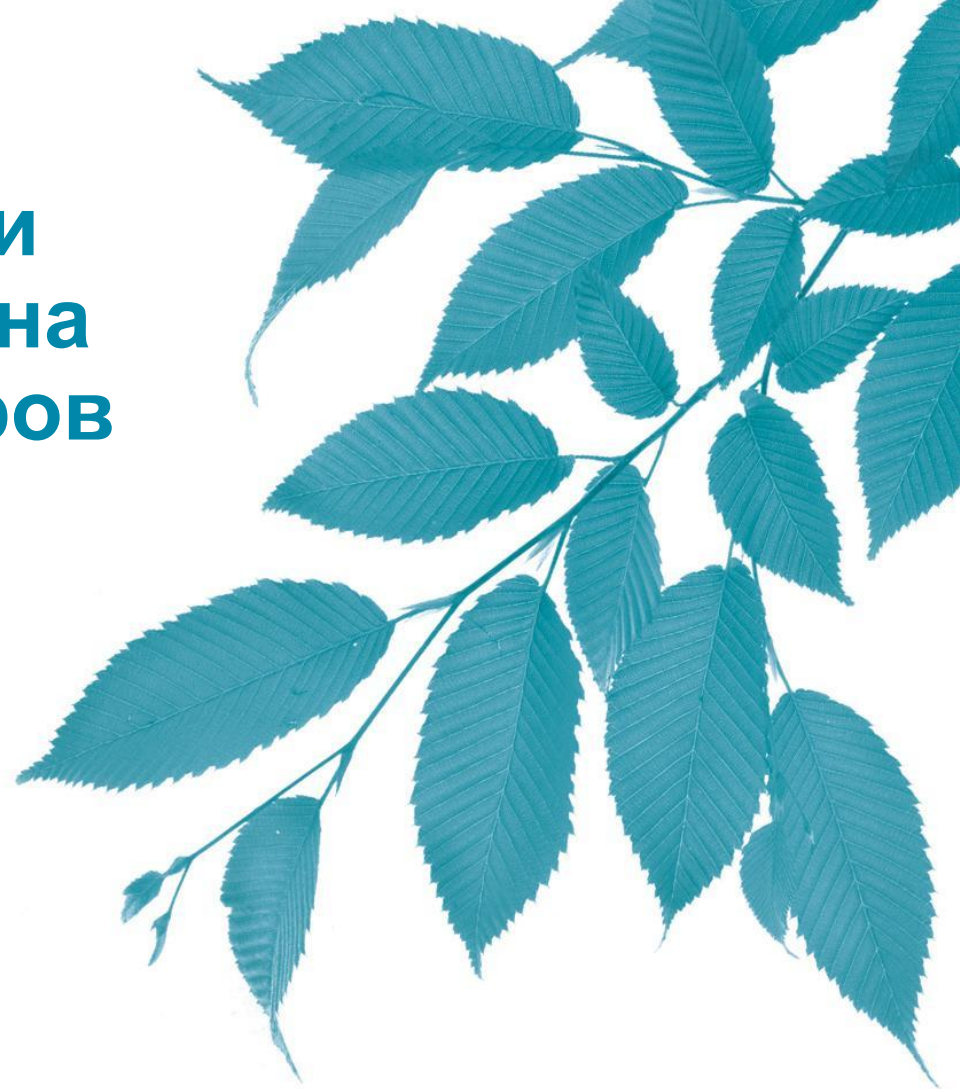


Технологии, применяемые при построении сетей на основе коммутаторов D-Link

Расширенный функционал

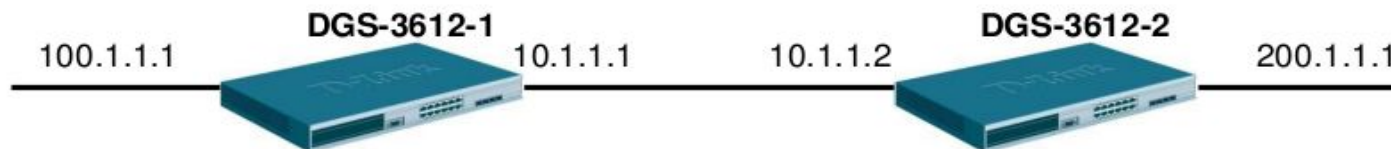
Бигаров Руслан, менеджер по проектам
e-mail: rbigarov@dlink.ru



Статическая и динамическая маршрутизации

Статическая маршрутизация

Пример использования статической маршрутизации



Настройка маршрутизации на DGS-3612-1

```
config ipif System ipaddress 10.1.1.1/24
create ipif Int1 100.1.1.1/24 Vlan1
create iproute 200.1.1.0/24 10.1.1.2
```

DES-3612:5#**sh iproute** ← Проверка состояния интерфейса
Command: show iproute
Routing Table

IP Address/Netmask	Gateway	Interface	Cost	Protocol
10.1.1.0/24	0.0.0.0	System	1	Local
100.1.1.0/24	0.0.0.0	Int1	1	Local
200.1.1.0/24	10.1.1.2	System	1	Static

Настройка маршрутизации на DGS-3612-2

```
config ipif System ipaddress 10.1.1.2/24
create ipif Int2 200.1.1.1/24 Vlan2
create iproute 100.1.1.0/24 10.1.1.1
```

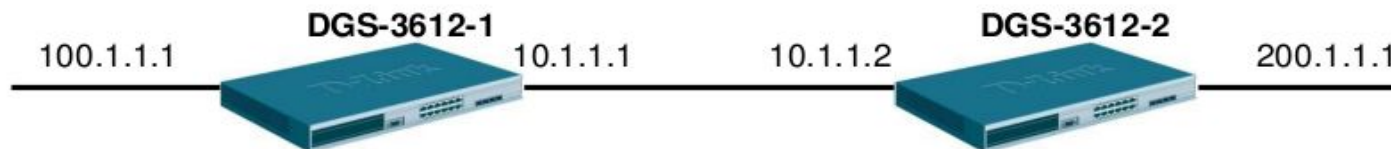
DES-3612:5#**sh iproute** ← Проверка состояния интерфейса
Command: show iproute
Routing Table

IP Address/Netmask	Gateway	Interface	Cost	Protocol
10.1.1.0/24	0.0.0.0	System	1	Local
200.1.1.0/24	0.0.0.0	Int2	1	Local
100.1.1.0/24	10.1.1.1	System	1	Static

- Inter-VLAN маршрутизация – это маршрутизация между VLAN-ами, созданными на одном коммутаторе.
- Данный тип маршрутизации работает по-умолчанию на L3 коммутаторах D-Link.
- Для работы Inter-VLAN маршрутизации достаточно: создать VLAN-ы и настроить порты для них, создать интерфейсы для данных VLAN-ов, которые будут шлюзами для своих подсетей.
- Для запрета маршрутизации между подсетями на коммутаторах используется механизм ACL.

Динамическая маршрутизация

Пример использования протокола RIPv1



Включение RIPv1 на обоих коммутаторах

```
enable rip  
config rip all tx_mode v1_only rx_mode v1_only state enable
```

DES-3612:5#**sh iproute** ← Проверка состояния интерфейса
Command: show iproute
Routing Table

IP Address/Netmask	Gateway	Interface	Cost	Protocol
10.1.1.0/24	0.0.0.0	System	1	Local
100.1.1.0/24	0.0.0.0	Int1	1	Local
200.1.1.0/24	10.1.1.2	System	2	RIP

DES-3612:5#**sh rip** ← Проверка версии и статуса RIP
Command: sh rip

RIP Global State : Enabled

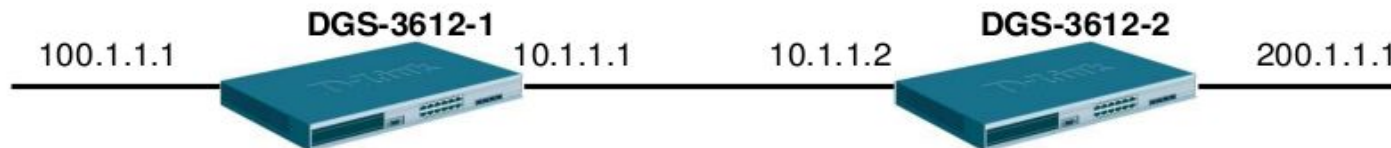
RIP Interface Settings

Interface	IP Address	TX Mode	RX Mode	Authentication	State
System	10.1.1.1/24	V1 Only	V1 Only	Disabled	Enabled
Int1	100.1.1.1/24	V1 Only	V1 Only	Disabled	Enabled

Total Entries : 2

Динамическая маршрутизация

Пример использования протокола RIPv2



Включение RIPv2 на обоих коммутаторах

```
enable rip  
config rip all tx_mode v2_only rx_mode v2_only state enable
```

DES-3612:5#**sh iproute** ← Проверка состояния интерфейса
Command: show iproute
Routing Table

IP Address/Netmask	Gateway	Interface	Cost	Protocol
10.1.1.0/24	0.0.0.0	System	1	Local
100.1.1.0/24	0.0.0.0	Int1	1	Local
200.1.1.0/24	10.1.1.2	System	2	RIP

DES-3612:5#**sh rip** ← Проверка версии и статуса RIP
Command: sh rip

RIP Global State : Enabled

RIP Interface Settings

Interface	IP Address	TX Mode	RX Mode	Authentication	State
System	10.1.1.1/24	V2 Only	V2 Only	Disabled	Enabled
Int1	100.1.1.1/24	V2 Only	V2 Only	Disabled	Enabled

Total Entries : 2

Динамическая маршрутизация

Пример использования RIP Authentication

DGS-3612:5#*config rip all authentication enable key*
Command: config rip all authentication enable key
Success.

DES-3612:5#*sh rip* ← Проверка версии и статуса RIP
Command: sh rip

RIP Global State : Enabled

RIP Interface Settings

Interface	IP Address	TX Mode	RX Mode	Authentication	State
System	10.1.1.1/24	V2 Only	V2 Only	Disabled	Enabled
Int1	100.1.1.1/24	V2 Only	V2 Only	Disabled	Enabled

Total Entries : 2



```
58 520.822746 10.1.1.2 224.0.0.9 RIPv2 Response
Frame 58 (86 bytes on wire, 86 bytes captured)
Ethernet II, Src: D-Link_5d:14:00 (00:1e:58:5d:14:00), Dst: IPv4mcast_00:00:09 (01:00:5e:00:00:09)
Internet Protocol, Src: 10.1.1.2 (10.1.1.2), Dst: 224.0.0.9 (224.0.0.9)
User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
Routing Information Protocol
  Command: response (2)
  Version: RIPv2 (2)
  Routing Domain: 0
  Authentication: Simple Password
    Authentication type: Simple Password (2)
    Password: key
  IP Address: 200.1.1.0, Metric: 1
```

Не всегда есть необходимость включать RIP на всех интерфейсах, особенно, на клиентских. В этом случае нужно либо настраивать RIP Authentication, чтобы неавторизованные маршрутизаторы не подключались к сети, либо не включать RIP на интерфейсе(ах). В этом случае для анонсирования локальных подсетей или статических маршрутов нужно использовать механизм **Route Redistribute**.

□ Анонсирование локальных подсетей:

create route redistribute dst rip src local

□ Анонсирование статических маршрутов:

create route redistribute dst rip src static

□ Просмотр созданных анонсов:

show route redistribute

□ Удаление анонсов:

delete route redistribute dst rip src local

IP-MAC-Port Binding (Привязка IP-MAC-порт)

Введение

D-Link IP-MAC-Port Binding (IMPВ) – это мощная, интегрированная функция для идентификации подключенного к сети устройства, которая гарантирует правильность связки MAC адреса, IP адреса и порта подключения. Она отслеживает информацию у ARP, DHCP, ND или IPv4/v6 пакетов, чтобы удостовериться, что все они от легальных источников, и предотвратить утечку данных к хакерам, прикидывающихся легальными сетевыми устройствами.

Где использовать IMPВ?

Имитируя шлюз или IP или MAC адрес компьютера, хакеры могут парализовать Интернет связь или тайно похищать важные данные. Сегодня очень много хакерских инструментов (программное обеспечение) можно найти в Интернете, и любой конечный пользователь может их скачать и использовать в своих целях.

D-Link IMPВ функция помогает изолировать нелегальные устройства или хакеров. Она подходит для применения на коммутаторах уровня доступа в сетях различного типа.

