168.8.0/24 eq 22 1521 3389 addrgroup ADMIN_BSD
permit tcp 192.168.8.12/32 eq 1521 192.168.83.20/31
```

Пример результата команды “show running-config”, допустимого для использования в поле 2:

```
object-group ip address ADMIN_BSD
host-info 10.237.92.131
host-info 10.22.145.132
host-info 10.22.145.136
host-info 10.22.145.141
```

Содержимое вывода команды “show object-group”:

```
IP address object group ADMIN_BSD
host 10.237.92.131
host 10.22.145.132
host 10.22.145.136
host 10.22.145.141
```

Также допустимы и другие форматы object-групп.

Пример допустимого фрагмента команды “show running-config”:

```
object-group network Servers
host 10.15.12.5
host 10.15.5.11
host 10.15.4.2
host 10.15.7.34

object-group service Ports1
tcp-udp eq domain
tcp-udp eq 88
udp range 3268 3269
tcp gt 49151
```

Пример того же фрагмента команды “show object-group”:

```
Network object group Servers
host 10.15.12.5
host 10.15.5.11
host 10.15.4.2
host 10.15.7.34

Service object group Ports1
tcp-udp eq domain
```

```
tcp-udp eq 88
udp range 3268 3269
tcp gt 49151
```

Поддерживаются и вложенные группы:

```
object-group network zzz
5.5.5.0 255.255.255.0
host 6.6.6.6

object-group network yyy
host 3.3.3.3
group-object zzz

object-group network xxx
host 1.1.1.1
group-object zzz
group-object yyy

permit udp any object-group xxx eq 21
permit tcp object-group xxx host 7.7.7.7 eq ftp
permit tcp object-group xxx object-group xxx eq 22
permit tcp object-group yyy object-group xxx eq ftp
```

После копирования ACL и object-групп необходимо нажать кнопку 3. В результате access-list будет распознан и отображён в развёрнутом виде (в случае использования object-групп) в поле 4. Если на этапе распознавания возникли ошибки, то они будут отображены в поле 22. Результат ошибки можно скопировать в буфер обмена.
Если нарушена нумерация строк, программа предложит их перенумеровать автоматически.

Строки, полученные из object-групп, дополняются спереди цифрой «0» (рис.2).

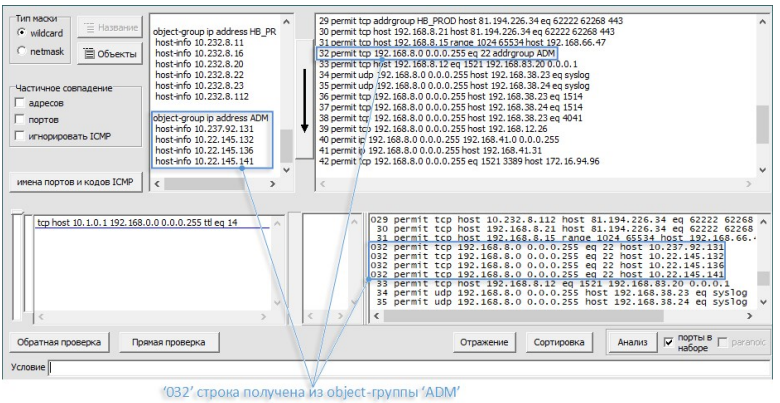



Рис.2 Строки, полученные из object-группы

Если access-лист скопирован вместе с его заголовком, то активируется кнопка 19, позволяющая использовать команды конфигурирования, содержащие имя access-листа. Посмотреть список используемых в access-листе object-групп можно кнопкой 20 (рис.3).

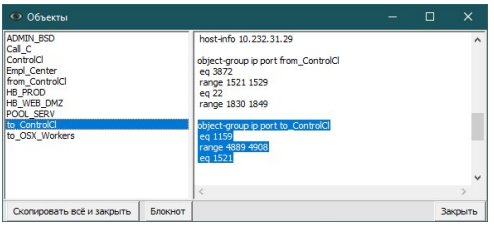



Рис.3 Список используемых object-групп

Названия используемых в access-листе объектов в упорядоченном виде отобразятся в левой части окна. Выбор соответствующей группы покажет её детали в правой части окна.

После того, как access-лист будет распознан, в поле 5 необходимо ввести условие для поиска интересуемого нас доступа и нажать кнопку 7 или клавишу «Enter». Результат поиска доступа отобразится в поле результатов проверки 9. Цвет этого поля меняется в зависимости от типа результата (разрешено, заблокировано, ошибка). При наличии разрешающего или блокирующего правила более детальная информация о нём появится в поле 11. Место расположения сработавших строк в access-листе можно определить по шкале 10. Вызов контекстного меню "Показать результат" по правой кнопке мыши на поле 9 или 11 выводит отдельное окно с подробностями результата проверки. В нём содержатся строки ACL, удовлетворяющие условию поиска. Такую же функцию выполняет двойной клик на поле 9. Данное окно выводится автоматически, если запуск проверки условия осуществляется нажатием комбинации «Alt+Enter». Закрыть это окно можно клавишей «Esc».

Если обнаружена несовместимая с адресом маска, то об этом будет выведено сообщение (рис.4). Программа предложит корректные значения для исправления адреса или маски.

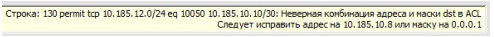



Рис.4 Несовместимые маска или адрес.

# Пример 1. Проверка существования доступа между заданными узлами по определённым портам

Предположим, нас интересует наличие доступа с хоста 192.168.1.2 по порту TCP 1521 на сервер 192.168.2.2 в следующем списке доступа:

