ACL の設定

Agenda

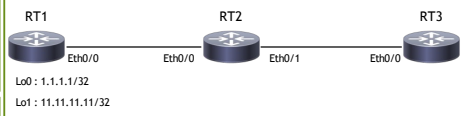
- Numbered standard ACLs
- Numbered extended ACLs
- Named ACLs
- Port ALCs (準備中)
- VLAN ACLs (準備中)

Numbered standard ACLs ～inbound～

```
Step1, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/6/6 ms

RT1#ping 192.168.23.3 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

RT1#ping 192.168.23.3 source 11.11.11.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 11.11.11.11
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/6 ms
```

```
Step2, ACLs を設定
RT2(config)#ip access-list standard 10
RT2(config-std-nacl)#10 deny 192.168.12.0 0.0.0.255
RT2(config-std-nacl)#20 permit host 1.1.1.1
```

```
Step3, 設定確認
RT2#show ip access-lists 10
Standard IP access list 10
    20 permit 1.1.1.1
    10 deny   192.168.12.0, wildcard bits 0.0.0.255
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

この例では、 Numbered Standard ACL を用いて RT2 の Eth0/0 に着信するトラフィックを制御する。

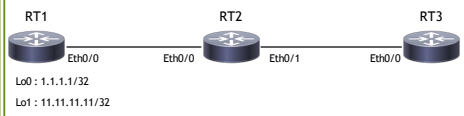
まず、 RT1 から RT3 (192.168.23.3) に対して 3つの IP から問題なく疎通できることを確認している。
その後、以下のルールで ACL を設定する。

シーケンス番号10 : Source IP 192.168.12.0/24 の通信を破棄
シーケンス番号20 : Source IP 1.1.1.1/32 の通信を許可

```
Step4, ACLs を適用
RT2(config)#interface ethernet 0/0
RT2(config-if)#ip access-group 10 in
```

```
Step5, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)

RT1#ping 192.168.23.3 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/7 ms

RT1#ping 192.168.23.3 source 11.11.11.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 11.11.11.11
UUUUU
Success rate is 0 percent (0/5)
```

```
Step6, 動作確認
RT2#show ip access-lists 10
Standard IP access list 10
    20 permit 1.1.1.1 (5 matches)
    10 deny   192.168.12.0, wildcard bits 0.0.0.255 (10 matches)
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

設定した ACL を RT2 の Eth0/0 に inbound 方向で適用する。

設定後に再度 RT1 から疎通確認を行うと、設定通り通信が制御されていることを確認できた。
※ 「11.11.11.11」 からの通信はどの ACE にも match しないため、 deny として処理されている。

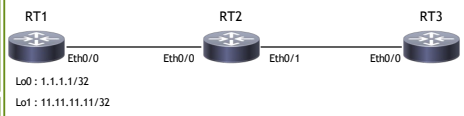
その後 ACLs を確認すると、対象 ACE に match した通信の数がカウントされていることがわかる。

Numbered standard ACLs ～outbound～

```
Step1, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/6/6 ms

RT1#ping 192.168.23.3 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

RT1#ping 192.168.23.3 source 11.11.11.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 11.11.11.11
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/6 ms
```

```
Step2, ACLs を設定
RT2(config)#ip access-list standard 10
RT2(config-std-nacl)#10 deny 192.168.12.0 0.0.0.255
RT2(config-std-nacl)#20 permit host 1.1.1.1
```

```
Step3, 設定確認
RT2#show ip access-lists 10
Standard IP access list 10
    20 permit 1.1.1.1
    10 deny   192.168.12.0, wildcard bits 0.0.0.255
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

この例では、 Numbered Standard ACL を用いて RT2 の Eth0/1 から送信されるトラフィックを制御する。

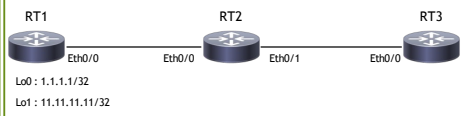
まず、 RT1 から RT3 (192.168.23.3) に対して 3つの IP から問題なく疎通できることを確認している。
その後、以下のルールで ACL を設定する。

シーケンス番号10 : Source IP 192.168.12.0/24 の通信を破棄
シーケンス番号20 : Source IP 1.1.1.1/32 の通信を許可

```
Step4, ACLs を適用
RT2(config)#interface ethernet 0/1
RT2(config-if)#ip access-group 10 out
```

```
Step5, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)

RT1#ping 192.168.23.3 source 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/7 ms

RT1#ping 192.168.23.3 source 11.11.11.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 11.11.11.11
UUUUU
Success rate is 0 percent (0/5)
------------------------------------------------------------------------------------------
RT2#ping 192.168.23.3 source 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
Packet sent with a source address of 192.168.12.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/5/6 ms
```

```
Step6, 動作確認
RT2#show ip access-lists 10
Standard IP access list 10
    20 permit 1.1.1.1 (5 matches)
    10 deny   192.168.12.0, wildcard bits 0.0.0.255 (10 matches)
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

設定した ACL を RT2 の Eth0/0 に outbound 方向で適用する。

設定後に再度 RT1 から疎通確認を行うと、設定通り通信が制御されていることを確認できた。
※「11.11.11.11」からの通信はどの ACE にも match しないため、deny として処理されている。

続いて RT2 (192.168.12.2) から RT2 (192.168.23.3) へ疎通確認を行うと、設定通りに通信が制御されていないことがわかる。
このように、outbound 方向に ACL を適用した場合、自身が生成する通信は制御の対象外となるため注意が必要。

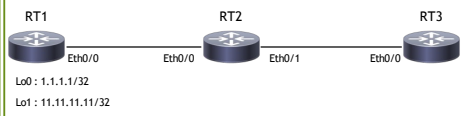
その後 ACLs を確認すると、対象 ACE に match した通信の数がカウントされていることがわかる。