IP-MAC-Port Binding

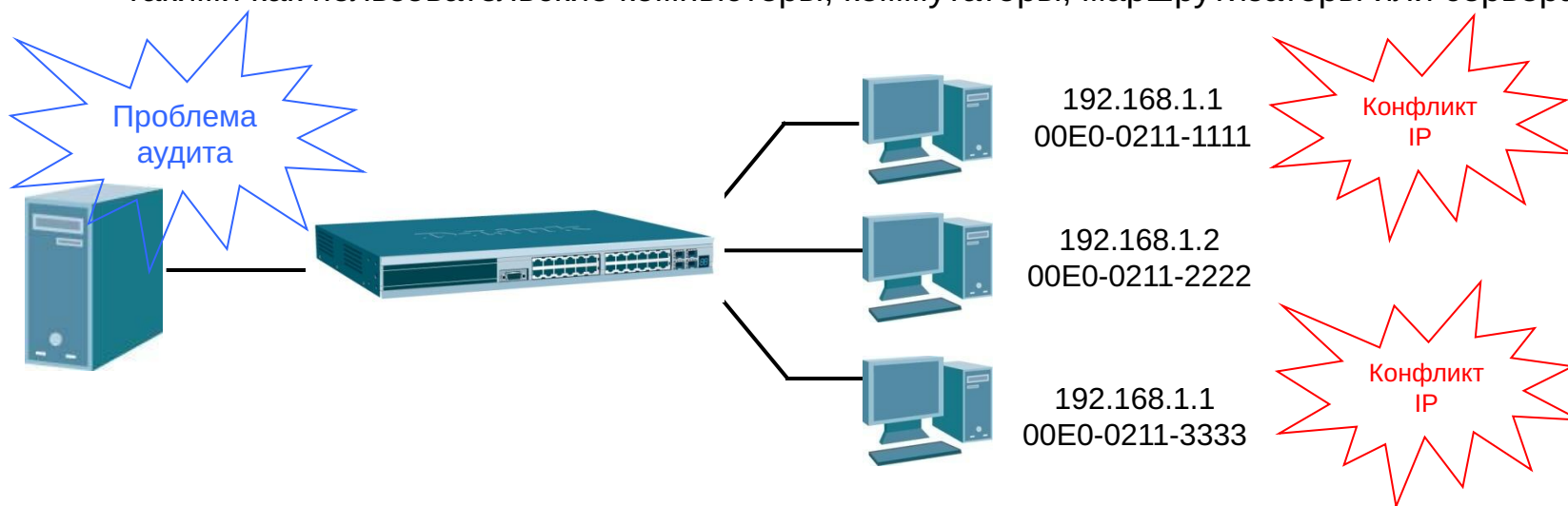
Проблема, вызванная неправильным управлением IP

Проблема аудита

Актуальные механизмы аудита, такие как syslog, application log, firewall log и т.д., базируются на IP информации. Информация лога бессмысленна, если IP может быть изменён пользователем без какого-либо контроля.

Проблема конфликта IP адресов

Конфликт IP адресов – это самая распространённая проблема в современных сетях. Пользователи меняют IP адрес вручную и происходит конфликт с другими ресурсами, такими как пользовательские компьютеры, коммутаторы, маршрутизаторы или сервера.



IP-MAC-Port Binding

Решение для улучшения управления IP адресами

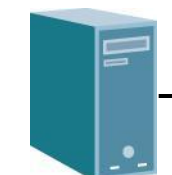
IP-MAC-Port Binding DHCP Snooping

ИМРВ изучит MAC и IP адреса клиента и автоматически и сохранит их в локальной базе.

Только клиентский трафик, у которого связка IP и MAC совпадает с сохраненной в “белом листе”, будет проброшен коммутатором.

IP-MAC записи будут созданы автоматически, когда на коммутатор придет пакет DHCP Offer с сервера

Не присутствует в списке!!!
PC-C блокируется

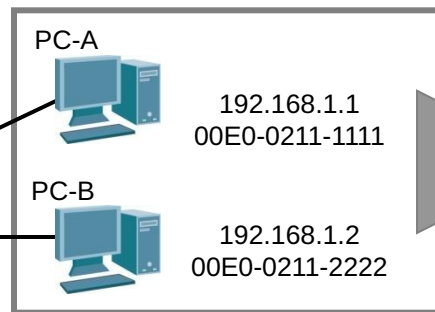


DHCP Server



Изучение адресов

Белый список		
192.168.1.1	00E0-0211-1111	Port1
192.168.1.2	00E0-0211-2222	Port2



PC-A

192.168.1.1
00E0-0211-1111

PC-B

192.168.1.2
00E0-0211-2222



PC-C

192.168.1.1
00E0-0211-3333

(IP настроен вручную пользователем)

PC-C пытается получить доступ к ресурсам сети

IP-MAC-Port Binding

ARP Inspection

ARP inspection проверяет ARP пакеты на предмет безопасности. Если ARP информация разрешённая, MAC адрес хоста будет добавлен в L2 Forwarding Database(FDB) и трафик его будет коммутироваться, в противном случае MAC адрес хоста будет добавлен в L2 Forwarding Database(FDB) как заблокированный и трафик будет отбрасываться.

ARP inspection проверяет в ARP пакетах следующую информацию:

Ethernet Header: Source Address

ARP Payload: Sender HW Address и Sender Protocol Address

Ethernet Header

ARP Payload

Destination Address FF-FF-FF-FF-FF-FF	Source Address 00-24-E8-11-22-33	HW Type	Sender HW Address 00-24-E8-11-22-33	Sender Protocol Address 10.10.10.1	Target HW Address 00-00-00-00-00-00	Target Protocol Address 10.10.10.2
--	-------------------------------------	-------	---------	-------	--	---------------------------------------	--	---------------------------------------

IP-MAC-Port Binding

IP Inspection

ARP inspection проверяет ARP пакеты на предмет безопасности. Если ARP информация разрешённая, MAC адрес хоста будет добавлен в L2 Forwarding Database(FDB) и трафик его будет коммутироваться, в противном случае MAC адрес хоста будет добавлен в L2 Forwarding Database(FDB) как заблокированный и трафик будет отбрасываться.

IP inspection проверяет в IP пакетах следующую информацию:

Ethernet Header: Source Address

IP Header: Source Address



IP-MAC-Port Binding

Режимы работы портов ARP Inspection

При активизации функции ARP Inspection на порте администратор должен указать режим его работы:

Strict Mode – в этом режиме порт по умолчанию заблокирован. Коммутатор не будет передавать пакеты до тех пор, пока не убедится в их соответствии с записями в «белом листе». Записи создаются статически или динамически в режиме DHCP Snooping.

Loose Mode – в этом режиме порт по умолчанию открыт. Порт будет заблокирован, как только через него пройдет первый недостоверный пакет. Порт проверяет только пакеты ARP и IP Broadcast.

IP-MAC-Port Binding

Пример настройки IMPV и статической записи

create address_binding ip_mac ipaddress 192.168.1.15 mac_address 00-00-5A-9E-B2-B2 ports 2
(Создаем запись IP-MAC-Port Binding, связывающую IP-MAC-адрес узла с портами подключения)

config address_binding ip_mac ports 2 arp_inspection strict ip_inspection enable protocol ipv4 allow_zeroip enable forward_dhcppkt enable

(Активизируем функцию на требуемых портах и указываем режимы работы портов)

```
DES-3200-52:admin#show address_binding ports 2
Command: show address_binding ports 2

ARP: ARP Inspection   IP: IP Inspection   ND: ND Inspection   Prot: Protocol
Port  ARP      IP      ND      Prot Zero IP      DHCP Packet  Stop Learning
-----|-----|-----|-----|-----|-----|-----|-----|
2     strict  Enabled Disabled IPv4 Allow      Forward      500/Normal

DES-3200-52:admin#show address_binding ip_mac all
Command: show address_binding ip_mac all

M(Mode) - D:DHCP, N:ND S:Static ACL - A:Active I:Inactive

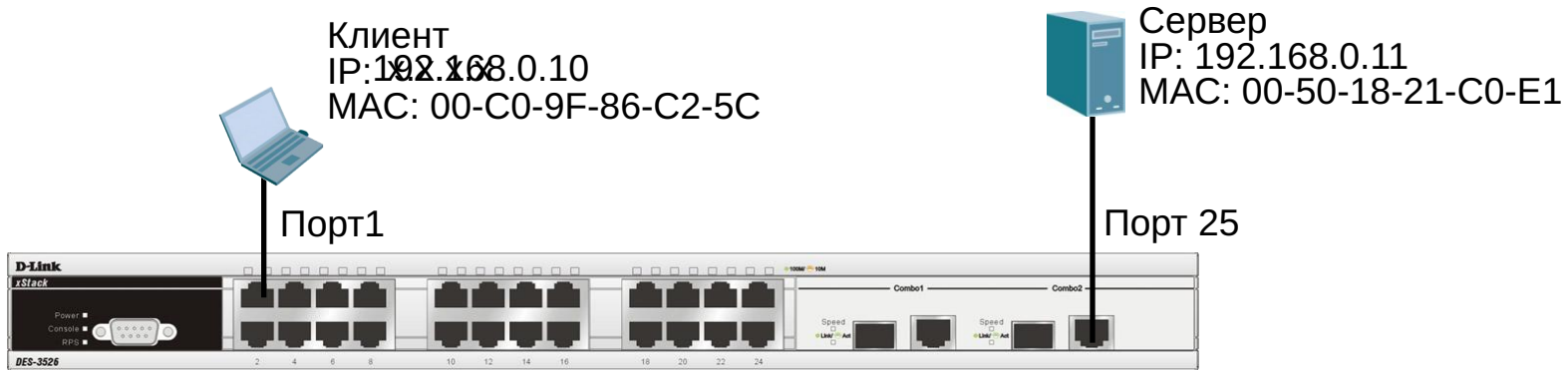
IP Address                MAC Address          M  ACL  Ports
-----|-----|-----|-----|
192.168.1.15              00-00-5A-9E-B2-B2  S  A   2

Total Entries : 1
DES-3200-52:admin#
```


IP-MAC-Port Binding

Пример работы IMPB в DHCP Snooping режиме

Коммутатор динамически создает запись IMPB после того, как клиент получит IP-адрес от DHCP-сервера.



DHCP Discovery Broadcast



DHCP Offer unicast



- 1 Компьютер в сети (DHCP - клиент) генерирует DHCP - запросы и широковещательно рассылает их в сеть.
- 2 DHCP сервер отвечает пакетом DHCP Offer unicast. Коммутатор берет необходимую информацию: MAC адрес клиента, выданный IP адрес и Lease time, и на базе этой информации создаёт запись в IMPB таблице.

IP-MAC-Port Binding

Пример настройки IMPB в DHCP Snooping режиме

```
enable address_binding dhcp_snoop
```

(Активизируем функцию IP-MAC-Port Binding в режиме DHCP Snooping глобально на коммутаторе.)

```
config address_binding dhcp_snoop max_entry ports 1-10 limit 10
```

(Указываем максимальное количество создаваемых в процессе автоизучения записей IP-MAC на порт.)

```
config address_binding ip_mac ports 2 arp_inspection strict ip_inspection enable protocol ipv4  
allow_zeroip enable forward_dhcppkt enable
```

(Активизируем функцию на требуемых портах и указываем режимы работы портов.)

```
DES-3200-52:admin#show address_binding dhcp_snoop binding_entry  
Command: show address_binding dhcp_snoop binding_entry
```

```
S (Status) - A: Active, I: Inactive  
Time - Left Time (sec)
```

IP Address	MAC Address	S	LT(sec)	Port
192.168.1.15	00-00-5A-9E-B2-B2	A	86373	2

```
Total Entries : 1
```

```
DES-3200-52:admin#
```

IP-MAC-Port Binding

Дополнительные параметры IMPB

- ***max_entry ... limit 1*** – максимальное кол-во записей IMP, которые может изучить коммутатор на порту в режиме DHCP Snooping. Возможные значения зависят от модели коммутатора и версии прошивки.
- ***allow_zeroip*** – возможность пропуска DHCP Discovery пакетов при включённой функции IMP с source_IP = 0.0.0.0. Необходима для полноценной работы OS: Vista, Win 7, MAC OS 10 и старше и т.д., по протоколу DHCP при использовании функции IMPB.
- ***forward_dhcppkt*** – коммутатор пробрасывает все DHCP пакеты по умолчанию. Если на порту задан Strict режим, все DHCP пакеты будут отброшены. В этом случае, включаем опцию forward_dhcppkt, чтобы коммутатор пробрасывал клиентские DHCP пакеты. Включение этой функции также гарантирует, что DHCP snooping работает правильно.
- ***stop_learning_threshold <value 0-500>*** – это ограничение кол-ва заносимых записей в FDB, при использовании IMPB. При использовании IMPB коммутатор блокирует записи не подходящие под сконфигурированную связку, но побочным эффектом становится занесение в FDB таблицу всех обработанных пакетов. Для предотвращения переполнения FDB при использовании IMPB, был введён новый параметр threshold, который наблюдает за кол-ом заблокированных записей и при превышении этого параметра, вводит порт в disable learning с соответствующим оповещением и созданием записи в лог. Использование функции позволит избежать атак со стороны пользователей и подвергать нарушителя административному взысканию, потому как порт придётся поднимать вручную. Значение 0 означает, что изучение MAC адресов не имеет ограничения.

ACL

Списки управления доступом,
Классификация трафика,
маркировка и отбрасывание

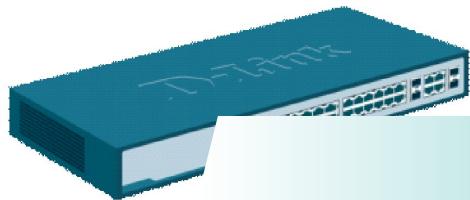
Контроль сетевых приложений

L2/3/4 ACL (Access Control List)

Коммутаторы D-Link предоставляют наиболее полный набор ACL, помогающих сетевому администратору осуществлять контроль над приложениями. При этом не будет потерь производительности, поскольку проверка осуществляется на аппаратном уровне.