```
ip access-list extended ACL
10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
20 permit tcp host 192.168.1.2 any
30 permit tcp host 192.168.1.3 any eq 1521
```

Копируем access-лист в поле 1 и нажимаем кнопку 3. В поле 5 вводим следующее условие:

```
tcp host 192.168.1.2 gt 1023 host 192.168.2.2 eq 1521
```

Нажимаем кнопку 7 или клавишу "Enter".

В поле 9 отобразится результат:

```
Разрешено в строке 1: 10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
Имеются ещё совпадения
```

Здесь значение “1:” является результатом сквозной нумерации строк распознанного ACL, а “10” – номер строки в исходном ACL. Надпись “Имеются ещё совпадения” означает, что в ACL присутствуют и другие строки, в которых теоретически может сработать наше условие. Результаты совпадения правил можно просмотреть в поле 11. Если на этом поле вызвать контекстное меню (правой кнопкой мыши) и выбрать пункт “Показать результат”, то появится дополнительное окно с выборкой сработавших строк ACL. Для поля 9 доступно такое же контекстное меню, либо двойной клик.

Проверку доступов можно осуществлять также набором портов. При этом будут найдены строки ACL, содержащие все порты из набора, заданного в условии. Строки, содержащие неполный набор портов, отображены не будут. Такое поведение программы можно изменить, активировав переключатель частичного совпадения портов 25 (рис.1).

Активируем переключатель 25 и введём условие в поле 5:

```
tcp host 192.168.1.3 gt 1023 host 192.168.2.2 eq 1522 1521
```

Поле 9 отобразит результат:

```
Разрешено в строке 1: 10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
Также обнаружены частичные совпадения
```

Если вызвать окно деталей результатов (например, двойным кликом поля 9), то отобразится следующий результат:

```
+ 10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
^ 30 permit tcp host 192.168.1.3 any eq 1521
```

Здесь строка, помеченная символом «^», означает совпадение неполным набором портов.

# Пример 2. Определение узлов заданной сети, к которым имеется доступ по определённому порту

Рассмотрим ситуацию, когда требуется выяснить, к каким серверам сети 192.168.2.0 /24 открыт доступ по SSH (TCP 22). Список доступа следующий:

```
ip access-list extended ACL
10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
20 permit tcp any 192.168.2.0 0.0.0.3 eq 22 3389
30 permit tcp host 192.168.1.3 host 192.168.2.254
40 permit tcp host 192.168.1.10 any
```

Копируем access-лист в поле 1 и нажимаем кнопку 3.

Активируем переключатель 17. Алгоритм будет учитывать строки ACL, в которых IP-адреса источника и назначения полностью или частично попадают в диапазон адресов, указанный в условии.

В поле 5 вводим следующее условие:

```
tcp any gt 1023 any eq 22
```

Нажимаем кнопку 7 или клавишу “Enter”.

В поле 9 отобразится результат:

