

ACLcheck

Утилита для анализа списков доступа сетевого оборудования Cisco

Руководство пользователя

Оглавление

Описание функционала	1
Интерфейс программы.	2
Основные шаги.	3
Пример 1. Проверка существования доступа между заданными узлами по определённому порту.	5
Пример 2. Определение узлов заданной сети, к которым имеется доступ по определённому порту.	5
Пример 3. Определение доступов, открытых между определёнными узлами.	6
Многострочный список условий (поле 6).	6
Сортировка (кнопка 13).	6
Анализ на конфликты и избыточность (кнопка 12).	8
Опции запуска программы.	10

Описание функционала.

Если Вы не раз сталкивались с большими списками доступа и/или входящими в них object-группами, то наверняка уже задавались вопросом, существует ли инструмент, позволяющий определить, пропустит ли access-лист некий заданный трафик и вообще, какие строки имеют к этому отношение.

Конечно, такие инструменты существуют и полностью или частично решают перечисленные задачи. Однако, эти инструменты как правило являются частью функционала больших "комбайнов" управления сетью, 90% функционала которых Вас не интересует.

Безусловно, никто не запрещает использовать регулярные выражения для поиска определённых строк списка доступа прямо в 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/сетями. Знающий сетевой администратор осведомлён о безрезультативности метода простого парсинга access-листа в таких ситуациях.

Данная 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 интересующими узлами или сетями.

Идея использования проста: Вы задаёте критерий, а программа находит строки access-листа, которые ему удовлетворяют. При этом, сам критерий выглядит как строка access-листа, но без использования оператора "permit" или "deny".

Если регулярно добавлять сетевые правила в access-лист без должной проверки их существования, то списки доступа могут содержать большое количество избыточных правил. Чтобы навести порядок в таких access-листах, в данной программе реализован функционал анализа списков доступа на избыточность. Вы можете выявить ненужные строки и освободить ресурсы оборудования.

Для анализа ACL с object-группами программе необходимо указать состав object-групп. Вывод ACL будет предоставлен в развёрнутом виде.

Интерфейс программы.

