РЕАЛІЗАЦІЯ MULTIWAN ТА ЙОГО МАРШРУТИЗАЦІЯ ЗА ДОПОМОГОЮ MIKROTIK ROUTEROS

Слюсарь Ігор Іванович
к.т.н, доцент, доцент кафедри інформаційних систем та технологій
Полтавської державної аграрної академії

Поліщук Юлій Валентинович
асистент кафедри інформаційних систем та технологій
Полтавської державної аграрної академії

Копішинська Олена Петрівна
к.ф.-м.н., доцент, професор кафедри інформаційних систем та технологій
Полтавської державної аграрної академії

Уткін Юрій Вікторович
к.т.н, доцент, завідувач кафедри інформаційних систем та технологій
Полтавської державної аграрної академії

На сьогоднішній день, в умовах впливу пандемії COVID-19 на функціонування підприємств та установ, які працюють в умовах обмежень, досить критичним є реалізація віддаленої роботи через Інтернет з дотриманням вимог до надійності та безпеки. Найбільш привабливим в цьому відношенні може бути створення з'єднання MultiWAN [1], яке в потенціалі забезпечує:

- резервування підключення до Інтернет через кількох провайдерів;
- розподіл навантаження між кількома провайдерами;
- доступ до маршрутизатора через кілька зовнішніх каналів.

Однак, зазвичай, на підприємствах невеликого розміру здійснюється підключення кількох провайдерів для резервування з'єднання з глобальною мережею без гнучкої маршрутизації та автоматизації процесу переключення між ними. Крім того, використання прямої незахищеної переадресації, наприклад, віддаленого доступу в ОС Windows за протоколом RDP [2], є легкою здобиччю для зловмисників.

Як наслідок, в роботі запропонований варіант MultiWAN, який вільний від вказаних недоліків. Він передбачає забезпечення наступних вимог:

– забезпечення автоматичного перемикання на резервного провайдера;

– передбачення можливості публікації сервісів з локальної мережі (LAN) в Інтернет (за допомогою функції DSTNAT [3]);

– налаштування фільтру «Файрволу» для забезпечення мінімально достатньої безпеки з боку Інтернет;

– забезпечення маршрутизації відповідних пакетів у канал, з якого вони прийшли.

При цьому необхідно враховувати, що при прийнятті рішення про маршрутизацію пакета з маркуванням маршруту, у випадку відсутності або недоступності маршруту з маркуванням, пакет буде оброблений в основній таблиці маршрутизації (main), а керування цим процесом можливо через ip route rule (lookuponly-in-table) [4]. Для підтвердження можливості практичної реалізації такого підходу до організації MultiWAN, в роботі розглянутий приклад конфігурації операційної системи Mikrotik RouterOS.

В якості допущення введено положення, що маршрутизатор має базові налаштування LAN та вихід в глобальну мережу через два канали (WAN), які надають різні провайдери (основний та резервний). Спочатку виконаємо основні налаштування безпеки. Приховуємо маршрутизатор від виявлення сусідства та управління з мереж провайдерів по MAC:

```
/ip neighbor discovery-settings set discover-interface-list=!WAN
/tool mac-server set allowed-interface-list=LAN
```

```
/tool mac-server mac-winbox set allowed-interface-list=LAN
```

Створюємо мінімально достатній набір правил фільтра файрвола для захисту маршрутизатора.

1. Правило забезпечує дозвіл для встановлених і споріднених з'єднань, які ініційовані як з підключених мереж, так і самим маршрутизатором:

```
/ip firewall filter add action=accept chain=input \
comment="Related Established Untracked Allow" \
connection-state=established,related,untracked
```

2. Дозвіл для вхідного трафіку по протоколу ICMP для використання утиліт ping та traceroute:

```
/ip firewall filter add action=accept chain=input \
comment="ICMP from ALL" protocol=icmp
```

3. Правило, яке закриваюче ланцюг input, забороняє все інше, що надходить з Інтернету:

```
/ip firewall filter add action=drop chain=input \
comment="All other WAN Drop" in-interface-list=WAN
```

4. Правило дозволяє встановлені і споріднені з'єднання, які проходять крізь маршрутизатор:

```
/ip firewall filter add action=accept chain=forward \
comment="Established, Related, Untracked allow" \
connection-state=established,related,untracked
```

5. Правило забороняє проходити крізь маршрутизатор пакетам, які йдуть з Інтернет і не пройшли процедуру DSTNAT:

```
/ip firewall filter add action=drop chain=forward \
comment="Drop all from WAN not DSTNATed" connection-nat-state=!dstnat\
connection-state=new in-interface-list=WAN
```

Це вбереже локальні мережі від зловмисників, які, перебуваючи в одному широкомовному домені з нашими зовнішніми мережами, встановлять в якості шлюзу наші зовнішні IP і так спробують «дослідити» локальну мережу.