Numbered extended ACLs

```
Step1, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/6/6 ms
------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3
Trying 192.168.23.3 ... Open


User Access Verification

Password:

RT3>
------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
test
HTTP/1.1 400 Bad Request
Date: Fri, 19 Aug 2022 09:25:09 GMT
Server: cisco-IOS
Accept-Ranges: none

400 Bad Request
[Connection to 192.168.23.3 closed by foreign host]
```

```
Step2, ACLs を設定
RT2(config)#ip access-list extended 100
RT2(config-ext-nacl)#10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq 23
RT2(config-ext-nacl)#20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255
```

```
Step3, 設定確認
RT2#show ip access-lists 100
Extended IP access list 100
    10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq telnet
    20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255
```



この例では、 Numbered extended ACL を用いて RT2 の Eth0/0 に着信するトラフィックを制御する。

まず、 RT1(192.168.12.1) から RT3 (192.168.23.3) に対して 3つの方法で通信ができることを確認している。
(1) ping
(2) telnet
(3) http

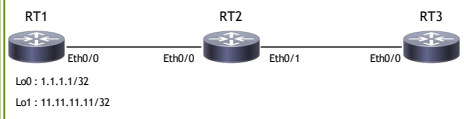
その後、以下のルールで ACL を設定する。

シーケンス番号10 : Source IP 192.168.12.0/24 から Destination IP 192.168.23.0/24 への 23_tcp 宛通信を廃棄
シーケンス番号20 : Source IP 192.168.12.0/24 から Destination IP 192.168.23.0/24 への通信を許可

```
Step4, ACLs を適用
RT2(config)#interface ethernet 0/0
RT2(config-if)#ip access-group 100 in
```

```
Step5, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/7 ms
---------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3
Trying 192.168.23.3 ...
% Destination unreachable; gateway or host down

---------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
test
HTTP/1.1 400 Bad Request
Date: Fri, 19 Aug 2022 09:28:41 GMT
Server: cisco-IOS
Accept-Ranges: none

400 Bad Request
[Connection to 192.168.23.3 closed by foreign host]
```

```
Step6, 動作確認
RT2#show ip access-lists 100
Extended IP access list 100
    10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq telnet (1 match)
    20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 (15 matches)
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

設定した ACL を RT2 の Eth0/0 に inbound 方向で適用する。
設定後に再度 RT1 から疎通確認を行うと、設定通り通信が制御されていることを確認できた。

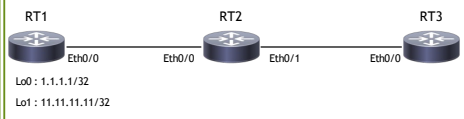
その後 ACLs を確認すると、対象 ACE に match した通信の数がカウントされていることがわかる。

Named ACLs

```
Step1, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/6/6 ms
------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3
Trying 192.168.23.3 ... Open


User Access Verification

Password:

RT3>
------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
test
HTTP/1.1 400 Bad Request
Date: Fri, 19 Aug 2022 09:25:09 GMT
Server: cisco-IOS
Accept-Ranges: none

400 Bad Request
[Connection to 192.168.23.3 closed by foreign host]
```

```
Step2, ACLs を設定
RT2(config)#ip access-list extended 100
RT2(config-ext-nacl)#10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq 23
RT2(config-ext-nacl)#20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255
```

```
Step3, 設定確認
RT2#show ip access-lists TEST
Extended IP access list TEST
    10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq telnet
    20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255
```



この例では、 Named を用いて RT2 の Eth0/0 に着信するトラフィックを制御する。
制御の内容としては先ほど紹介した Numbered extended ACL と同様とする。

まず、 RT1(192.168.12.1) から RT3 (192.168.23.3) に対して 3つの方法で通信ができることを確認している。
(1) ping
(2) telnet
(3) http

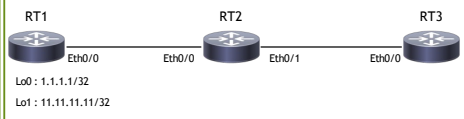
その後、以下のルールで ACL を設定する。

シーケンス番号10 : Source IP 192.168.12.0/24 から Destination IP 192.168.23.0/24 への 23_tcp 宛通信を廃棄
シーケンス番号20 : Source IP 192.168.12.0/24 から Destination IP 192.168.23.0/24 への通信を許可

```
Step4, ACLs を適用
RT2(config)#interface ethernet 0/0
RT2(config-if)#ip access-group TEST in
```

```
Step5, 疎通確認
RT1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/7 ms
---------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3
Trying 192.168.23.3 ...
% Destination unreachable; gateway or host down

---------------------------------------------------------------------------------------------
RT1#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
test
HTTP/1.1 400 Bad Request
Date: Fri, 19 Aug 2022 09:28:41 GMT
Server: cisco-IOS
Accept-Ranges: none

400 Bad Request
[Connection to 192.168.23.3 closed by foreign host]
```

```
Step6, 動作確認
RT2#show ip access-lists TEST
Extended IP access list TEST
    10 deny tcp 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 eq telnet (1 match)
    20 permit ip 192.168.12.0 0.0.0.255 192.168.23.0 0.0.0.255 (15 matches)
```



Lo0 : 1.1.1.1/32

Lo1 : 11.11.11.11/32

設定した ACL を RT2 の Eth0/0 に inbound 方向で適用する。
設定後に再度 RT1 から疎通確認を行うと、設定通り通信が制御されていることを確認できた。

その後 ACLs を確認すると、対象 ACE に match した通信の数がカウントされていることがわかる。