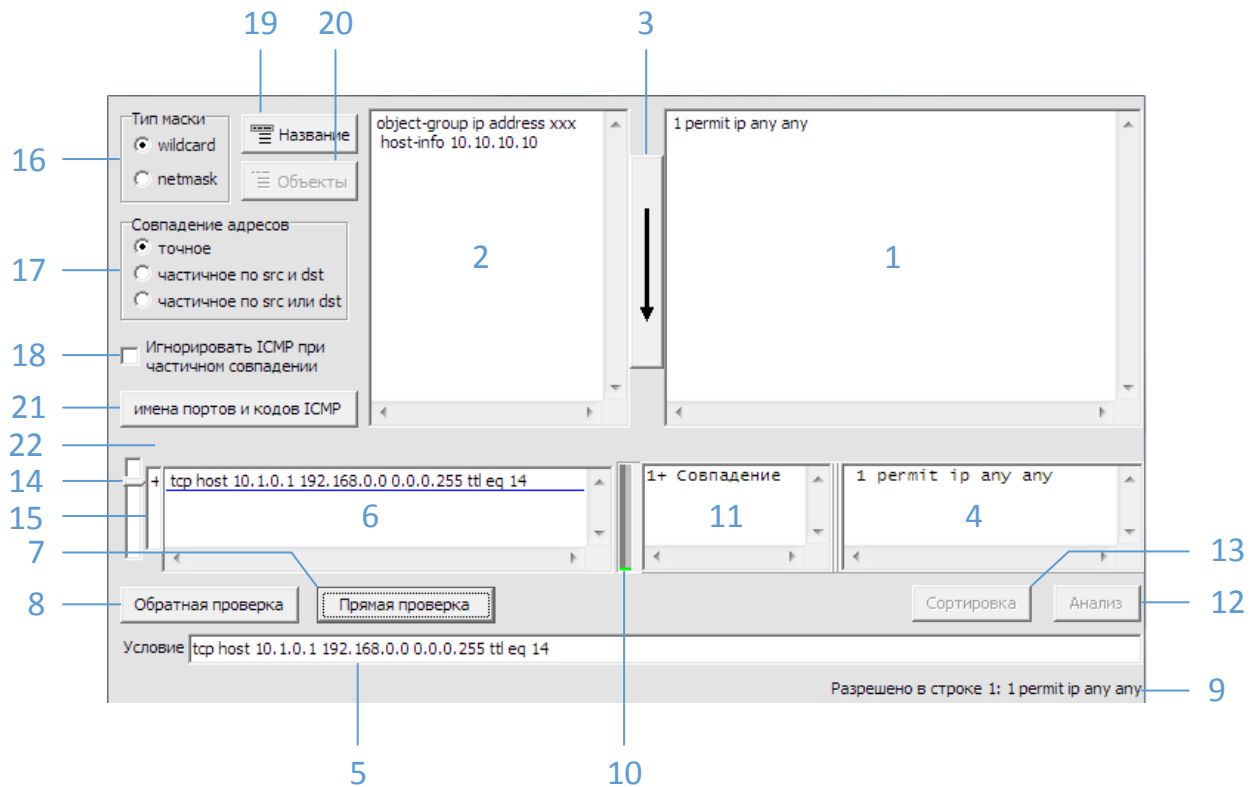


Рис.1 Главное окно

На рисунке 1 представлено главное окно программы со следующими элементами:

- 1 – Поле ввода access-листа
- 2 – Поле ввода object-групп
- 3 – Кнопка распознавания access-листа
- 4 – Поле вывода access-листа в детальном виде
- 5 – Однострочное поле ввода условия
- 6 – Многострочный список ввода условий
- 7 – Кнопка прямой проверки
- 8 – Кнопка обратной проверки
- 9 – Поле результата проверки
- 10 – Шкала позиционирования поля детальных результатов (11) по отношению ко всему ACL
- 11 – Поле просмотра детальных результатов проверки
- 12 – Кнопка анализа ACL на конфликты и избыточность
- 13 – Кнопка сортировки строк ACL по различным критериям
- 14 – Маркер текущего активного условия в многострочном списке (6)
- 15 – Шкала сокращённого обозначения результатов проверки условий многострочного списка
- 16 – Переключатель типа маски для различного типа сетевого устройства
- 17 – Переключатель алгоритма проверки адресов источника и назначения
- 18 – Активация режима игнорирования строк ACL с ICMP протоколом в режиме частичного совпадения адресов
- 19 – Меню выбора вариантов CLI команд в составе с именем ACL
- 20 – Вывод object-групп, используемых в распознанном ACL
- 21 – Вывод списка известных программе именованных протоколов, а также типов и кодов ICMP
- 22 – Поле вывода ошибок, возникающих в процессе распознавания ACL

