IPv6 트래픽을 전달하도록 ASA 구성

목차

소개

이 문서에서는 ASA 버전 7.0(1) 이상에서 IPv6(Internet Protocol Version 6) 트래픽을 전달하기 위해 Cisco ASA(Adaptive Security Appliance)를 구성하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 대한 특정 요건이 없습니다.

사용되는 구성 요소

이 문서의 정보는 Cisco ASA 버전 7.0(1) 이상을 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

현재 IPv6는 시장 점유 측면에서 아직 비교적 새로운 분야입니다.그러나 IPv6 컨피그레이션 지원 및 문제 해결 요청이 꾸준히 증가했습니다.이 문서의 목적은 이러한 요구 사항을 해결하고 다음을 제공하는 데 있습니다.

- IPv6 사용에 대한 일반적인 개요
- ASA의 기본 IPv6 컨피그레이션
- ASA를 통한 IPv6 연결 문제 해결 방법에 대한 정보
- Cisco TAC(Technical Assistance Center)에서 확인한 가장 일반적인 IPv6 문제 및 솔루션 목록

  참고:IPv6가 전 세계적으로 IPv4 교체로 아직 초기 단계에 있으므로 이 문서는 정확성과 관련성을 유지하기 위해 정기적으로 업데이트됩니다.

IPv6 기능 정보

다음은 IPv6 기능에 대한 몇 가지 중요한 정보입니다.

- IPv6 프로토콜은 ASA 버전 7.0(1)에 처음 도입되었습니다.
- 투명 모드에서 IPv6에 대한 지원이 ASA 버전 8.2(1)에 도입되었습니다.

IPv6 개요

IPv6 프로토콜은 1990년대 중반 ~ 후반에 개발되었으며, 주로 퍼블릭 IPv4 주소 공간이 빠르게 감소하기 때문입니다.NAT(Network Address Translation)가 IPv4에 큰 도움을 주고 이 문제를 지연시켰지만, 결국 대체 프로토콜이 필요하다는 것은 부인할 수 없습니다.IPv6 프로토콜은 1998년 12월 RFC 2460에 공식적으로 자세히 설명되어 있습니다.프로토콜에 대한 자세한 내용은 IETF(Internet Engineering Task Force) 웹 사이트에 있는 공식 RFC 2460 문서에서 확인할 수 있습니다.

IPv4를 통한 IPv6 개선

이 섹션에서는 IPv6 프로토콜과 이전 IPv4 프로토콜의 향상된 기능에 대해 설명합니다.

확장된 주소 지정 기능

IPv6 프로토콜은 더 많은 수준의 주소 지정 계층, 훨씬 더 많은 주소 지정 가능한 노드 수, 더 간단한 주소 자동 구성을 지원하기 위해 IP 주소 크기를 32비트에서 128비트로 늘립니다.멀티캐스트 주소에 *범위* 필드를 추가하여 멀티캐스트 라우팅의 확장성을 향상합니다. 또한 애니캐스트 *주소라고* 하는 새 주소 유형이 정의됩니다.이는 그룹 내 한 노드에 패킷을 전송하는 데 사용됩니다.

헤더 형식 간소화

패킷 처리 시 발생하는 일반적인 처리 비용을 줄이고 IPv6 헤더의 대역폭 비용을 제한하기 위해 일부 IPv4 헤더 필드가 삭제되거나 선택 사항으로 설정되었습니다.

확장 및 옵션 지원 향상

IP 헤더 옵션이 인코딩되는 방식을 변경하면 포워딩의 효율성이 높아지고, 옵션의 길이가 덜 제한되며, 향후 새로운 옵션이 도입될 수 있는 유연성이 향상됩니다.

플로우 레이블 지정 기능

발신자가 특별 처리를 요청하는 특정 트래픽 *흐름*에 속하는 패킷의 레이블 지정(예: 비기본 QoS(Quality of Service) 또는 *실시간* 서비스)을 활성화하기 위해 새로운 기능이 추가되었습니다.

인증 및 개인 정보 보호 기능

인증, 데이터 무결성 및 (선택 사항) 데이터 기밀성을 지원하는 데 사용되는 확장명은 IPv6에 대해 지정됩니다.