Найважливішим завданням реалізації MultiWAN є коректна маршрутизація трафіку, а саме: незалежно від того, в який канал провайдера налаштовано маршрут за замовчуванням, він повинен повертати відповідь саме

в той канал, з якого пакет прийшов. У простій локальної мережі маршрутизація так і працює без додаткових налаштувань. Але у разі кількох зовнішніх каналів, будь-який вузол в Інтернеті доступний через кожен із зовнішніх каналів, а не тільки через єдиний, як в простій локальної мережі. І проблема полягає в тому, що якщо до маршрутизатору прийшов запит на IP-адресу резервного каналу, то без додаткових налаштувань відповідь піде через основний канал, оскільки туди направлений шлюз і в результаті буде відкинута провайдером, як некоректна. Для вирішення цього завдання використаємо два інструменти RouterOS: connection mark і routing mark. Connection mark дозволяє помітити потрібне з'єднання і надалі працювати з цією міткою, як умовою для застосування routing mark. А вже з routing mark можливо працювати з таблицею маршрутизації.

Мітка для нових вхідних з'єднань від кожного з провайдерів:

```
/ip firewall mangle add action=mark-connection chain=prerouting \
comment="Connmark in from ISP1" connection-mark=no-mark in-
interface=ether1 new-connection-mark=conn_isp1 passthrough=no

/ip firewall mangle add action=mark-connection chain=prerouting \
comment="Connmark in from ISP2" connection-mark=no-mark in-
interface=ether2 new-connection-mark=conn_isp2 passthrough=no
```

Направляємо вихідний транзитний трафік з міткою за відповідними таблицями маршрутизації:

```
/ip firewall mangle add action=mark-routing chain=prerouting \
comment="Routemark transit out via ISP1" connection-mark=conn_isp1 \
dst-address-type=!local in-interface-list=!WAN new-routing-mark=to_isp1
passthrough=no

/ip firewall mangle add action=mark-routing chain=prerouting \
comment="Routemark transit out via ISP2" connection-mark=conn_isp2 \
dst-address-type=!local in-interface-list=!WAN new-routing-mark=to_isp2
passthrough=no
```

Направляємо вихідний локальний трафік з міткою за відповідними таблицями маршрутизації:

```
/ip firewall mangle add action=mark-routing chain=output \
comment="Routemark local out via ISP1" connection-mark=conn_isp1 \
dst-address-type=!local new-routing-mark=to_isp1 passthrough=no

/ip firewall mangle add action=mark-routing chain=output \
comment="Routemark local out via ISP2" connection-mark=conn_isp2 \
dst-address-type=!local new-routing-mark=to_isp2 passthrough=no
```

На цьому етапі завдання підготовки до відправки відповіді в той канал Інтернет, з якого прийшов запит можна вважати вирішеною.

Для перемикання каналів за алгоритмом, який заданий за допомогою вартості маршрутів Distance, використовуємо механізм check gateway на основі методу рекурсивних маршрутів для більш глибокого аналізу стану каналу. Суть даного методу полягає в тому, що маршрутизатору вказується шлях до свого шлюзу не безпосередньо, а через проміжний шлюз. В якості такого шлюзу оптимальним варіантом будуть вузли в мережі Інтернет, які гарантовано працюють постійно. Наприклад, публічні DNS сервери CloudFlare (1.1.1.1) або Google (8.8.8.8). Налаштування маршрутів до проміжних шлюзів провайдерів може бути записано наступним чином:

```
/ip route add check-gateway=ping comment="For recursion via ISP1" \
distance=1 dst-address=1.1.1.1 gateway=x.x.x.x scope=10

/ip route add check-gateway=ping comment="For recursion via ISP2" \
distance=1 dst-address=8.8.8.8 gateway=y.y.y.y scope=10
```

Рекурсивні маршрути за замовчуванням для трафіку без routing mark (для резервного каналу параметр distance=2 що робить його неактивним до тих пір поки вузол 1.1.1.1 доступний через першого провайдера):

```
/ip route add check-gateway=ping comment="Unmarked via ISP1" \

distance=1 gateway=1.1.1.1

/ip route add check-gateway=ping comment="Unmarked via ISP2" \

distance=2 gateway=8.8.8.8
```

Рекурсивні маршрути за замовчуванням для трафіку з мітками:

```
/ip route add comment="Marked via ISP1" distance=1 gateway=1.1.1.1 \
routing-mark=to_isp1

/ip route add comment="Marked via ISP2" distance=1 gateway=8.8.8.8 \
routing-mark=to_isp2
```

Таким чином, після цих налаштувань основний трафік без міток буде спрямований через першого головного провайдера і, у випадку втрати з'єднання до проміжного шлюзу, автоматично буде перемикатись на резервного. При цьому, вхідний трафік, який має mark route, буде спрямований на шлюз

відповідного провайдера не залежно від того, який в даний момент активний шлюз для таблиці main. Запропонований підхід має можливість здійснювати переадресації портів DSNAT з обох провайдерів одночасно. Цей ефект важливий, наприклад, для поштового сервера з двома MX, які «дивляться» в різні канали Інтернет.

Список використаних джерел:

1. Настройка роутера MikroTik на два провайдера. – URL: https://www.technotrade.com.ua/Articles/mikrotik_2isp_setup_2015-06-19.php

2. Remote Desktop Protocol. – URL: https://uk.wikipedia.org/wiki/Remote_Desktop_Protocol

3. Manual:IP/Firewall/NAT. – URL: https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT

4. Князев И. Реализация MultiWAN. Вопросы, проблемы и решения. – URL: https://docplayer.ru/139710631-Realizaciya-multiwan-voprosy-problemy-i-resheniya.html