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 по прямым командам "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
```

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

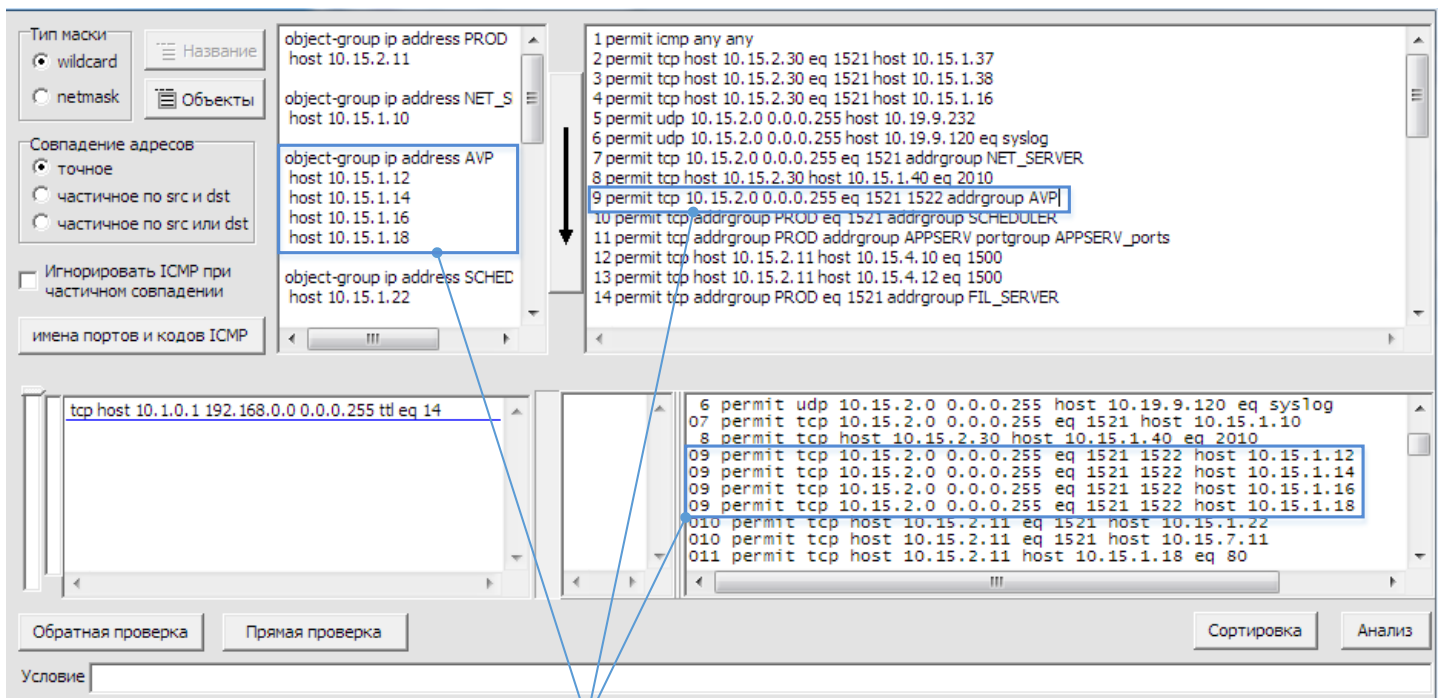
```
host 10.15.12.5
host 10.15.5.11
host 10.15.4.2
host 10.15.7.34
```

Service object group Ports1

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

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.

Если номер строки конечного access-листа дополнен '0', это означает, что данная строка получена из object-группы (рис.2).



'09' строка получена из object-группы 'AVP'

Рис.2 Стока получена из object-группы

Если access-лист скопирован вместе с его заголовком, то активируется кнопка 19, позволяющая использовать команды конфигурирования, содержащие имя access-листа.

После распознавания ACL необходимо в поле 5 ввести условие для поиска интересующего нас доступа и нажать кнопку 7. Результат поиска доступа отобразится в поле 9. В случае наличия доступа более детальная информация появится в поле 11. Вызов контекстного меню "Показать результат" по правой кнопке мыши на поле 11 позволит отобразить строки ACL, удовлетворяющие условию поиска.

Пример 1. Проверка существования доступа между заданными узлами по определённому порту.

Предположим, нас интересует существование 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 всех строк конечного (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.

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.