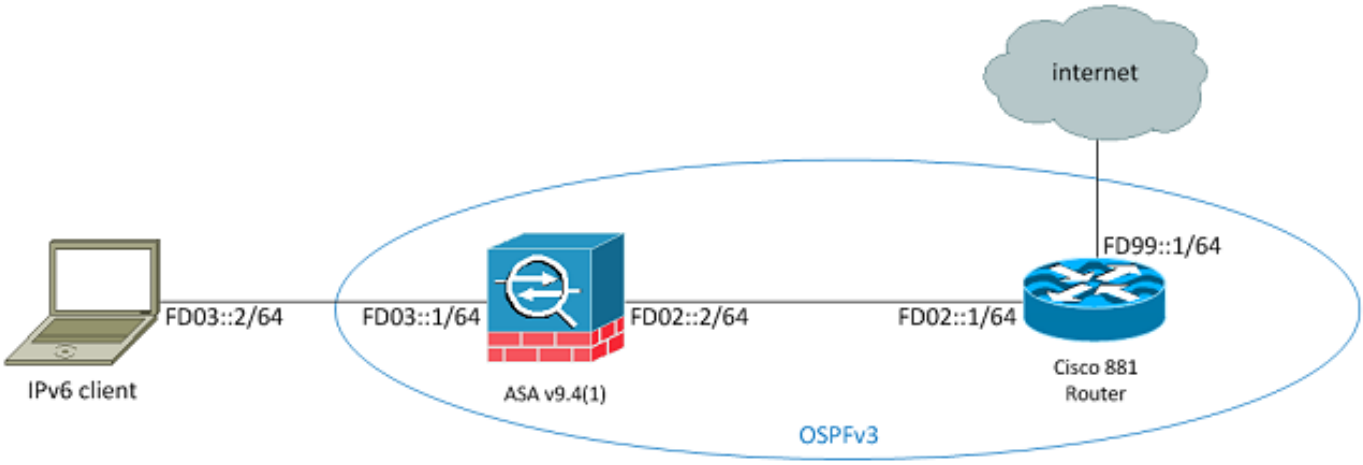
구성

이 섹션에서는 IPv6를 사용하도록 Cisco ASA를 구성하는 방법에 대해 설명합니다.

> **참고:**이 섹션에 사용된 명령에 대한 자세한 내용을 보려면 Command Lookup Tool(등록된 고객만 해당)을 사용합니다.

네트워크 다이어그램

다음은 이 문서에서 사용되는 예제의 IPv6 토폴로지입니다.



IPv6용 인터페이스 구성

ASA를 통해 IPv6 트래픽을 전달하려면 먼저 두 개 이상의 인터페이스에서 IPv6를 활성화해야 합니다.다음 예에서는 **Gi0/0**의 내부 인터페이스에서 **Gi0/1**의 외부 인터페이스로 트래픽을 전달하기 위해 IPv6을 활성화하는 방법을 설명합니다.

```
ASAv(config)# interface GigabitEthernet0/0
ASAv(config-if)# ipv6 enable

ASAv(config)# interface GigabitEthernet0/1
ASAv(config-if)# ipv6 enable
```

이제 두 인터페이스에서 IPv6 주소를 구성할 수 있습니다.

> **참고:**이 예에서는 fc00::/7의 ULA(Unique Local Addresses) 영역에 있는 주소가 사용되므로 모든 주소가 FD(예: fdxx:xxxx:xxxx:xxxx..)로 시작합니다..) 또한 IPv6 주소를 작성할 때 이중 콜론(::)을 사용하여 **FD01:0000:00000:00000:0000:0000:0000:00000:0000:00000:000000:0000000000000000000**0000000000000000000000000000000000000000000000 1.

```
ASAv(config)# interface GigabitEthernet0/0
ASAv(config-if)# ipv6 address fd03::1/64
ASAv(config-if)# nameif inside
ASAv(config-if)# security-level 100

ASAv(config)# interface GigabitEthernet0/1
ASAv(config-if)# ipv6 address fd02::2/64
ASAv(config-if)# nameif outside
ASAv(config-if)# security-level 0
```