ACL в коммутаторах D-Link могут фильтровать пакеты, основываясь на информации разных уровней:

- ✓ Порт коммутатора
- ✓ MAC/ IPv4/ IPv6-адрес
- ✓ Тип Ethernet/ Тип протокола
- ✓ VLAN
- ✓ 802.1p/ DSCP
- ✓ TCP/ UDP-порт [тип приложения]
- ✓ Содержание пакета [поле данных приложения]



- ACL могут проверять содержимое пакетов на предмет наличия новых изменённых потоков



- Инфицированные клиенты
- Неисправные сервера/ точки доступа
- Компьютеры злоумышленников
- Несанкционированные пользователи



Сетевой трафик

• Управляемые коммутаторы D-Link могут эффективно предотвращать проникновение вредоносного трафика в сеть

Настройка ACL профилей и правил

- Проанализируйте задачи фильтрации и определитесь с типом профиля доступа - Ethernet или IPv4 или IPv6
- Зафиксируйте стратегию фильтрации
- Основываясь на этой стратегии, определите какая необходима маска профиля доступа (access profile mask) и создайте её. (команда **create access_profile**)
- Добавьте правило профиля доступа (access profile rule), связанное с этой маской (команда **config access_profile**)
- Правила профиля доступа проверяются в соответствии с номером access_id. Чем меньше ID, тем раньше проверяется правило. Если не одно правило не сработало, пакет пропускается.
- При необходимости, когда срабатывает правило, биты 802.1p/DSCP могут быть заменены на новые значения перед отправкой пакета, выступая в качестве “**Маркера**” в модели DSCP PHB (Per-Hop Behavior – пошаговое поведение).

Типы профилей

1. Ethernet:

- VLAN
- MAC источника
- MAC назначения
- 802.1p
- Тип Ethernet

2. IPv4:

- VLAN
- IP источника
- IP назначения
- DSCP
- Протокол (ICMP, IGMP, TCP, UDP)
- TCP/UDP-порт

3. IPv6:

- Traffic class (приоритет)
- IP источника
- IP назначения*
- Flow label (метка потока)
- Протокол (TCP, UDP)

4. Фильтрация по содержимому пакета (первые 80 или 128* байт пакета).

* В зависимости от модели

Ethernet профиль ACL

Создание Ethernet профиля ACL

Add ACL Profile Safeguard

Select Profile ID

Select ACL Type

Ethernet ACL IPv4 ACL Packet Content ACL

IPv6 ACL

You can select the field in the packet to create filtering mask

MAC Address	VLAN	802.1p	Ethernet Type	PayLoad
-------------	------	--------	---------------	---------

MAC Address

Source MAC Mask

Destination MAC Mask

802.1Q VLAN

VLAN

VLAN Mask (0-FFF)

Ethernet профиль ACL

Настройка правил в Ethernet профиле

Add Access Rule

Profile Information

Profile ID	1	Profile Type	Ethernet
Owner Type	ACL	VLAN	0xFFFF
Source MAC	00-00-00-00-00-00	Destination MAC	00-00-00-00-00-00
802.1P	Yes	Ethernet Type	Yes

Rule Detail
(Keep an input field as blank to treat the corresponding option as do not care)

Access ID (1-65535) Auto Assign

VLAN Name

VLAN ID (1-4094) Mask (0-FFF)

Source MAC Address e.g.(00-00-00-00-FF-FF)

Source MAC Mask e.g.(00-00-00-00-FF-FF)

Destination MAC Address e.g.(00-00-00-00-FF-FF)

Destination MAC Mask e.g.(00-00-00-00-FF-FF)

802.1P (0-7)

Ethernet Type (0-FFFF)

Rule Action

Action

Priority (0-7)

Replace Priority

Replace DSCP (0-63)

Time Range Name

Counter

e.g.(1,4-6,9)

Ethernet профиль ACL

Синтаксис CLI создание и настройка Ethernet профиля

Создание Ethernet профиля доступа:

```
create access_profile [ ethernet {vlan {<hex 0x0-0x0fff>} | source_mac <macmask> | destination_mac <macmask> | 802.1p | ethernet_type} ] profile_id <value 1-512>
```

Удаление профиля доступа:

```
delete access_profile [profile_id <value 1-512> | all]
```

Создание / удаление правила Ethernet профиля доступа:

```
config access_profile [profile_id <value 1-512>] [add access_id [auto_assign | <value 1-65535>] [ethernet {[vlan <vlan_name 32> | vlan_id <vid>} {mask <hex 0x0-0x0fff>} | source_mac <macaddr> {mask <macmask>} | destination_mac <macaddr> {mask <macmask>} | 802.1p <value 0-7> | ethernet_type <hex 0x0-0xffff>}] [port [<portlist>|all]] [permit {priority<value 0-7> {replace_priority} | replace_dscp_with <value0-63>| counter [enable | disable]} | deny | mirror ] { time_range <range_name 32>} | delete access_id <value 1-65535>]
```

Просмотр имеющихся на коммутаторе профилей доступа и правил профилей доступа:

```
show access_profile {profile_id <value 1-512>}
```

Ethernet type - Ethertype

Как видно из предыдущего слайда, Ethernet профили доступа позволяют анализировать пакеты на основании поля Ethernet Type или Ethertype.

Поле Ethertype указывает на протокол, инкапсулированный в кадр Ethernet.

Ниже приведен пример ARP Request пакета, для которого значение Ethertype равно 0x0806, что и указывает на протокол ARP:

СМЕЩЕНИЕ	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F
0000	FFFF	FFFF	FFFF	0021	918C	F1D4	8100	0002
0010	0806	0001	0800	0604	0001	0021	918C	F1D4
0020	0A5A	5ADD	0000	0000	0000	0A5A	5AB3	

Таким образом, при помощи правил ACL можно ограничить трафик в сети для определенных сетевых протоколов, либо вообще запретить клиенту использование определенных протоколов.

Ниже приведен список значений Ethertype наиболее “распространенного” в сети трафика:

0x0800	Internet Protocol, Version 4 (IPv4)
0x86DD	Internet Protocol, Version 6 (IPv6)
0x0806	Address Resolution Protocol (ARP)
0x0842	Wake-on-LAN Magic Packet
0x8809	Slow Protocols (IEEE 802.3) - используется LACP
0x8863	PPPoE Discovery Stage
0x8864	PPPoE Session Stage
0x88CC	LLDP
0x9000	Configuration Test Protocol (Loop)

IP профиль ACL

Создание IP профиля в ACL

Add ACL Profile

Select Profile ID: 1

Select ACL Type:

- Ethernet ACL
- IPv4 ACL
- IPv6 ACL
- Packet Content ACL

Field: TCP

Select

You can select the field in the packet to create filtering mask

L2 Header	VLAN	IPv4 DSCP	IPv4 Address	TCP
-----------	------	-----------	--------------	-----

802.1Q VLAN

- VLAN
- VLAN Mask (0-FFF)

IPv4 DSCP

- DSCP

IPv4 Address

- Source IP Mask
- Destination IP Mask

TCP

- TCP
- Source Port Mask (0-FFFF)
- Destination Port Mask (0-FFFF)
- TCP Flag Bits
- Check All
- URG
- ACK
- PSH
- RST
- SYN
- FIN

<< Back Create

IP профиль ACL

Настройка правил в IP профиле

Add Access Rule

Safeguard

Profile Information

Profile ID	1	Profile Type	IP
Owner Type	ACL	VLAN	0xFFF
Source IP	255.255.255.0	Destination IP	255.255.255.0
DSCP	Yes	TCP	Yes
TCP Source Port	0xFFFF	TCP Destination Port	0xFFFF

Rule Detail
(Keep an input field as blank to treat the corresponding option as do not care)

Access ID (1-65535) Auto Assign

VLAN Name

VLAN ID (1-4094) Mask (0-FFF)

Source IP Address e.g.(192.168.1.10)

Source IP Mask e.g.(192.168.1.10)

Destination IP Address e.g.(192.168.1.10)

Destination IP Mask e.g.(192.168.1.10)

DSCP e.g.(0-63)

TCP

Source Port e.g.(0-65535) Mask (0-FFFF)

Destination Port e.g.(0-65535) Mask (0-FFFF)

Rule Action

Action

Priority (0-7)

Replace Priority

Replace DSCP (0-63)

Time Range Name

Counter

IP профиль ACL

Синтаксис CLI создание и настройка IP профиля

Создание IP профиля доступа:

```
create access_profile [ ip {vlan {<hex 0x0-0x0fff>} | source_ip_mask <netmask> | destination_ip_mask <netmask> | dscp | [ icmp {type | code} | igmp {type} | tcp {src_port_mask <hex 0x0-0xffff> | dst_port_mask <hex 0x0-0xffff> | flag_mask [ all | {urg | ack | psh | rst | syn | fin} (1) ] } | udp {src_port_mask <hex 0x0-0xffff> | dst_port_mask <hex 0x0-0xffff> } | protocol_id_mask<0x0-0xff> } } ] profile_id <value 1-512>
```

Удаление профиля доступа:

```
delete access_profile [profile_id <value 1-512> | all]
```

Создание / удаление правила IP профиля доступа:

```
config access_profile [profile_id <value 1-512>] [add access_id [auto_assign | <value 1-65535>] [ip {[vlan <vlan_name 32> | vlan_id <vid>] {mask <hex 0x0-0x0fff>} | source_ip <ipaddr> {mask <netmask>} | destination_ip <ipaddr> {mask <netmask>} | dscp <value 0-63> | [ icmp {type <value 0-255> code <value 0-255>} | igmp {type <value 0-255>} | tcp {src_port <value 0-65535> {mask <hex 0x0-0xffff>} | dst_port <value 0-65535> {mask <hex 0x0-0xffff>} | flag [all | {urg | ack | psh | rst | syn | fin} (1) ] } | udp {src_port <value 0-65535> | dst_port <value 0-65535> } | protocol_id <value 0-255> } (1) ]}] [port [<portlist>|all]] [permit {priority<value 0-7> {replace_priority} | replace_dscp_with <value0-63>} | counter [enable | disable] } | deny | mirror ] { time_range <range_name 32>} | delete access_id <value 1-65535>]
```

Просмотр имеющихся на коммутаторе профилей доступа и правил профилей доступа:

```
show access_profile {profile_id <value 1-512>}
```

Ethernet ACL в коммутаторах. Пример 1.

Пример: Разрешить некоторым пользователям выход в Internet по MAC- адресам



Шлюз Internet:
IP = 10.254.254.251/8
00-50-BA-99-99-99

Разрешён доступ в Internet:
PC1:10.1.1.1/8,
00-50-BA-11-11-11
PC2:10.2.2.2/8,
00-50-BA-22-22-22
Шлюз = 10.254.254.251

Другие PC (доступ в Internet запрещён):
IP: 10.x.x.x/8

Ethernet ACL в коммутаторах. Пример 1.

Правила:

Правило 1: Если MAC назначения = Шлюз, то запретить (на портах 3-24)

Правило 2: В противном случае разрешить всё остальное (по умолчанию).

Правило 1

```
create access_profile ethernet destination_mac FF-FF-FF-FF-FF-FF profile_id 10
config access_profile profile_id 10 add access_id 10 ethernet destination_mac
00-50-ba-99-99-99 port 3-24 deny
```

Правило 2: Другие пакеты разрешены по умолчанию

Проверка:

Компьютеры кроме PC1 и PC2 не могут получить доступ в Internet (в соответствии с правилом 1 обращение к MAC адресу шлюза запрещено на портах 3-24).

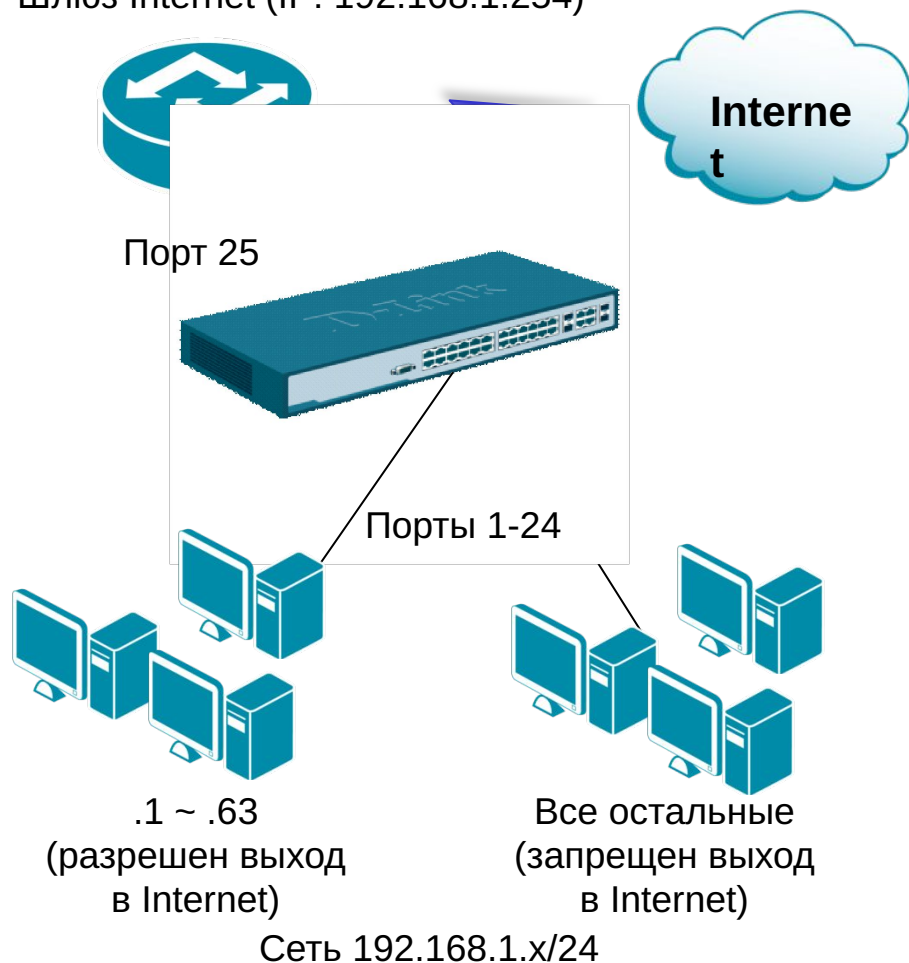
PC1, PC2 могут получить доступ в Internet (так как в правиле 1 не указаны порты 1 и 2, к которым они подключены).