```
Блок
```

Результаты совпадения правил можно просмотреть в поле 11. Если на этом поле вызвать контекстное меню (правой кнопкой мыши) и выбрать пункт “Показать результат”, то появится дополнительное окно с выборкой сработавших строк ACL. Символ “?” в этом окне означает частичное совпадение по адресам.

Если активировать оба переключателя группы «Частичное совпадение» 17 и 25, то строки ACL, совпадающие частично по адресам и частично по портам, будут в результатах помечены точкой (символом «.»). Условные обозначения символов отображаются в нижней части окна деталей результатов проверки.

# Пример 3. Определение доступов, открытых между определёнными узлами

Выясним, какие доступы открыты от узла 192.168.1.10 к узлу 192.168.2.254 в следующем ACL:

```
ip access-list extended ACL
10 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 range 1521 1522
20 permit tcp any 192.168.2.0 0.0.0.3 eq 22 3389
30 permit tcp host 192.168.1.3 host 192.168.2.254
40 permit tcp host 192.168.1.10 any
```

Копируем access-лист в поле 1 и нажимаем кнопку 3.

Активируем переключатель 17.

В поле 5 вводим следующее условие:

```
ip host 192.168.1.10 host 192.168.2.254
```

Метод состоит в том, что заданное условие рассматривается как access-лист, а каждая строка исходного ACL как отдельное условие. Другими словами, условие и ACL меняются ролями. Именно поэтому кнопка (8), решающая эту задачу, называется “Обратная проверка”.

Нажимаем кнопку 8 или комбинацию “Ctrl-Enter”.

В поле 9 отобразится результат:

```
Блок
```

Результаты совпадения правил можно просмотреть в поле 11. Если на этом поле вызвать контекстное меню (правой кнопкой мыши) и выбрать пункт “Показать результат”, то появится дополнительное окно с выборкой сработавших строк ACL. Символ “?” в этом окне означает частичное совпадение по адресам.

Важным требованием при такой проверке является необходимость активации переключателя 17.

Зачастую для правильной работы сети необходимо открывать протокол ICMP полностью для всех сегментов. Описанная в этом разделе проверка доступов по протоколу IP между узлами среди прочих результатов будет отображать срабатывание таких правил ICMP. Если ICMP открыт по всей сетевой инфраструктуре, то наличие доступов по ICMP можно считать по умолчанию и не отображать их в результатах анализа access-листов. Для игнорирования доступов с ICMP предусмотрен переключатель 18.

# Многострочный список условий (поле 6)

Список условий (6) предназначен для ввода нескольких условий и последовательной их проверки. Метод подходит для проверки некого шаблонного набора доступов, либо для сравнения разных доступов, имеющих небольшие различия. Для ввода каждого следующего условия (новой строки) предусмотрена комбинация “Shift+Enter”. Поддерживается вставка текста с буфера обмена. Для проверки условия из списка необходимо установить на него курсор и нажать кнопку 7 (Enter) или 8 (Ctrl+Enter). В поле 15 напротив строки условия отобразится соответствующий символ результата. Он сохранится до изменения условия в этой строке списка. Маркер 14 указывает на активное условие в списке. Нажание клавиши ‘Enter’ (‘Ctrl+Enter’) совместно с клавишей «Alt» автоматически выводит отдельное окно с подробностями результатов проверки, закрыть которое можно клавишей «Esc».

# Сортировка (кнопка 13)

Распознанный access-list, выведенный в развёрнутом виде в поле 4, можно упорядочить по различным критериям и их комбинации. При нажатии на кнопку сортировки (13) открывается дополнительное окно (см. рис. 5).

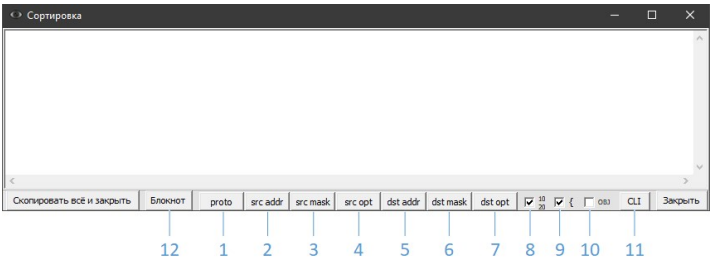



**Рис.5 Окно сортировки**

1-7 — кнопки включения элементов в цепь сортировки
8 — отображение исходных номеров строк
9 — режим группирования результатов сортировки
10 — сворачивать в object-группы
11 — вывод команд CLI, необходимых для сортировки access-листа «наживую» без его полной замены. Набор перемещаемых строк выбирается с учётом наименьшего количества перестановок.
12 — вывод содержимого в блокнот

Каждый следующий критерий в цепочке выбирается соответствующей кнопкой.

Рассмотрим следующий список доступа:

```
1 permit udp 192.168.8.0 0.0.0.255 host 192.168.38.24 eq syslog
2 permit tcp 192.168.8.0 0.0.0.255 host 192.168.38.23 eq 1514
3 permit tcp 192.168.8.0 0.0.0.255 host 192.168.38.24 eq 1514
4 permit tcp 192.168.8.0 0.0.0.255 host 192.168.38.23 eq 4041
5 permit tcp 192.168.8.0 0.0.0.255 host 192.168.12.26
6 permit ip 192.168.8.0 0.0.0.255 192.168.41.0 0.0.0.255
7 permit ip 192.168.8.0 0.0.0.255 host 192.168.41.31
```

Чтобы упорядочить эти строки сначала по IP адресу назначения, а затем по протоколу, необходимо нажать последовательно кнопки 5 и 1. Полученный результат представлен на рисунке 6.

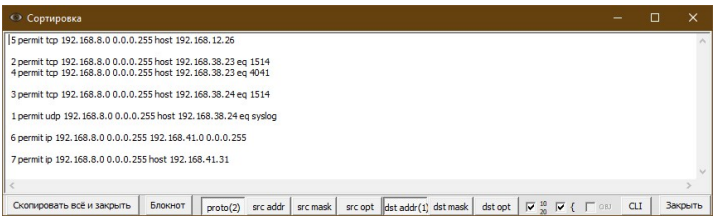



**Рис.6 Результат сортировки**

Цифры в круглых скобках на соответствующих кнопках указывают позицию элемента в цепочке сортировки. При отключении элемента из цепочки также исключаются все элементы с номерами выше отключенного.

Опция 10 предназначена для сортировки ACL с сохранением object-групп. Сначала строки с object-группами трансформируются в элементарные правила. После этого производится упорядочивание строк. На этом этапе строки, полученные из одной исходной строки с object-группой, оказываются в разных местах ACL. В такой ситуации расположение строки с object-группой в отсортированном ACL определяется по максимальной концентрации правил, полученных из исходной строки.

Рассмотрим следующий ACL:

```
ip access-list extended Test_ACL
10 permit tcp 192.168.8.0 0.0.0.255 eq 1521 addrgroup Empl_Center
20 permit udp 192.168.5.0 0.0.0.255 host 10.232.202.18
30 permit udp 192.168.8.0 0.0.0.255 host 192.168.7.34 eq ntp domain
40 permit tcp host 192.168.8.21 addrgroup HB_WEB_DMZ eq 12040 12060
50 permit tcp 192.168.8.0 0.0.0.255 eq 1521 host 10.237.49.254
```

Имеющий следующие object-группы:

```
object-group ip address Empl_Center
host-info 10.237.49.100
host-info 10.237.49.6
host-info 10.237.130.15