Ставим переключатель 17 в положение "частичное по src и dst". 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.

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.

Ставим переключатель 17 в положение "частичное по src и dst".

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. 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 установки переключателя 17 в среднее положение.

Многострочный список условий (поле 6).

Список условий (6) предназначен для ввода нескольких условий и последовательной их проверки. Для ввода каждого следующего условия (новой строки) предусмотрена комбинация "Shift-Enter". Для проверки условия из списка необходимо установить на него курсор и нажать кнопку 7 (Enter) или 8 (Ctrl-Enter). На шкале 15 напротив строки запрошенного условия отобразится соответствующий символ результата. Он сохранится до изменения условия в этой строке списка.

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. 3).

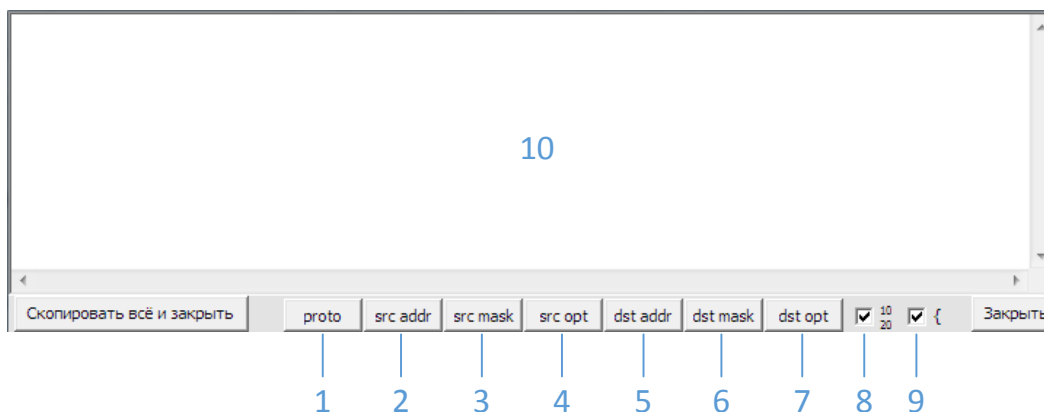


Рис.3 Окно сортировки

- 1-7 – Кнопки включения элементов в цепь сортировки
- 8 – Отображение исходных номеров строк
- 9 – Режим группирования результатов сортировки

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 рис. 4

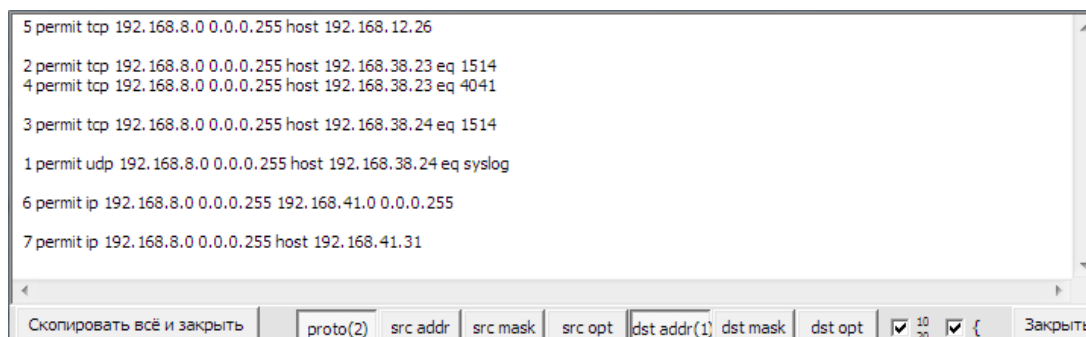


Рис.4 Результат сортировки

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.

Анализ на конфликты и избыточность (кнопка 12).

Кнопка "Анализ" (12) становится активной после распознавания 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. 5):

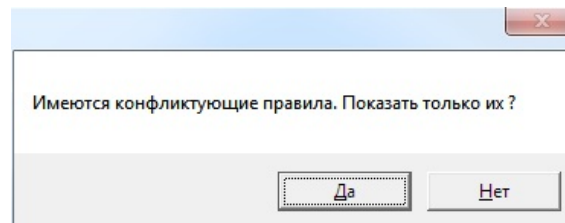


Рис.5 Сообщение о наличии конфликтов