PC1, PC2 и другие могут получить доступ друг к другу (Intranet ОК, в соответствии с тем, что в правиле 1 запрещено обращение только к MAC адресу шлюза).

IP ACL в коммутаторах. Пример 2.

Пример: Разрешить некоторым пользователям выход в Internet по IP

Шлюз Internet (IP: 192.168.1.254)



- Доступ в Internet разрешён:
192.168.1.1 ~ 192.168.1.63
- Остальные пользуются только
Intranet

IP ACL в коммутаторах. Пример 2.

Правила:

Правило 1: Если IP источника = 192.168.1.0/26 и IP назначения = 192.168.1.254/32, то разрешить (для .1 - .63 разрешить доступ в Internet)

Правило 2: Если IP источника = 192.168.1.0/24 и IP назначения = 192.168.1.254/32, то запретить (для .1 - .254 запретить доступ в Internet)

Правило 3: В противном случае все запретить остальное по умолчанию

Правило 1: Разрешить для .1 - .63 доступ в Internet

```
create access_profile ip source_ip_mask 255.255.255.192 destination_ip_mask 0.0.0.0 profile_id 10
config access_profile profile_id 10 add access_id auto_assign ip source_ip 192.168.1.0 destination_ip 0.0.0.0
port 1-24 permit
```

Правило 2: Разрешить для .1 - .254 доступ в Intranet(локальную сеть)

```
create access_profile ip source_ip_mask 255.255.255.0 destination_ip_mask 255.255.255.0 profile_id 20
config access_profile profile_id 20 add access_id auto_assign ip source_ip 192.168.1.0 destination_ip
192.168.1.0 port 1-24 permit
```

Правило 3: Все остальное запретить

```
create access_profile ip source_ip_mask 0.0.0.0 profile_id 30
config access_profile profile_id 30 add access_id auto_assign ip source_ip 0.0.0.0 port 1-24 deny
```

Проверка:

1. 192.168.1.1 - 192.168.1.63 могут получить доступ к Internet и ко всем остальным PC .64 - .253 (правило 1).
2. PC .64 - .253 могут иметь доступ к PC .1 - .253 (правило 2), но не могут выйти в Internet (правило 3).

Блокировка SMB трафика. Пример 3.

1. Фильтрация TCP портов 135, 137, 138, 139, 445.

Команды CLI:

```
create access_profile ip tcp dst_port_mask 0xFFFF profile_id 30
config access_profile profile_id 30 add access_id auto_assign ip tcp dst_port 135 port 1-24 deny
config access_profile profile_id 30 add access_id auto_assign ip tcp dst_port 137 port 1-24 deny
config access_profile profile_id 30 add access_id auto_assign ip tcp dst_port 138 port 1-24 deny
config access_profile profile_id 30 add access_id auto_assign ip tcp dst_port 139 port 1-24 deny
config access_profile profile_id 30 add access_id auto_assign ip tcp dst_port 445 port 1-24 deny
```

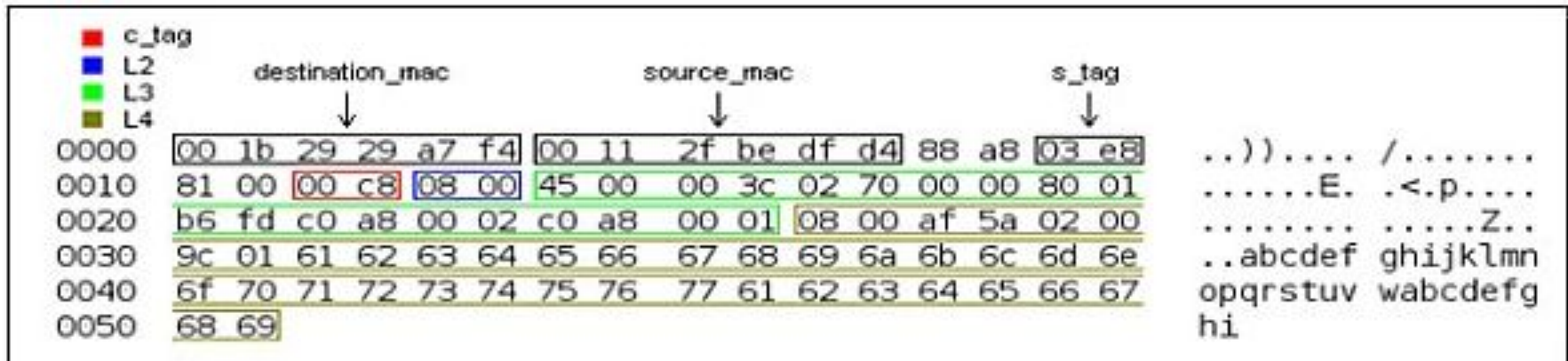
2. Фильтрация UDP портов 135, 137, 138, 139, 445

Команды CLI:

```
create access_profile ip udp dst_port_mask 0xFFFF profile_id 40
config access_profile profile_id 40 add access_id auto_assign ip udp dst_port 135 port 1-24 deny
config access_profile profile_id 40 add access_id auto_assign ip udp dst_port 137 port 1-24 deny
config access_profile profile_id 40 add access_id auto_assign ip udp dst_port 138 port 1-24 deny
config access_profile profile_id 40 add access_id auto_assign ip udp dst_port 139 port 1-24 deny
config access_profile profile_id 40 add access_id auto_assign ip udp dst_port 445 port 1-24 deny
```

Packet Content Filtering профиль ACL

Принцип работы Packet content filtering (PCF) на примере **ICMP echo request** трафика:



Основные части пакета, которыми можно оперировать при составлении правила PCF:

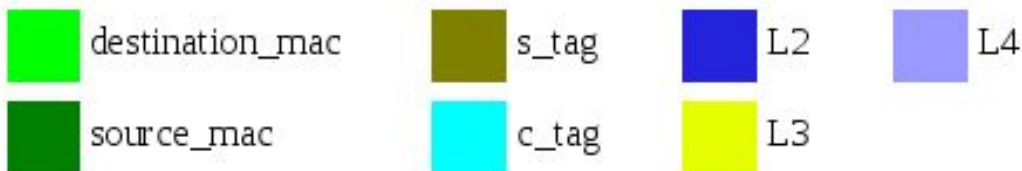
- destination_mac — MAC-адрес назначения;
- source_mac — MAC-адрес источника;
- s_tag — внешний тег (при использовании Q-in-Q);
- c_tag — тег 802.1Q;
- L2 — часть, начинающаяся сразу после тега 802.1Q (или от начала ether type);
- L3 — часть, начинающаяся по окончании ether type;
- L4 — часть, начинающаяся по окончании IP заголовка.

Запрет ICMP трафика с помощью PCF. Пример 3.

Задача : Запрет ICMP echo request трафика

Приведем соответствующий пакет:

0000	001b	2929	a7f4	0011	2fbe	dfd4	88a8	03e8
0010	8100	00c8	0800	4500	003c	0270	0000	8001
0020	b6fd	c0a8	0002	c0a8	0001	0800	af5a	0200
0030	9c01	6162	6364	6566	6768	696a	6b6c	6d6e
0040	6f70	7172	7374	7576	7761	6263	6465	6667
0050	6869							



Выборку будем осуществлять по ether type (IP), протоколу (icmp) и по типу icmp 8 (request).

В рассматриваемом нами пакете ether type находится в L2 части в 1 и 2 байте (смещение 0), информация о протоколе (icmp) находится в L3 части в 10 байте (смещение 8), тип icmp находится в первом байте L4 части (смещение 0).

Анализируемые значения выделены красным.

Запрет ICMP трафика с помощью PCF. Пример 3.

Создадим сначала профиль:

```
create access_profile packet_content_mask offset1 I2 0 0xffff offset2 I3 8 0x00ff offset3 I4 0 0xff00 profile_id 1
```

Приведенная выше команда означает, что первым обрабатываемым полем (offset1) в нашем случае будет нулевое смещение в L2 части. Маску значений задаем равной 0xffff, это означает, что в создаваемых в этом профиле правилах, манипулировать мы будем 1 и 2 байтами L2 части.

Второе обрабатываемое поле (offset2) будет находиться по восьмому смещению в L3 части. Маску значений задаем равной 0x00ff, в итоге вместе со смещением это означает, что в создаваемых в этом профиле правилах, манипулировать мы будем 10-м байтом L3 части.

И наконец, третье обрабатываемое поле (offset3) будет находиться по нулевому смещению в L4 части. Маску значений задаем равной 0xff00, это означает, что в создаваемых в этом профиле правилах, манипулировать мы будем 1-м байтом L4 части.

Создадим и само правило с учетом приведенных выше значений.

```
config access_profile profile_id 1 add access_id auto_assign packet_content offset1 0x0800 offset2 0x0001 offset3 0x0800 port 1 deny
```

В общем виде получим следующее:

```
create access_profile packet_content_mask offset1 I2 0 0xffff offset2 I3 8 0x00ff offset3 I4 0 0xff00 profile_id 1  
config access_profile profile_id 1 add access_id auto_assign packet_content offset1 0x0800 offset2 0x0001 offset3 0x0800 port 1 deny
```






Результат: указанное выше правило запретит ICMP реквесты на 1 порту коммутатора.

Запрет SMB трафика с помощью PCF. Пример 4.

Задача: Заблокировать SMB трафик (порты 135, 137, 138, 139, 445) на физических портах коммутатора 1-24.

Рассмотрим такой пакет, взятый из Ethereal:

0000	0012	3456	7890	0022	b0de	0f4e	0800	4500
0010	0034	fe74	4000	4006	b945	c0a8	00dd	c0a8
0020	00dc	8f36	008b	baee	6042	0a54	36b8	8010
0030	005c	68fd	0000	0101	080a	0028	9e33	0000
0040	0000							

	destination_mac		L2		L4
	source_mac		L3		

Выборку будем осуществлять по порту назначения.

В рассматриваемом нами пакете такая информация находится в L4 части в 3 и 4 байте (смещение 2). Анализируемые значения выделены красным. В данном случае значение — 008b (шестнадцатиричная CC), что соответствует числу 139 в десятичной системе счисления, а значит блокировать мы будем 139 порт.

Запрет SMB трафика с помощью PCF. Пример 4.

Создадим сначала профиль и правило, согласно оговоренному ранее условию:

```
create access_profile packet_content_mask offset1 I4 2 0xFFFF profile_id 1  
config access_profile profile_id 1 add access_id 1 packet_content offset1 0x008b port 1-24 deny
```

По аналогии создадим правила, блокирующие другие порты: 135 (87h), 137 (89h), 138 (8ah), 445 (1bdh)

```
config access_profile profile_id 1 add access_id 2 packet_content offset1 0x0087 port 1-24 deny  
config access_profile profile_id 1 add access_id 3 packet_content offset1 0x0089 port 1-24 deny  
config access_profile profile_id 1 add access_id 4 packet_content offset1 0x008a port 1-24 deny  
config access_profile profile_id 1 add access_id 5 packet_content offset1 0x01bd port 1-24 deny
```

Таким образом, общий вид правил будет следующим:

```
create access_profile packet_content_mask offset1 I4 2 0xFFFF profile_id 1  
config access_profile profile_id 1 add access_id 1 packet_content offset1 0x008b port 1-24 deny  
config access_profile profile_id 1 add access_id 2 packet_content offset1 0x0087 port 1-24 deny  
config access_profile profile_id 1 add access_id 3 packet_content offset1 0x0089 port 1-24 deny  
config access_profile profile_id 1 add access_id 4 packet_content offset1 0x008a port 1-24 deny  
config access_profile profile_id 1 add access_id 5 packet_content offset1 0x01bd port 1-24 deny
```


PCF ACL в коммутаторах серии DGS-3600.

Принцип работы Packet content filtering (PCF) на серии DGS-36xx виден из описания синтаксиса команд CLI, используемых при создании правил:

```
create access_profile packet_content_mask { offset_chunk_1 <value 0-31> <hex 0x0-0xffffffff> |  
offset_chunk_2 <value 0-31> <hex 0x0-0xffffffff> | offset_chunk_3 <value 0-31> <hex 0x0-0xffffffff> |  
offset_chunk_4 <value 0-31> <hex 0x0-0xffffffff>} profile_id <value 1-14>
```

```
config access_profile profile_id <value 1-14> add access_id [auto_assign | <value 1-128>]  
packet_content {offset_chunk_1 <hex 0x0-0xffffffff> | offset_chunk_2 <hex 0x0-0xffffffff> |  
offset_chunk_3 <hex 0x0-0xffffffff> | offset_chunk_4 <hex 0x0-0xffffffff>} port [<portlist> | all]  
[permit {priority <value 0-7> {replace_priority} | rx_rate [no_limit | <value 1-156249>]} |  
replace_dscp <value 0-63> | counter [enable | disable]} | mirror {group_id <value 1-4>} | deny]
```

Как видно, при помощи **4-х ячеек chunk** можно анализировать первые **128 байт** пакета.

Каждая ячейка chunk позволяет осуществлять выборку до **4-х байт**.

Соответствие каждого из 128 байт пакета номеру chunk приведено на следующем слайде.

PCF ACL в коммутаторах серии DGS-3600.