object-group ip address HB_WEB_DMZ
host-info 10.232.202.12
host-info 10.232.202.16
host-info 10.232.202.19
```

Сортировка по адресу назначения (кнопка “dst addr”) отобразит следующий результат (рис.7):

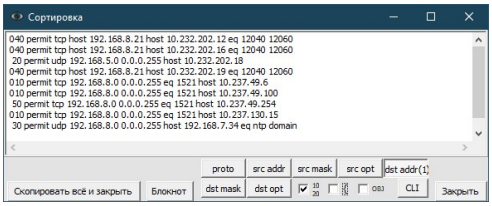



**Рис.7 Результат сортировки при наличии object-групп**

Присутствие “0” вначале строк означает, что строка получена путём извлечения содержимого object-групп. Обратите внимание, как изменилось положение исходных строк 20 и 40.

Активируем сворачивание результатов обратно в object-группы (переключатель 10). Получим следующий результат (рис.8):

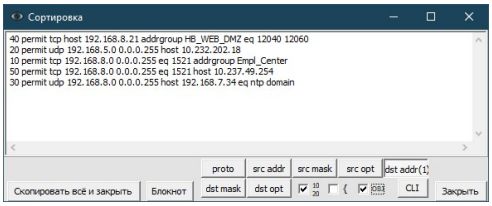



**Рис.8 Результат сортировки с поддержкой object-групп**

В реальных задачах упорядочивание ACL, содержащего object-группы, может привести к большому разбросу строк, полученных из этих object-групп. В отсортированном ACL новым местом для сворачивания таких строк обратно в object-группы алгоритм выбирает ту развёрнутую строку, к которой ближе всех находятся остальные строки её группы.

Для использования полученного упорядоченного ACL достаточно снять галочку нумерации строк и скопировать содержимое.

В промышленной среде может иметься ограничение, не допускающее временное удаление access-листа с интерфейса оборудования в целях его упорядочивания. Для этого предназначена кнопка 11 «CLI» (Рис.5), которая выводит команды CLI, необходимые для сортировки access-листа «наживую» без его полной замены. Набор перемещаемых строк выбирается алгоритмом с учётом наименьшего количества перестановок. В нашем случае вывод будет следующим:

```
ip access-list extended TestACL
no 40
no 30

ip access-list resequence TestACL 2 2

ip access-list extended TestACL
1 permit tcp host 192.168.8.21 addrgroup HB_WEB_DMZ eq 12040 12060
7 permit udp 192.168.8.0 0.0.0.255 host 192.168.7.34 eq ntp domain

ip access-list resequence TestACL 10 10
```

## Анализ на конфликты и избыточность (кнопка 12)

Кнопка “Анализ” (12) становится активной после распознания access-листа. Её нажатие запускает процесс анализа строк access-листа на конфликты и избыточность. Конфликтующей является строка access-листа, которая никогда не сработает из-за вышестоящего правила противоположного значения (“deny” после “permit” или наоборот).

К примеру, загрузим следующий ACL:

```
10 permit icmp any any
20 permit tcp host 10.15.2.11 eq 1521 host 10.15.1.10
30 deny tcp 10.15.2.0 0.0.0.255 10.15.0.0 0.0.31.255
40 permit udp 10.15.2.0 0.0.0.255 host 10.19.9.232
50 permit udp 10.15.2.0 0.0.0.255 host 10.19.9.120 eq syslog
60 permit tcp host 10.15.2.11 eq 1521 host 10.15.7.11
```

Распознаем его (кнопка 3) и нажмём кнопку “Анализ” (12). Программа предупредит нас о имеющихся конфликтах (рис. 9):

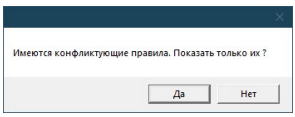



**Рис.9 Сообщение о наличии конфликтов**

Кнопка “Да” откроет окно с результатами анализа, включающими только конфликты (рис. 10):

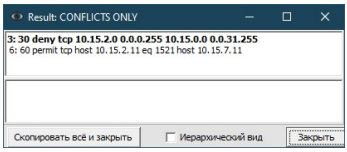



**Рис.10 Окно результатов анализа конфликтов**

Если нажать кнопку ‘Нет’ (рис.9), то откроется окно, включающее как конфликтующие, так и избыточные правила.

Рассмотрим следующий access-list:

```
10 permit icmp any any
20 permit tcp host 192.168.1.10 host 192.168.2.20 eq 22
30 permit tcp host 192.168.1.10 host 192.168.2.20
40 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

Анализ такого ACL отобразит следующие результаты:

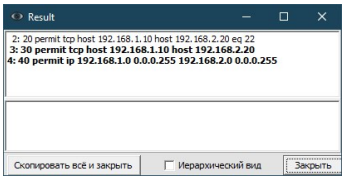



Рис.11 Окно результатов анализа

Жирным шрифтом выделены строки, описывающие более общие правила для других более детальных правил, имеющихся в ACL. Далее такие избыточные более детальные правила будем называть производными. Остальные строки (обычный шрифт) являются производными правилами.