Кнопка "Да" откроет окно с результатами анализа, включающими только конфликты (рис. 6):

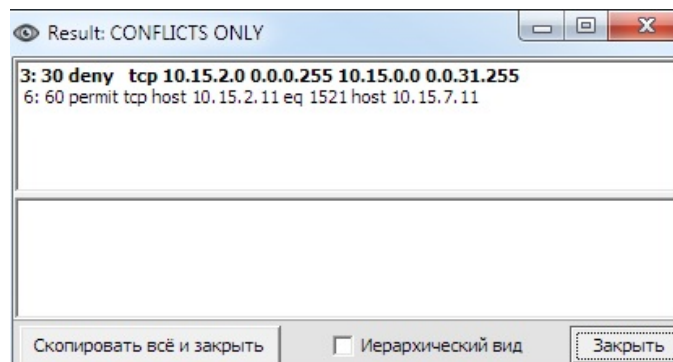


Рис.6 Окно результатов анализа конфликтов

Если нажать кнопку 'Нет' (рис.5), 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:

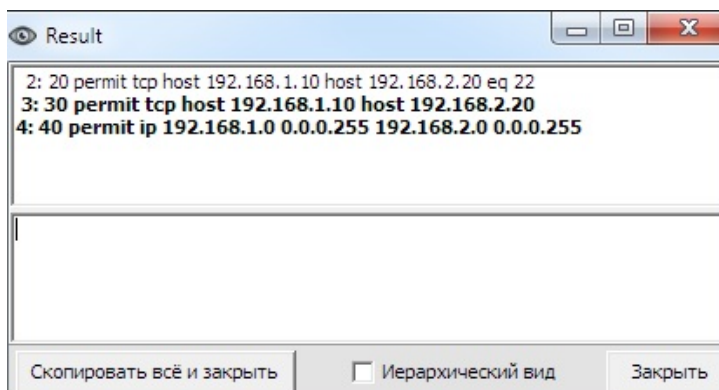


Рис.7 Окно результатов анализа

Здесь жирным шрифтом выделены строки, если имеются другие правила, попадающие под их действие. Остальные строки (обычный шрифт) являются производными правилами. Установив курсор на определённой строке, удерживая нажатой клавишу "Ctrl", получим детальную информацию в нижней части окна (рис.8):

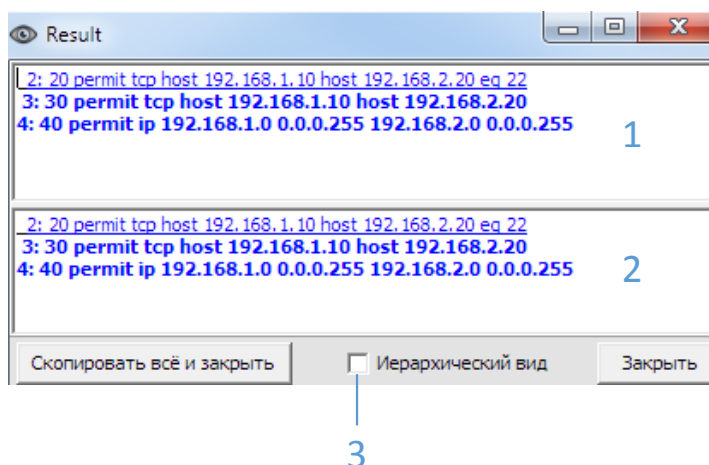


Рис.8 Детализация анализа строки

- 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. Можно выделить диапазон интересующих строк в поле 1 и вызвать контекстное меню правой кнопкой мыши, в котором выбрать варианты удаления избыточных строк.

Следует учитывать намеренное чередование операторов "permit" и "deny" в одном ACL для его упрощения. В таких случаях следует анализировать ACL частями. Например, до и после операторов "deny". Либо анализировать полностью, но дополнительно обращать

внимание на порядок следования конфликтующих строк в ACL и не удалять такие производные строки из исходного ACL.

Опции запуска программы.

Предусмотрены следующие опции запуска exe-файла:

/h, /?, /help - вызов справки параметров запуска
/l (rus) - выбор языка
/nm - включение режима "netmask"
/pm (and,or) - выбор режима совпадения адресов
/skipicmp - включение режима "игнорировать ICMP при частичном совпадении".