Соответствие каждого из 128 байт пакета номеру chunk:






<u>chunk0</u>	<u>chunk1</u>	<u>chunk2</u>	<u>chunk3</u>	<u>chunk4</u>	<u>chunk5</u>	<u>chunk6</u>	<u>chunk7</u>
b126	b2	b6	b10	b14	b18	b22	b26
b127	b3	b7	b11	b15	b19	b23	b27
b0	b4	b8	b12	b16	b20	b24	b28
b1	b5	b9	b13	b17	b21	b25	b29
<u>chunk8</u>	<u>chunk9</u>	<u>chunk10</u>	<u>chunk11</u>	<u>chunk12</u>	<u>chunk13</u>	<u>chunk14</u>	<u>chunk15</u>
b30	b34	b38	b42	b46	b50	b54	b58
b31	b35	b39	b43	b47	b51	b55	b59
b32	b36	b40	b44	b48	b52	b56	b60
b33	b37	b41	b45	b49	b53	b57	b61
<u>chunk16</u>	<u>chunk17</u>	<u>chunk18</u>	<u>chunk19</u>	<u>chunk20</u>	<u>chunk21</u>	<u>chunk22</u>	<u>chunk23</u>
b62	b66	b70	b74	b78	b82	b86	b90
b63	b67	b71	b75	b79	b83	b87	b91
b64	b68	b72	b76	b80	b84	b88	b92
b65	b69	b73	b77	b81	b85	b89	b93
<u>chunk24</u>	<u>chunk25</u>	<u>chunk26</u>	<u>chunk27</u>	<u>chunk28</u>	<u>chunk29</u>	<u>chunk30</u>	<u>chunk31</u>
b94	b98	b102	b106	b110	b114	b118	b122
b95	b99	b103	b107	b111	b115	b119	b123
b96	b100	b104	b108	b112	b116	b120	b124
b97	b101	b105	b109	b113	b117	b121	b125

Запрет SMB трафика с помощью PCF. Пример 5.

Задача : Заблокировать SMB трафик (порты 135, 137, 138, 139, 445) на физических портах коммутатора 1-24.

Вновь рассмотрим пакет, взятый из Ethereal:

0000	0012	3456	7890	0022	b0de	0f4e	0800	4500
0010	0034	fe74	4000	4006	b945	c0a8	00dd	c0a8
0020	00dc	8f36	008b	baee	6042	0a54	36b8	8010
0030	005c	68fd	0000	0101	080a	0028	9e33	0000
0040	0000							

	destination_mac		L2		L4
	source_mac		L3		

Информация о порте назначения содержится в байтах 24h и 25h (36 и 37 в десятичной СС соответственно). Обратимся к таблице, приведенной выше. Байты 36 и 37 входят в chunk9. Создадим профиль на коммутаторе:

Запрет SMB трафика с помощью PCF. Пример 5.

Создадим профиль на коммутаторе:

```
create access_profile profile_id 1 packet_content_mask offset_chunk_1 9 0x0000ffff
```

Приведенная выше команда означает, что первым обрабатываемым chunk (offset_chunk_1) в нашем случае будет chunk9. Маску значений задаем равной 0x0000ffff, это означает, что в создаваемых в этом профиле правилах, манипулировать мы будем 36 и 37 байтами (вторая половина chunk).

Далее создадим правило в нашем профиле:

```
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x1BD port 1-24 deny
```

В этом правиле указан номер созданного выше профиля (profile_id 1). Значение, принимаемое обрабатываемым нами offset_chunk_1, задаем равным 0x008b (139 порт в десятичной СС). Физические порты коммутатора для которых 139 порт будет блокироваться: 1-24.

По аналогии создадим правила, блокирующие другие порты: 135 (87h), 137 (89h), 138 (8Ah), 445 (1BDh).

Окончательный вид правил для блокировки SMB выглядит следующим образом:

```
create access_profile profile_id 1 packet_content_mask offset_chunk_1 9 0x0000ffff  
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x87 port 1-24 deny  
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x89 port 1-24 deny  
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x8A port 1-24 deny  
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x8B port 1-24 deny  
config access_profile profile_id 1 add access_id auto_assign packet_content offset_chunk_1 0x1BD port 1-24 deny
```

Приоритезация трафика с помощью ACL.

1. Приоритизировать можно любой трафик, попадающий под правило ACL. Таким образом, первым шагом является создание “IP профиля доступа”.
2. Следующим шагом является написание “Правило IP профиля доступа”. При попадании пакета под правило, мы можем:
 - проассоциировать пакет с очередью приоритетов 802.1p (параметр ***priority***);
 - заменить значение 802.1p перед передачей пакета далее (параметр ***replace_priority***);
 - задать пакету новое значение DSCP (параметр ***replace_dscp_with***).
3. Если пакет проассоциирован с очередью приоритетов 802.1p, он, затем, будет обработан в соответствии с “Пользовательским приоритетом 802.1p” для проведения соответствия приоритета 802.1p одной из очередей приоритетов.

Приоритезация трафика с помощью ACL.

Rule Detail	
(Keep an input field as blank to treat the corresponding option as do not care)	
Access ID (1-65535)	1 <input type="checkbox"/> Auto Assign
Source MAC Address	<input type="text"/> e.g.(00-00-00-00-FF-FF)
Source MAC Mask	<input type="text"/> e.g.(00-00-00-00-FF-FF)

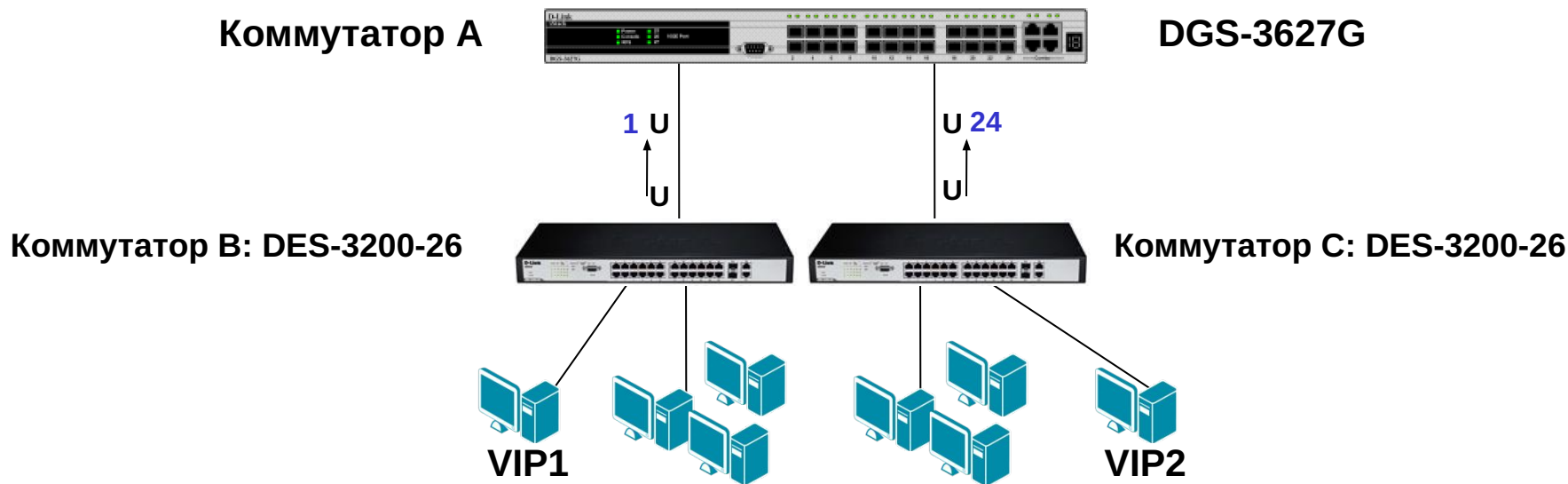
Rule Action	
Action	Permit <input type="button" value="v"/>
Priority (0-7)	<input type="text"/> <input type="checkbox"/>
Replace Priority	<input type="checkbox"/>
Replace DSCP (0-63)	<input type="text"/> <input type="checkbox"/>

Rule Detail	
(Keep an input field as blank to treat the corresponding option as do not care)	
Access ID (1-65535)	1 <input type="checkbox"/> Auto Assign
Source IP Address	<input type="text"/> e.g.(192.168.1.10)
Source IP Mask	<input type="text"/> e.g.(192.168.1.10)

Rule Action	
Action	Permit <input type="button" value="v"/>
Priority (0-7)	<input type="text"/> <input type="checkbox"/>
Replace Priority	<input type="checkbox"/>
Replace DSCP (0-63)	<input type="text"/> <input type="checkbox"/>

Rule Detail	
(Keep an input field as blank to treat the corresponding option as do not care)	
Time Range Name	<input type="text"/> <input type="checkbox"/>
Counter	Disabled <input type="button" value="v"/>
Ports	<input type="text"/> e.g.(1,4-6,9)

Приоритезация трафика с помощью ACL.



Пример: Промаркировать пакеты с определённым DSCP определённым приоритетом 802.1p и поставить в соответствующую очередь

Последующие правила промаркируют пакеты следующим образом:

Очередь 1 - данные с dscp = 10 = приоритет 802.1p = 3

Очередь 2 – данные с dscp = 20 = приоритет 802.1p = 5

Очередь 3 – данные с dscp = 30 = приоритет 802.1p = 7

```
create access_profile ip dscp profile_id 10
```

```
config access_profile profile_id 10 add access_id 10 ip dscp 30 port 1 permit priority 7 replace_priority
```

```
config access_profile profile_id 10 add access_id 20 ip dscp 30 port 24 permit priority 7 replace_priority
```

```
config access_profile profile_id 10 add access_id 30 ip dscp 20 port 1 permit priority 5 replace_priority
```

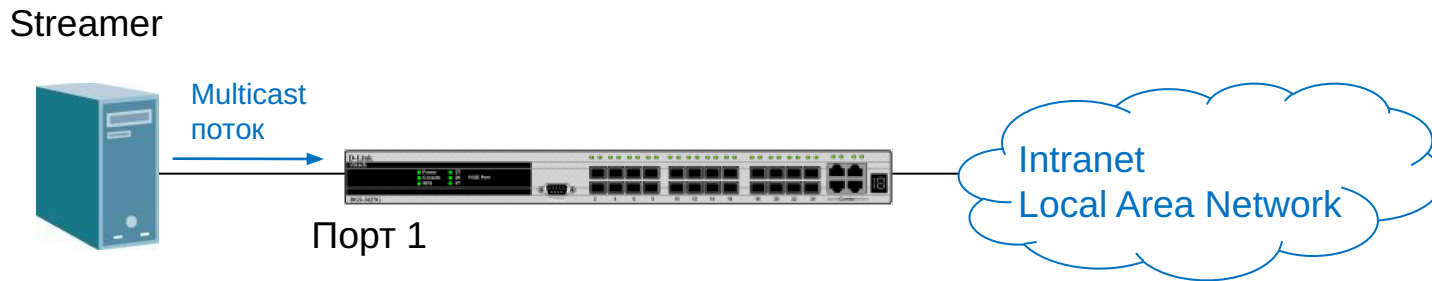
```
config access_profile profile_id 10 add access_id 40 ip dscp 20 port 24 permit priority 5 replace_priority
```

```
config access_profile profile_id 10 add access_id 50 ip dscp 10 port 1 permit priority 3 replace_priority
```

```
config access_profile profile_id 10 add access_id 60 ip dscp 10 port 24 permit priority 3 replace_priority
```

Основываясь на соответствии “802.1p User Priority” пакет будет поставлен в очередь с наивысшим приоритетом и будет обработан первым.

Приоритезация IPTV трафика с помощью ACL.



Задача:

Промаркировать Multicast трафик на входе с помощью ACL.

Создадим сначала профиль и правило, согласно оговоренному ранее условию:

```
create access_profile profile_id 1 ip destination_ip_mask 240.0.0.0
config access_profile profile_id 1 add access_id 1 ip destination_ip 224.0.0.0 port 1 permit priority 4 replace_priority
```

Результат:

Коммутатор маркирует Multicast трафик, обрабатывает его в соответствующей очереди и далее пробрасывает с заданной меткой 802.1p.

Если в сети нет VoIP сервиса, то Multicast трафик можно промаркировать QoS 5, если есть, тогда рекомендуется

VoIP трафик маркировать QoS 5, а Multicast трафик QoS 4.

Контроль полосы пропускания с помощью ACL.

- **Задача:**

Ограничить весь Intranet трафик полосой пропускания в 20Mbps для подсети 192.168.1.x/24 с помощью ACL.

- **Создаем профиль и правило с ограничением:**

```
create access_profile profile_id 1 ip destination_ip_mask 255.255.255.0
```

```
config access_profile profile_id 1 add access_id 1 ip destination_ip 192.168.1.0 port 1-24 permit rx_rate 320
```

- **Проверка настроек с помощью программы iperf:**

```
iperf -c 192.168.1.20 -i 1 -t 10
```

```
-----  
Client connecting to 192.168.1.20, TCP port 5001
```

```
TCP window size: 8.00 KByte (default)  
-----
```

```
[1912] local 192.168.1.15 port 2870 connected with 192.168.1.20 port 5001
```

[ID]	Interval	Transfer	Bandwidth
[1912]	0.0- 1.0 sec	3.35 MBytes	28.1 Mbits/sec
[1912]	1.0- 2.0 sec	2.09 MBytes	17.5 Mbits/sec
[1912]	2.0- 3.0 sec	2.58 MBytes	21.6 Mbits/sec
[1912]	3.0- 4.0 sec	2.53 MBytes	21.2 Mbits/sec
[1912]	4.0- 5.0 sec	2.34 MBytes	19.6 Mbits/sec
[1912]	5.0- 6.0 sec	2.02 MBytes	17.0 Mbits/sec
[1912]	6.0- 7.0 sec	2.82 MBytes	23.7 Mbits/sec
[1912]	7.0- 8.0 sec	2.05 MBytes	17.2 Mbits/sec
[1912]	8.0- 9.0 sec	2.60 MBytes	21.8 Mbits/sec
[1912]	9.0-10.0 sec	1.92 MBytes	16.1 Mbits/sec
[1912]	0.0-10.1 sec	24.3 MBytes	20.2 Mbits/sec

* 320 шагов * 64kbps = 20480kbps = **20Mbps**

CPU Interface Filtering

CPU Interface Filtering

- CPU Interface Filtering, или иначе Software ACL – это списки доступа, предназначенные для фильтрации пакетов, которые не могут быть отброшены аппаратными ACL.
- К таким пакетам относится трафик, обрабатываемый CPU коммутатора:
 - трафик управления (telnet, SSH), а также весь трафик к System интерфейсу
 - SNMP
 - широковещательный трафик во VLAN, в котором находится управляющий интерфейс
 - многоадресная рассылка (multicast)
- Рекомендуется применять для снижения загрузки CPU, в случаях «подвисаний» управления и для фильтрации нежелательных многоадресных рассылок.