Установив курсор на определённой строке с нажатой клавишей “Ctrl”, получим детальную информацию в нижней части окна (рис.12):

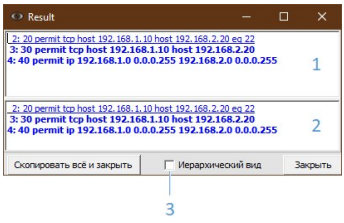



Рис.12 Детализация анализа строки

1 — поле результатов
2 — поле детализации выбранного правила
3 — иерархический вид детализации

В данном случае правило 2 является производным от правила 3. В свою очередь, правило 3 входит в правило 4. Визуально уровень такой вложенности можно определить по отступам строки вправо или выбрать иерархический вид (3). При иерархическом виде производные правила будут выведены ниже строк, в которые они входят. В поле 1 можно выделить диапазон интересуемых строк и вызвать контекстное меню (правой кнопкой мыши) со следующими вариантами действий над результатами анализа:

— “Скопировать всё”. Данный выбор копирует все правила поля 1 с исходными номерами строк вне зависимости от выделенного диапазона;

— “Скопировать с префиксом ‘no’”. Данный пункт скопирует правила выделенного диапазона с префиксом ‘no’ вначале каждой строки. Номера строк не копируются. Используется для безусловного удаления строк из ACL;

— “Скопировать с префиксом ‘no’ избыточные правила”. Данный пункт скопирует избыточные правила выделенного диапазона с префиксом ‘no’ вначале каждой строки. Номера строк не копируются;

— “Скопировать с префиксом ‘no’ избыточные правила, не имеющие производных”. Данный пункт работает как и предыдущий, но для правил, которые сами по себе не имеют производных от них избыточных правил. Рекомендован для поэтапного удаления избыточных правил от более детальных к более общим. После каждого прохода необходимо загружать обновлённый ACL и анализировать его заново. Такой способ позволяет принять решение по оптимизации правил на определённом уровне детализации;

— “Скрыть производные этого вхождения”. Пункт исключает из результатов текущее указанное правило и производные от него.

Загрузим и проанализируем следующий access-лист:

```
10 permit tcp any 192.168.2.20/32 eq 22
20 permit tcp 192.168.1.10/32 192.168.2.20/32 eq 22
30 permit tcp 192.168.1.10/32 any eq 22
40 permit ip 192.168.1.0/24 192.168.2.0/24
```

В окне анализа (рис.13) подробности по строке 2 можно получить двумя способами:

— установить на неё курсор с нажатой клавишей “Ctrl”;
— установить на неё курсор, а затем нажать “Enter”.

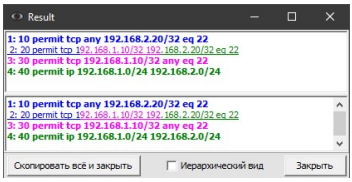



Рис.13 Множественная избыточность

По цветовому выделению можно определить для каких более общих правил эта строка является производной (избыточной).

Следует учитывать, что избыточные строки access-листа, полученные из object-групп, нельзя удалить, т.к. в исходном ACL они не присутствуют отдельными правилами. По этой причине при попадании таких строк в диапазон к удалению они не будут скопированы, а на экране появится предупреждение (рис.14):

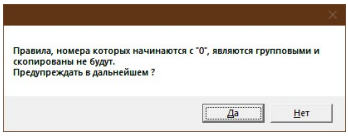



Рис.14 Предупреждение об исключении групповых правил

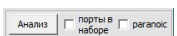
В некоторых случаях операторы “permit” и “deny” намеренно присутствуют в определённых местах одного ACL для его упрощения. Следует учитывать этот факт и анализировать такие ACL частями: отдельно до операторов “deny”, отдельно после. В альтернативе можно анализировать ACL полностью, но дополнительно обращать внимание на порядок следования конфликтующих строк и не удалять такие производные строки из исходного ACL.

По-умолчанию признаком избыточного правила является наличие более общего правила, набор портов TCP/UDP которого содержит все порты избыточного правила. Такое поведение программы можно изменить, если снять переключатель анализа отдельных портов из набора (переключатель 23 рис.1). При анализе будет учитываться избыточность отдельных портов из всего набора, содержащегося в отдельно взятой строке ACL.

Для примера рассмотрим следующий ACL:

```
10 permit udp host 192.168.8.10 host 192.168.7.34 eq ntp domain
20 permit udp 192.168.8.0 0.0.0.255 192.168.7.0 0.0.0.255 eq ntp
30 permit udp host 192.168.8.20 host 192.168.7.34 eq ntp
```

Снимем галочку «порты в наборе» переключателя 23 (рис.1).