이제 외부 VLAN의 업스트림 라우터에 대한 기본 레이어 2(L2)/레이어 3(L3) 연결이 fd02::1:로 설정되어 있어야 합니다.

```
ASAv(config-if)# ping fd02::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fd02::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

IPv6 라우팅 구성

IPv4와 마찬가지로, 직접 연결된 서브넷의 호스트와의 IPv6 연결이 있더라도 외부 네트워크에 연결하는 방법을 알아보려면 외부 네트워크에 대한 경로가 있어야 합니다.첫 번째 예는 다음 hop 주소가 fd02::1인 외부 인터페이스를 통해 모든 IPv6 네트워크에 연결하기 위해 고정 기본 경로를 구성하는 방법을 보여줍니다.

IPv6에 대한 고정 라우팅 구성

IPv6에 대한 고정 라우팅을 구성하려면 다음 정보를 사용합니다.

```
ASAv(config)# ipv6 route outside 0::0/0 fd02::1
ASAv(config)# show ipv6 route

IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, B - BGP
L fd02::2/128 [0/0]
via ::, outside
C fd02::/64 [0/0]
via ::, outside
L fd03::1/128 [0/0]
via ::, inside
C fd03::/64 [0/0]
via ::, inside
L fe80::/10 [0/0]
via ::, inside
via ::, outside
L ff00::/8 [0/0]
via ::, inside
via ::, outside
S   ::/0 [1/0]
via fd02::1, outsideASAv(config)#
```

표시된 대로 이제 외부 서브넷의 호스트에 대한 연결이 설정되었습니다.

```
ASAv(config)# ping fd99::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to fd99::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
ASAv(config)#
```

참고:IPv6에 대한 라우팅을 처리하기 위해 동적 라우팅 프로토콜이 필요한 경우 이를 구성할 수도 있습니다.이 내용은 다음 섹션에서 설명합니다.

OSPFv3을 사용하여 IPv6에 대한 동적 라우팅 구성

먼저 업스트림 Cisco 881 Series ISR(Integrated Services Router)에서 OSPFv3(Open Shortest Path First Version 3) 컨피그레이션을 검토해야 합니다.

```
C881#show run | sec ipv6
ipv6 unicast-routing

!--- This enables IPv6 routing in the Cisco IOS®.

.....
ipv6 ospf 1 area 0
address-family ipv6 unicast
passive-interface default
no passive-interface Vlan302

!--- This is the interface to send OSPF Hellos to the ASA.

default-information originate always

!--- Always distribute the default route.

redistribute static
ipv6 route ::/0 FD99::2

!--- Creates a static default route for IPv6 to the internet.
```

관련 인터페이스 컨피그레이션은 다음과 같습니다.

```
C881#show run int Vlan302
interface Vlan302
....
ipv6 address FD02::1/64
ipv6 ospf 1 area 0
C881#
```

ASA 패킷 캡처를 사용하여 외부 인터페이스의 ISR에서 OSPF *Hello* 패킷이 표시되는지 확인할 수 있습니다.

```
ASAv(config)# show run access-list test_ipv6
access-list test_ipv6 extended permit ip any6 any6
ASAv(config)# show cap
capture capout type raw-data access-list test_ipv6 interface outside
[Capturing - 37976 bytes]
ASAv(config)# show cap capout
```

```
367 packets captured

1: 11:12:04.949474 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
neighbor sol: who has fe80::250:56ff:fe9d:34a8 [class 0xe0]
2: 11:12:06.949444 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
neighbor sol: who has fe80::250:56ff:fe9d:34a8 [class 0xe0]
   3: 11:12:07.854768       fe80::c671:feff:fe93:b516 > ff02::5: ip-proto-89 40
[hlim 1]
4: 11:12:07.946545 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
neighbor sol: who has fe80::250:56ff:fe9d:34a8 [class 0xe0]
5: 11:12:08.949459 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
neighbor sol: who has fe80::250:56ff:fe9d:34a8 [class 0xe0]
6: 11:12:09.542772 fe80::217:fff:fe17:af80 > ff02::5: ip-proto-89 40
[hlim 1]
....
  13: 11:12:16.983011       fe80::c671:feff:fe93:b516 > ff02::5: ip-proto-89 40
[hlim 1]
14: 11:12:18.947170 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
neighbor sol: who has fe80::250:56ff:fe9d:34a8 [class 0xe0]
15: 11:12:19.394831 fe80::217:fff:fe17:af80 > ff02::5: ip-proto-89 40
[hlim 1]
16: 11:12:19.949444 fe80::250:56ff:fe9d:34a8 > ff02::1:ff9d:34a8: icmp6:
  21: 11:12:26.107477       fe80::c671:feff:fe93:b516 > ff02::5: ip-proto-89 40
[hlim 1]
ASAv(config)#
```

이전 패킷 캡처에서 ISR의 올바른 인터페이스에 해당하는 IPv6 링크-로컬 주소에서 OSPF(ip-proto-89) 패킷이 도착함을 확인할 수 있습니다.