Пример использования CPU Interface Filtering

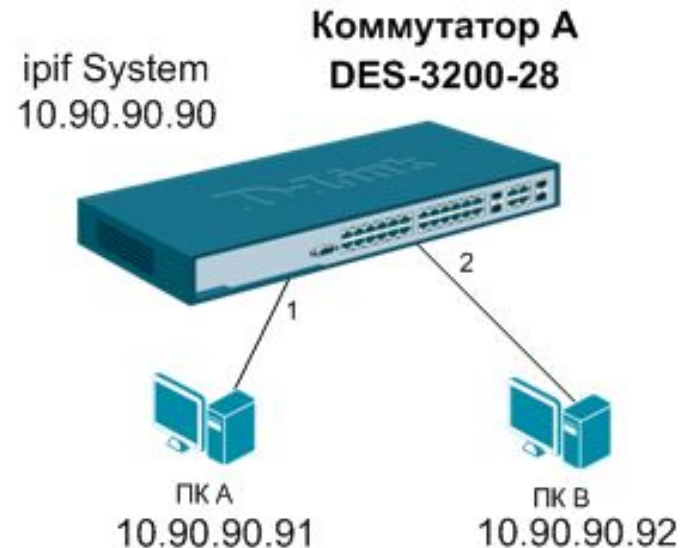
- Необходимо ограничить доступ к System интерфейсу коммутатора
- ПК А видит по ICMP ПК В и не видит ipif System
- Настройка коммутатора:

```
enable cpu_interface_filtering
```

```
create cpu_access_profile profile_id 1 ip source_ip_mask 255.255.255.128 icmp
```

```
config cpu_access_profile profile_id 1 add access_id 1 ip source_ip 10.90.90.91 icmp port 1 deny
```

- Интерфейс коммутатора System огражден от ICMP пакетов ПК А
- Точно так же можно запретить любой вид трафика.
- CPU access_profile не отображаются в общем списке ACL, посмотреть их можно командой **show cpu access_profile**



Safeguard Engine

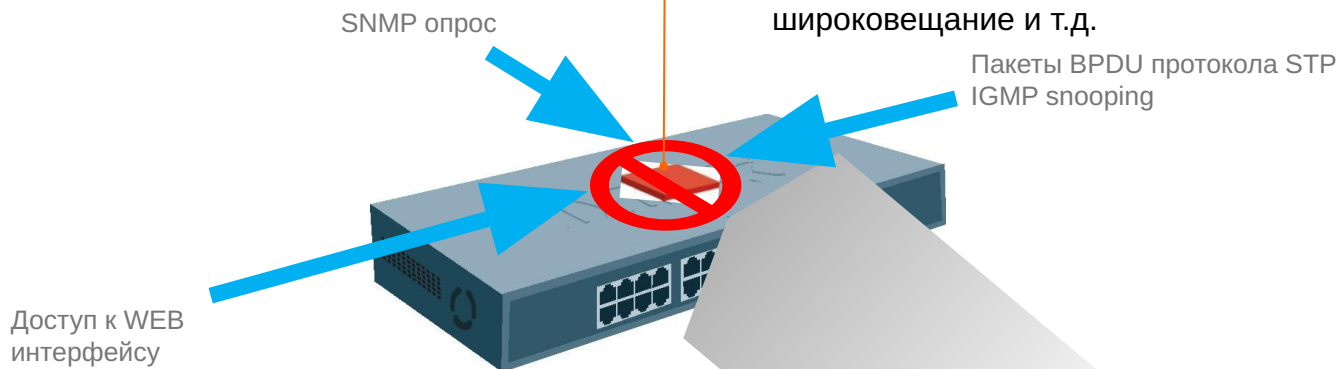
Почему Safeguard Engine?

Safeguard Engine™ разработан для того, чтобы повысить надёжность новых коммутаторов и общую доступность и отказоустойчивость сети.

Весь этот трафик загружает CPU и не дает ему возможности обрабатывать более важные задачи, такие как административный доступ, STP, SNMP опрос.

CPU коммутатора предназначен для обработки управляющей информации, такой как STP, SNMP, доступ по WEB-интерфейсу и т.д.

Также CPU обрабатывает некоторый специфичный трафик, такой как ARP широковещание, пакеты с неизвестным IP-адресом назначения, IP широковещание и т.д.



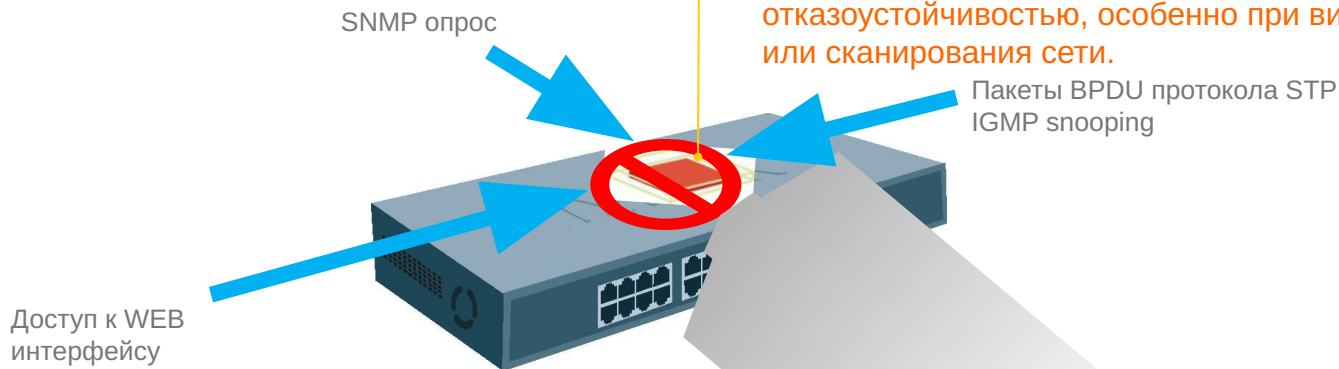
Но в современных сетях достаточно много вирусов и вредоносного трафика. Обычно они генерируют много «интересного» для CPU трафика (такого как ARP широковещание например).

ARP широковещание
Пакеты с неизвестным IP-адресом назначения
IP широковещание

Почему Safeguard Engine?

Safeguard Engine™ разработан для того, чтобы повысить надёжность новых коммутаторов и общую доступность и отказоустойчивость сети.

Весь этот трафик загружает CPU и не даёт ему возможности обрабатывать более важные задачи, такие как административный доступ, STP, SNMP опрос.



D-Link Safeguard Engine позволяет идентифицировать и приоритезировать этот «интересный» для CPU трафик с целью отбрасывания ненужных пакетов для сохранения функциональности коммутатора.

Таким образом с применением Safeguard Engine, коммутатор D-Link будет обладать отказоустойчивостью, особенно при вирусных атаках или сканирования сети.

Но в современных сетях достаточно много вирусов и вредоносного трафика. Обычно они генерируют много «интересного» для CPU трафика (такого как ARP широковещание например).

ARP широковещание
Пакеты с неизвестным IP-адресом назначения
IP широковещание

Safeguard Engine

- Если загрузка CPU становится выше порога **Rising Threshold** (20-100%), коммутатор войдёт в **Exhausted Mode** (режим высокой загрузки).
- Если загрузка CPU становится ниже порога **Falling Threshold** (20-100%), коммутатор выйдет из Exhausted Mode и механизм Safeguard Engine отключится.
- Действия коммутатора при работе Safeguard Engine:
 - **Ограничение полосы пропускания для широковещательных ARP-пакетов**
Strict-mode – коммутатор перестает получать arp пакеты
Fuzzy-mode – ограничивается полоса пропускания для arp пакетов
 - **Ограничение полосы пропускания для широковещательных IP пакетов**
Strict-mode – коммутатор перестает получать все широковещательный IP пакеты
Fuzzy-mode – динамически ограничивается полоса пропускания для широковещательных IP пакетов
- Interval - Удвоенного времени переключения в Exhausted режим:
 - При использовании "Удвоенного времени переключения в Exhausted режим", коммутатор может избежать постоянного переключения в exhausted mode без надобности.
 - Максимальное значение этого времени - 320 секунд. В ситуации, когда коммутатор постоянно входит в exhausted mode, и когда это время достигает максимального значения, коммутатор не выйдет за это значение.

Пример использования функции Safeguard Engine

IP-адрес коммутатора:
10.31.3.254/8



PC2

IP-адрес PC2: 10.31.3.2/8

1. PC2 постоянно посылает ARP-пакеты, например со скоростью 1000 пакетов в секунду.
2. Загрузка CPU при этом изменяется от нормальной до 100%.
3. Если прекратить генерацию ARP пакетов на PC2, загрузка CPU опять станет в пределах нормы.
4. Настройки коммутатора:
`config safeguard_engine state enable`
`config safeguard_engine utilization rising 80 falling 50`

Задача: Снизить загрузку CPU при помощи Safeguard Engine.

Пример использования функции Safeguard Engine

```
DES-3200-28:4#show safeguard_engine
```

```
Command: show safeguard_engine
```

```
Safe Guard Engine State      : Enabled
```

```
Safe Guard Engine Current Status : Normal mode
```

```
=====
```

```
CPU utilization information:
```

```
Interval                    : 5 sec
```

```
Rising Threshold(20-100)   : 80 %
```

```
Falling Threshold(20-100)  : 50 %
```

```
Trap/Log                    : Disabled
```

Пример использования функции Safeguard Engine

Результаты теста:

- Перед активацией Safeguard Engine, при генерации PC2 большого количества ARP пакетов, загрузка CPU будет держаться в районе 100%.
- После включения функции Safeguard Engine, PC2 продолжает генерировать большое количество ARP пакетов. Загрузка CPU снизится до значения нижнего предела и коммутатор будет держать интервал между переключениями 5 секунд (значение по умолчанию).

Вывод:

Функция SafeGuard Engine функционирует следующим образом. При превышении загрузкой CPU верхнего предела, коммутатор отбрасывает все ARP пакеты. При значении загрузки между двумя пределами, коммутатор обрабатывает только ARP пакеты, предназначенные ему. При снижении загрузки ниже нижнего предела коммутатор обрабатывает все ARP пакеты.

Возможные побочные эффекты

- После того как коммутатор переключится в режим `exhausted` при настроенном строгом режиме, административный доступ к коммутатору будет недоступен, так как в этом режиме отбрасываются все ARP-запросы. В качестве решения можно предложить указать MAC-адрес коммутатора в статической ARP-таблице управляющей рабочей станции, для того чтобы она могла напрямую обратиться к интерфейсу управления коммутатором без отсылки ARP-запроса.
- **Для коммутаторов L2/L3, переход в режим `exhausted` не будет влиять на коммутацию пакетов на уровне L2.**
- Для коммутатора L3, при переходе в строгий режим `exhausted`, не только административный доступ будет недоступен, но и связь между подсетями может быть нарушена тоже, поскольку будут отбрасываться ARP-запросы на IP-интерфейсы коммутатора тоже.
- Преимуществом нестрогого режима `exhausted` является то, что в нём он не просто отбрасываются все ARP-пакеты или пакеты IP-широковещания, а динамически изменяется полоса пропускания для них. Таким образом даже при серьёзной вирусной эпидемии, коммутатор L2/L3 будет доступен по управлению, а коммутатор L3 сможет обеспечивать взаимодействие между подсетями.

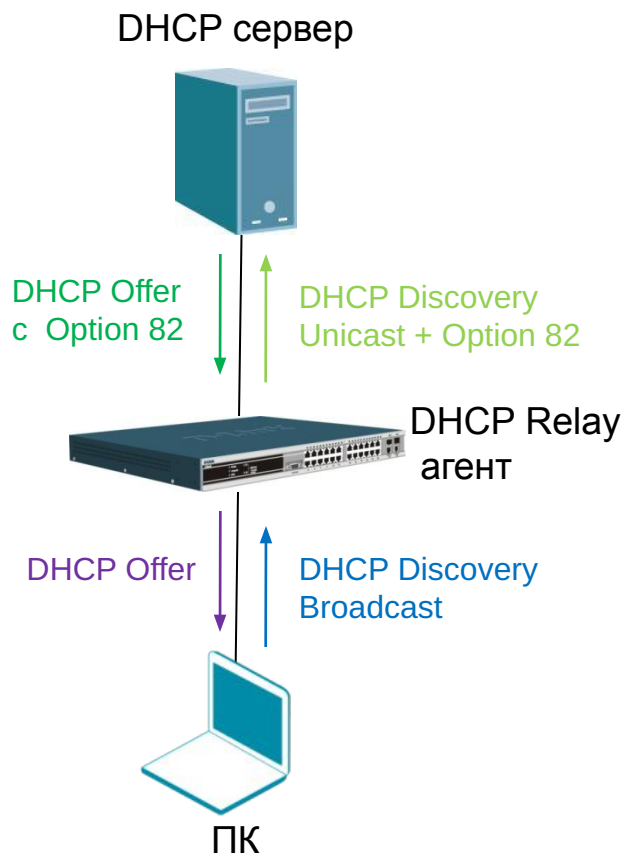
DHCP Relay Option 82 – информация от агента DHCP Relay

Информация DHCP Relay Agent (Option 82)

- Option 82 используется Relay Agent (агентом перенаправления запросов) для добавления дополнительной информации в DHCP – запрос клиента. Эта информация может быть использована для применения политик, направленных на увеличение уровня безопасности и эффективности сети.
- Она описана в стандарте RFC 3046.