Нажмём кнопку «Анализ». В появившемся окне результатов, зажав клавишу «Ctrl», установим курсор в первую строку:

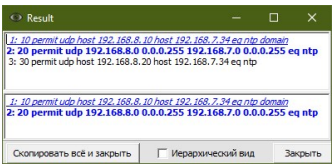



**Рис. 15 Результат анализа отдельных портов из набора**

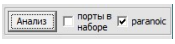
На рисунке 15 первая строка выделена курсивом. Это означает, что избыточным является не всё правило, а только часть портов из набора. В данном случае набор первого правила состоит из 2 портов: ntp (123) и domain (53). Однако, порт ntp здесь лишний, т.к. он поглощается правилом во втором условии, выделенном жирным шрифтом.

В режиме анализа избыточности отдельных портов из набора (снят флажок 23) становится доступной опция 24 («paranoic»).

Загрузим следующий ACL:

```
10 permit udp host 192.168.8.10 host 192.168.7.34 range 9000 9003
20 permit udp 192.168.8.0 0.0.0.255 192.168.7.0 0.0.0.255 eq 9002
```

Активируем переключатель 24 (paranoic):




Нажмём кнопку «Анализ». Результат отобразит частичную избыточность строки 10. Её следует разбить на две строки: первая должна содержать диапазон портов 9000-9001, вторая — только порт 9003. Порт 9002 является избыточным, т.к. попадает под действие более общего правила 20. Т.к. устранение данной избыточности ведёт к появлению двух правил вместо одного, то данный режим следует использовать в особых случаях. Режим «paranoic» выключен по-умолчанию.

# Вывод зеркального ACL (кнопка 26)

Кнопка “Отражение” (26) выводит ACL, в котором источник и назначение меняются местами. Такой функционал может быть востребован при изменении направления действия ACL.

Загрузим следующий ACL:

```
permit udp 192.168.10.0 0.0.0.255 192.168.50.0 0.0.0.255 eq syslog
permit udp 192.168.10.0 0.0.0.255 host 192.168.50.5 range snmp snmptrap
permit tcp 192.168.10.0 0.0.0.255 eq 5985 192.168.50.0 0.0.0.255
permit tcp host 192.168.10.10 eq www cmd talk 5000 5986 10.0.0.0 0.255.255.255 gt 1023
permit udp host 192.168.10.10 192.168.36.32 0.0.0.3 eq 623
permit tcp host 192.168.10.10 gt 1023 192.168.50.20 0.0.0.2 eq cmd
permit tcp 10.55.55.10 0.0.0.1 eq smtp 10.0.0.0 0.255.255.255 gt 1023
```

Кнопка «Отражение» (26) выведет следующий результат:

```
permit udp 192.168.50.0 0.0.0.255 eq 514 192.168.10.0 0.0.0.255
permit udp host 192.168.50.5 range 161 162 192.168.10.0 0.0.0.255
permit tcp 192.168.50.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 5985
permit tcp 10.0.0.0 0.255.255.255 gt 1023 host 192.168.10.10 eq 80 514 517 5000 5986
permit udp 192.168.36.32 0.0.0.3 eq 623 host 192.168.10.10
permit tcp 192.168.50.20 0.0.0.2 eq 514 host 192.168.10.10 gt 1023
permit tcp 10.0.0.0 0.255.255.255 gt 1023 10.55.55.10 0.0.0.1 eq 25
```

При этом имена портов будут заменены их числовыми значениями. Это позволяет применять новый ACL на устройствах, отличных от того, с которого был взят исходный ACL.

# Поддерживаемые протоколы (кнопка 21)

Кнопка 21 отображает список поддерживаемых программой имён портов TCP и UDP, а также кодов ICMP-сообщений (рис.16).


**Рис.16 Фрагмент списка портов и кодов ICMP**

# Опции запуска программы

Предусмотрены следующие опции запуска exe-файла:

/h, /?, /help — вызов справки параметров запуска
/l rus — выбор русского языка
/nm — включение режима “netmask”
/pa — включение режима частичного совпадения адресов
/pp — включение режима частичного совпадения портов
/skipicmp — включение режима “игнорировать ICMP при частичном совпадении”.