Пример работы DHCP Relay Agent и добавление Option 82

Когда вы включаете опцию DHCP Relay Agent Option 82 на коммутаторе D-link, происходит следующее:



- 1 Компьютер в сети (DHCP - клиент) генерирует DHCP - запросы и широковещательно рассылает их в сеть.
- 2 Коммутатор (DHCP Relay Agent) перехватывает DHCP - запрос packet и добавляет в него информацию Relay Agent Information Option (Option 82). Эта информация содержит MAC – адрес коммутатора (поле опции Remote ID) и VLAN ID, в котором находится DHCP - клиент и SNMP ifindex порта, с которого получен запрос (поле опции Circuit ID).
Коммутатор перенаправляет DHCP - запрос с полями опции Option 82 на DHCP - сервер.
- 3 DHCP - сервер получает пакет. Если сервер поддерживает опцию Option 82, он может использовать поля Remote ID и/или Circuit ID для назначения IP-адреса и применения политик, таких как ограничения количества IP-адресов, выдаваемых одному Remote ID или Circuit ID. Затем DHCP сервер копирует поле Option 82 в DHCP - ответе. Если сервер не поддерживает Option 82, он игнорирует поля этой опции и не отправляет их в ответе.
DHCP - сервер отвечает в Unicast-е агенту перенаправления запросов. Агент проверяет предназначен ли он его клиенту, путём анализа IP - адреса назначения пакета.
- 4 Агент удаляет поля Option 82 и направляет пакет на порт, к которому подключён DHCP - клиент, пославший пакет DHCP - запроса.

Примеры конфигурации DHCP Relay

DHCP Relay per interface:

```
config dhcp_relay add ipif System <ipaddr>
```

На L2 коммутаторах это всегда ipif System, на L3 коммутаторах нужно указывать клиентский интерфейс, т.е. тот, который принимает DHCP Discovery пакеты от клиента.

DHCP Relay per VLAN:

```
config dhcp_relay add vlanid <vlan_id_list> <ipaddr>
```

Указываются VLAN-ы, в которых будет работать функция DHCP Relay.

<ipaddr> - задается IP адрес DHCP сервера, на который пересылаются DHCP пакеты от клиентов.

DHCP Relay per port:

```
config dhcp_relay ports <portlist> state enable
```

Используется для того, чтобы коммутатор не реагировал на транзитные unicast DHCP пакеты.

Формат полей DHCP Option 82

Поле опции DHCP Option 82 имеет следующий формат :

Формат поля опции Circuit

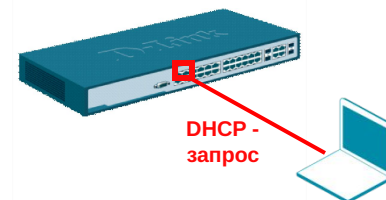
ID:	1.	2.	3.	4.	5.	6.	7.
	1	6	0	4	VLAN	Modul	Port
	1 байт	1 байт	1 байт	1 байт	2 байта	1 байт	1 байт

1. Тип подопции
2. Длина: длина поля с октета 3 по октет 7
3. Тип Circuit ID
4. Длина: длина поля с октета 5 по октет 7
5. VLAN: номер VLAN ID в DHCP – пакете клиент.
6. Модуль: Для отдельно стоящего коммутатора, поле Модуль всегда равно 0; Для коммутатора в стеке, поле Модуль это Unit ID.
7. Порт: номер порта, с которого получен DHCP - запрос, номер порта начинается с 1.

Формат поля опции Remote

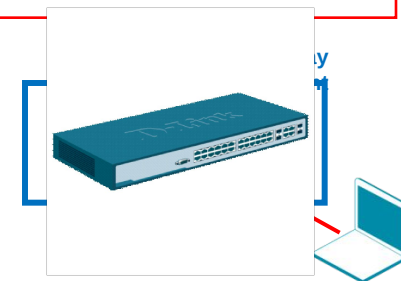
ID:	1.	2.	3.	4.	5.
	2	8	0	6	MAC address
	1 байт	1 байт	1 байт	1 байт	6 байт

1. Тип подопции
2. Длина
3. Тип Remote ID
4. Длина
5. MAC-адрес: MAC-адрес коммутатора.



С какого порта
получен
DHCP - запрос

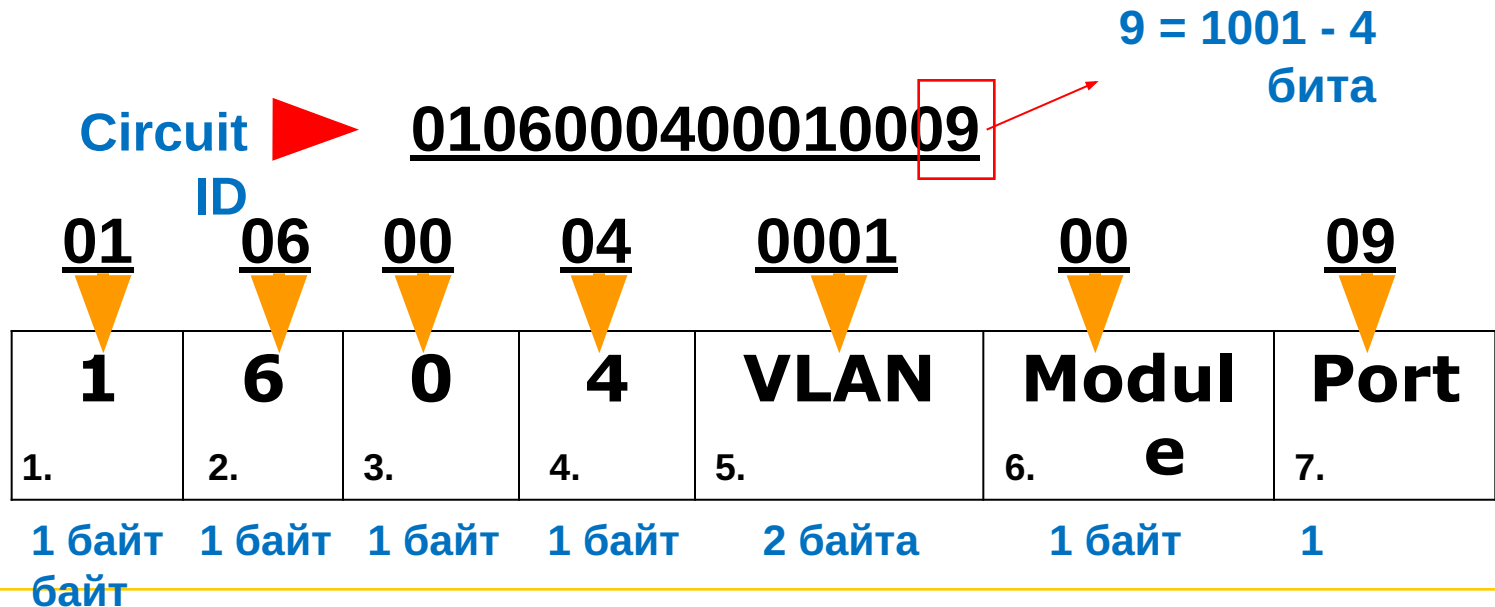
Локальный идентификатор агента,
который получил DHCP – пакет от клиента.



Для идентификации удалённого узла.
DHCP – сервер может использовать
эту
опцию для выбора специфических
параметров пользователей, узлов.

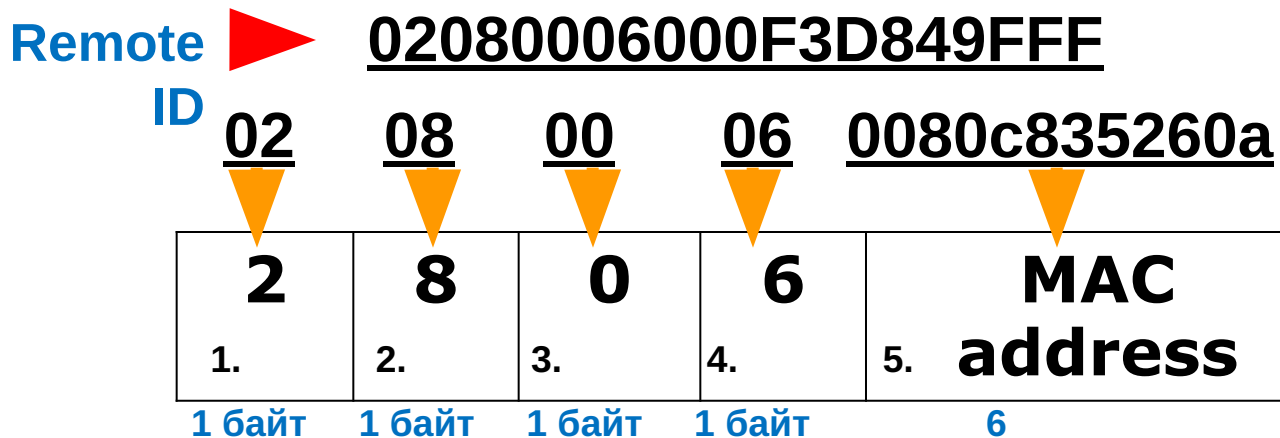
Поле
remote ID должно быть уникально в
сети.

Формат поля опции Circuit ID



1. Тип подопции - **01** (подопция Agent Circuit ID)
2. Длина - **06**
3. Тип Circuit ID - **00**
4. Длина - **04**
5. VLAN: VLAN ID в DHCP – пакете клиента. - **0001**
6. Модуль: Для отдельно стоящего коммутатора, поле Модуль всегда равно 0; Для коммутатора в стеке, поле Модуль это Unit ID. - **00**
7. Порт: номер порта, с которого получен DHCP – пакет клиента, номер порта начинается с 1. - **09**

Формат поля опции Remote ID



- | | | | | |
|----|------------------------------------|---|---------------------|----------------------------|
| 1. | Тип подопции | - | <u>02</u> | (подопция Agent Remote ID) |
| 2. | Длина | - | <u>08</u> | |
| 3. | Тип Remote ID | - | <u>00</u> | |
| 4. | Длина | - | <u>06</u> | |
| 5. | MAC-адрес : MAC-адрес коммутатора. | - | <u>0080C835260A</u> | |

Circuit

(0106)000400010009

+

(0208)00060080c835260a

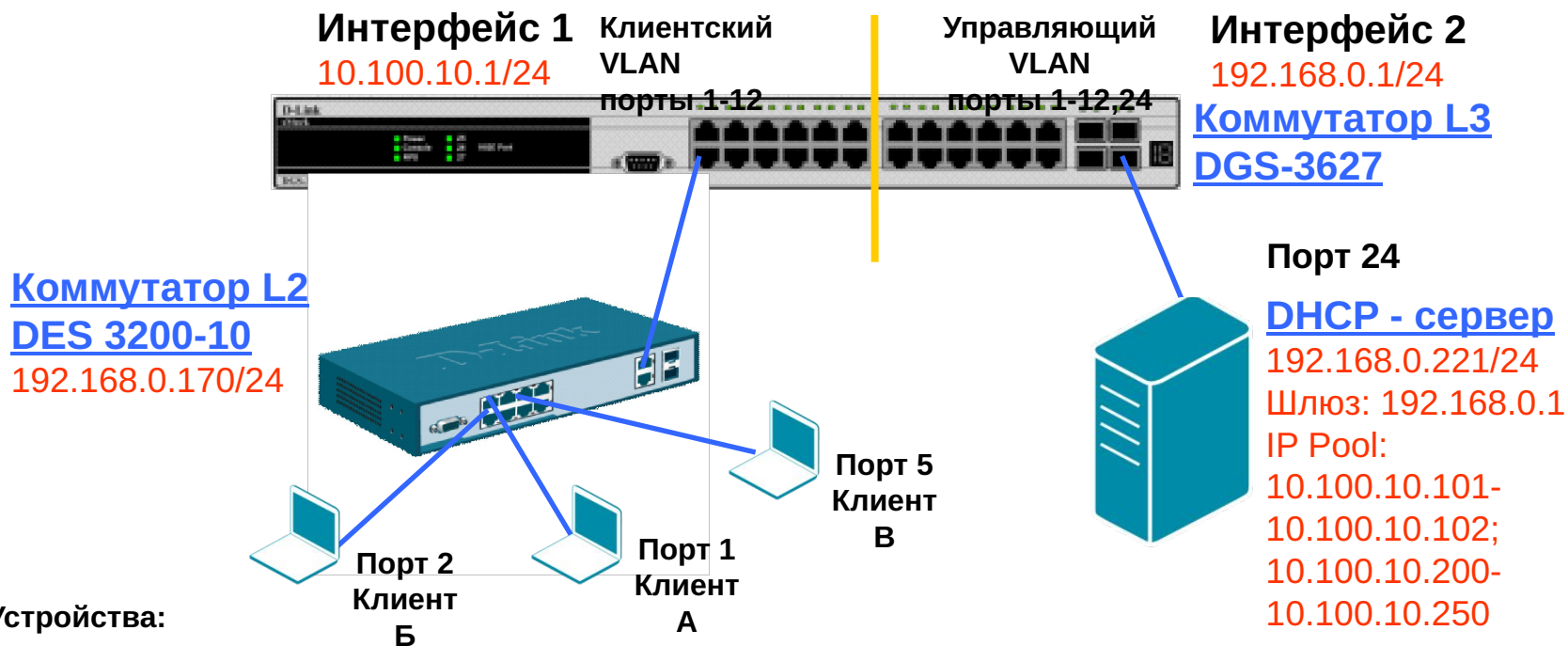
Remote ID



00040001000900060080c835260a

DHCP – сервер назначит определённый IP-адрес, исходя из этой информации

Пример настройки и использования Option 82



Устройства:

1. DHCP - сервер 192.168.0.221 в подсети 192.168.0.0/24
2. Маршрутизатор или коммутатор L3, выступающий в роли шлюза для 2-ух подсетей
192.168.0.1 в подсети 192.168.0.0/24 10.100.10.1 в подсети 10.100.10.0/24
3. Коммутатор L2 (DES-3200-10) выступает в роли DHCP Relay Agent 192.168.0.170 в подсети 192.168.0.0/24
MAC – адрес 00-24-01-FC-8F-D8
4. 3 ноутбука, выступающих в роли DHCP – клиентов, подключённых к коммутатору L2 – порты 1, 2 и 5

Сервер с поддержкой DHCP Option 82

- DHCP – сервер использует динамический пул IP-адресов 10.100.10.200 – 10.100.10.250 для назначения IP-адресов любому DHCP – клиенту, запрос от которого будет перенаправлен DHCP Relay Agent-ом 192.168.0.170 (Если DHCP – клиент, подключён к любому порту коммутатора, кроме портов 1 и 2, он получит IP-адрес из пула.)

--- Для обычного DHCP – запроса клиента

- Когда какой-либо DHCP – клиент подключается к порту 1 коммутатора L2, DHCP – сервер выдаст ему IP-адрес 10.100.10.101; когда DHCP – подключается к порту 2 коммутатора L2, DHCP – сервер выдаст ему IP-адрес 10.100.10.102. (например, DHCP – клиент, подключённый к порту 1 коммутатора, получит IP-адрес 10.100.10.101)

--- Для DHCP – запросов клиента с option 82

Настройка L3 коммутатора

Настройка коммутатора L3 (DGS-3627):

Настройте влан, в котором будут находиться DHCP – клиенты

```
create vlan client tag 555
```

```
config vlan client add tagged 1-12
```

Настройте управляющий влан, в котором будет находиться DHCP сервер

```
create vlan management tag 1234
```

```
config vlan management add tagged 1-12
```

```
config vlan default delete 24
```

```
config vlan management add untagged 24
```

Сконфигурируйте и создайте IP-интерфейсы в VLAN client и management

```
config ipif System ipaddress 10.90.90.90/24
```

```
create ipif client_gw 10.100.10.1/24 client state enable
```

```
create ipif manag_gw 192.168.0.1/24 management state enable
```

Сохраните настройки

```
save
```

Настройка L2 коммутатора

Настройка коммутатора L2 (DES-3200-10):

Настройте клиентский и управляющий вланы на DES-3200-10

```
config vlan default delete 1-8
```

```
create vlan client tag 555
```

```
config vlan client add tagged 9-10
```

```
config vlan client add untagged 1-8
```

```
create vlan management tag 1234
```

```
config vlan management add tagged 9-10
```

Настройте управляющий интерфейс

```
config ipif System ipaddress 192.168.0.170/24 vlan management
```

Настройте DHCP Relay

```
enable dhcp_relay
```

```
config dhcp_relay option_82 state enable
```

```
config dhcp_relay option_82 check disable
```

```
config dhcp_relay option_82 policy replace
```

```
config dhcp_relay option_82 remote_id default
```

```
config dhcp_relay add ipif System 192.168.0.221
```

Разрешите клиентам доступ в управляющем влане, только к DHCP серверу. Остальное запретите

```
create access_profile ip destination_ip 255.255.255.255 profile_id 5
```

```
config access_profile profile_id 5 add access_id 1 ip destination_ip 192.168.0.221 port 1-8 permit
```

```
create access_profile ip destination_ip 255.255.255.0 profile_id 6
```

```
config access_profile profile_id 6 add access_id 1 ip destination_ip 192.168.0.0 port 1-8 deny
```

Сохраните настройки

```
save
```

Настройка DHCP сервера

Рассмотрим пример настройки сервера `isc-dhcpd`.
Ниже приведено содержимое `dhcpd.conf`:

Настройка основных параметров

```
lease-file-name "/var/log/dhcpd.leases";
log-facility local7;
authoritative;
default-lease-time 86400;
ddns-update-style none;
local-address 192.168.0.221;
one-lease-per-client true;
deny duplicates;
```

Настройка логирования (в лог записываются MAC адрес, влан и порт клиента, запросившего IP адрес)

```
if exists agent.circuit-id {
log(info, concat("Lease", " IP ", binary-to-ascii(10, 8, ".", leased-address),
" MAC ", binary-to-ascii(16, 8, ":", substring(hardware, 1, 6)),
" port ", binary-to-ascii(10, 16, "", substring(option agent.circuit-id, 4,
2)),
" VLAN ", binary-to-ascii(10, 16, "", substring(option agent.circuit-id, 2, 2))
));
}
```

Сравниваются Remote ID и Circuit ID с заданными. Согласно дизайну преобразования `binary-to-ascii` незначащие нули слева отбрасываются

```
class "sw170-1" {
match if binary-to-ascii(16, 8, ":", suffix(option agent.remote-id, 5))
= "24:1:fc:8f:d8" and binary-to-ascii(10, 8, "", suffix(option
agent.circuit-id, 1)) = "1";
}
class "sw170-2" {
match if binary-to-ascii(16, 8, ":", suffix(option agent.remote-id, 5))
= "24:1:fc:8f:d8" and binary-to-ascii(10, 8, "", suffix(option
agent.circuit-id, 1)) = "2";
}
```


Настройка DHCP сервера

Продолжение содержимого файла dhcpd.conf:

```
shared-network test {  
# Включить опцию, позволяющую клиенту корректно продлевать аренду IP адреса прямым запросом на сервер , не  
содержащим Option 82 (минуя DHCP Relay Agent)  
stash-agent-options true;  
# Запретить выдавать IP-адреса из подсети 192.168.0.0/24 (в этой подсети находятся управляющие интерфейсы коммутаторов и  
доступ клиентов в эту подсеть должен быть ограничен)  
subnet 192.168.0.0 netmask 255.255.255.0 {  
deny unknown-clients;  
}  
# Описать выдаваемые клиенту по DHCP параметры  
subnet 10.100.10.0 netmask 255.255.255.0 {  
option broadcast-address 10.100.10.255;  
option domain-name-servers 10.100.10.1;  
option routers 10.100.10.1;  
option subnet-mask 255.255.255.0;  
# Задать адреса, получаемые клиентами :  
# клиентом , подключенным к порту 1  
pool { range 10.100.10.101; allow members of "sw170-1";}   
# клиентом , подключенным к порту 2  
pool { range 10.100.10.102; allow members of "sw170-2";}   
# клиентами, находящимися на других портах  
pool { range 10.100.10.200 10.100.10.250;}  
}  
}
```

Информация DHCP Relay Agent (Option 82)

Результаты теста:

1. Клиенту А будет выдан IP-адрес **10.100.10.101**
2. Клиенту Б будет выдан IP-адрес **10.100.10.102**
3. Клиенту В будет выдан IP-адрес **10.100.10.200**

Функции управления и
отслеживание работы сети:

Протокол SNMP

Протокол SNMP

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) – это протокол 7–ого уровня модели OSI(Уровень приложений) разработан для управления и отслеживания работы устройств сети. SNMP позволяет станциям управления сети прочитать и изменить параметры настройки шлюзов, маршрутизаторов, коммутаторов и других устройств сети. Используйте SNMP, чтобы настраивать устройства для правильного их функционирования, отслеживания работы сети и обнаружения потенциальных проблем на коммутаторе или группе коммутаторов или сети.

SNMP компоненты

□ SNMP менеджер:

SNMP менеджер – это программное приложение, которое контактирует с SNMP агентами, опрашивая или изменяя базу данных агента.

□ SNMP агент:

SNMP агент – это программное обеспечение, которое запущено на сетевом оборудовании(хост, маршрутизатор, принтер или другое оборудование), и поддерживает информацию о ее конфигурации и текущий статус в базе данных.

□ MIB:

Информация в базе данных хранится на основе информации об управлении (MIB), как карта с иерархической последовательностью всех управляемых объектов и деталями, описывающих возможности каждого объекта

Протокол SNMP

Версии протокола SNMP

На данный момент всего три версии протокола SNMP:

- SNMPv1 (RFC 1157),
- SNMPv2c (RFC 1901-1908)
- SNMP v3 (RFC 3411-3418)

В **SNMP v.1** и **v.2c** пользовательское установление подлинности осуществляется с помощью строк сообществ 'community strings', которые функционируют как пароли. Удалённый пользователь SNMP приложения и коммутатор с поддержкой протокола SNMP должны использовать одинаковую строку сообщества. SNMP пакеты от любой станции, которые не прошли проверку на подлинность, будут игнорироваться(отброшены).

На коммутаторе для управления и отслеживания работы сети настроены и используются следующие строки сообществ для **SNMP v.1** и **v.2c** при настройках по умолчанию на коммутаторе:

- **public** - чтение
- **private** – чтение/запись

Внимание: Нужно изменить настройки по умолчанию строк сообществ по причинам безопасности.

SNMP v.3 использует списки пользователей и паролей, которые хранятся и передаются в хешированном виде, что даёт более высокий уровень безопасности при использовании данного протокола.

Протокол SNMP

Management Information Base

- Каждому объекту, которым управляют, назначают идентификатор объекта (OID).
- OID-ы определены в файле MIB.
- OID может быть представлен как последовательность целых чисел, отделенных десятичными запятыми, или текстовой строкой.
- Когда SNMP менеджер опрашивает объект, он посылает OID SNMP агенту.

SNMP менеджер



SNMP get-request



SNMP агент



SNMP get-response



1. SNMP менеджер посылает запрос SNMP агенту:
`snmpget -v2c -c public 192.168.0.128 1.3.6.1.2.1.1.1.0`

2. SNMP агент отвечает SNMP менеджеру:
`SNMPv2-MIB::sysDescr.0 = STRING: D-Link DES-3528 Fast Ethernet Switch`

Протокол SNMP

Основные SNMP команды:

- **snmpget** – для просмотра конкретногоOID-а.
- **snmpwalk** – для просмотра дерева OID-ов.
- **snmpset** – для изменения настроек OID-а(ов).

Внимание: MIB-ы лежат в свободном доступе для коммутаторов или серий коммутаторов на [ftp.dlink.ru](ftp://ftp.dlink.ru) в папочке SNMP, например: <ftp://ftp.dlink.ru/pub/Switch/DES-3528/SNMP/>

Пример:

```
snmpwalk -v2c -c public 192.168.0.128 1.3.6.1.2.1.2.2.1.5
IF-MIB::ifSpeed.1 = Gauge32: 0
IF-MIB::ifSpeed.2 = Gauge32: 0
IF-MIB::ifSpeed.3 = Gauge32: 0
IF-MIB::ifSpeed.4 = Gauge32: 0
IF-MIB::ifSpeed.5 = Gauge32: 0
IF-MIB::ifSpeed.6 = Gauge32: 0
IF-MIB::ifSpeed.7 = Gauge32: 0
IF-MIB::ifSpeed.8 = Gauge32: 0
IF-MIB::ifSpeed.9 = Gauge32: 0
IF-MIB::ifSpeed.10 = Gauge32: 0
IF-MIB::ifSpeed.11 = Gauge32: 0
IF-MIB::ifSpeed.12 = Gauge32: 0
IF-MIB::ifSpeed.13 = Gauge32: 0
IF-MIB::ifSpeed.14 = Gauge32: 0
IF-MIB::ifSpeed.15 = Gauge32: 0
IF-MIB::ifSpeed.16 = Gauge32: 0
IF-MIB::ifSpeed.17 = Gauge32: 0
IF-MIB::ifSpeed.18 = Gauge32: 0
IF-MIB::ifSpeed.19 = Gauge32: 0
IF-MIB::ifSpeed.20 = Gauge32: 0
IF-MIB::ifSpeed.21 = Gauge32: 0
IF-MIB::ifSpeed.22 = Gauge32: 0
IF-MIB::ifSpeed.23 = Gauge32: 0
IF-MIB::ifSpeed.24 = Gauge32: 0
IF-MIB::ifSpeed.25 = Gauge32: 0
IF-MIB::ifSpeed.26 = Gauge32: 0
IF-MIB::ifSpeed.27 = Gauge32: 0
IF-MIB::ifSpeed.28 = Gauge32: 1000000000
```

D-Link 2013 Q4

**Спасибо
за
внимание!**

Бигаров Руслан, менеджер по проектам
e-mail: rbigarov@dlink.ru