```
C881#show ipv6 interface brief
......
Vlan302 [up/up]
    FE80::C671:FEFF:FE93:B516
FD02::1
C881#
```

이제 ISR과 인접성을 설정하기 위해 ASA에서 OSPFv3 프로세스를 생성할 수 있습니다.

```
ASAv(config)# ipv6 router ospf 1
ASAv(config-rtr)# passive-interface default
ASAv(config-rtr)# no passive-interface outside
ASAv(config-rtr)# log-adjacency-changes
ASAv(config-rtr)# redistribute connected
ASAv(config-rtr)# exit
```

ASA 외부 인터페이스에 OSPF 컨피그레이션을 적용합니다.

```
ASAv(config)# interface GigabitEthernet0/1
ASAv(config-if)# ipv6 ospf 1 area 0
ASAv(config-if)# end
```

이로 인해 ASA가 IPv6 서브넷에서 브로드캐스트 OSPF Hello 패킷을 전송할 수 있습니다.라우터와 인접성을 확인하려면 **show ipv6 ospf neighbor** 명령을 입력합니다.

```
ASAv# show ipv6 ospf neighbor

Neighbor ID Pri State Dead Time Interface ID Interface
    14.38.104.1 1     FULL/BDR        0:00:33     14              outside
```

또한 기본적으로 ID에 대해 구성된 가장 높은 IPv4 주소를 사용하므로 ISR에서 인접 디바이스 ID를 확인할 수도 있습니다.

```
C881#show ipv6 ospf 1
 Routing Process "ospfv3 1" with ID 14.38.104.1
Supports NSSA (compatible with RFC 3101)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an autonomous system boundary router
Redistributing External Routes from,
static
 Originate Default Route with always

!--- Notice the other OSPF settings that were configured.

Router is not originating router-LSAs with maximum metric
....

C881#
```

이제 ASA가 ISR에서 기본 IPv6 경로를 학습해야 합니다.이를 확인하려면 show ipv6 route 명령을 입력합니다.

```
ASAv# show ipv6 route

IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, B - BGP
O 2001:aaaa:aaaa:aaaa::/64 [110/10]
via ::, outside
L fd02::2/128 [0/0]
via ::, outside
C fd02::/64 [0/0]
via ::, outside
L fd03::1/128 [0/0]
via ::, inside
C fd03::/64 [0/0]
via ::, inside
L fe80::/10 [0/0]
via ::, inside
via ::, outside
L ff00::/8 [0/0]
via ::, inside
via ::, outside
OE2  ::/0 [110/1], tag 1

!--- Here is the learned default route.

    via fe80::c671:feff:fe93:b516, outside
ASAv#
```

이제 ASA에서 IPv6에 대한 인터페이스 설정 및 라우팅 기능의 기본 컨피그레이션이 완료되었습니다.

다음을 확인합니다.

현재 이 구성에 대해 사용 가능한 확인 절차가 없습니다.

문제 해결

IPv6 연결에 대한 문제 해결 절차는 IPv4 연결 문제를 해결하는 데 사용되는 대부분의 동일한 방법론을 따릅니다.트러블슈팅 관점에서 IPv4와 IPv6의 가장 중요한 차이점 중 하나는 ARP(Address Resolution Protocol)가 더 이상 IPv6에 없다는 것입니다. 로컬 LAN 세그먼트에서 IP 주소를 확인하기 위해 ARP를 사용하는 대신 IPv6에서는 ND(Neighbor Discovery)라는 프로토콜을 사용합니다.

또한 ND는 MAC(Media Access Control) 주소 확인을 위해 ICMPv6(Internet Control Message Protocol Version 6)을 활용한다는 점을 이해해야 합니다.IPv6 ND에 대한 자세한 내용은 CLI *Book 1*의 IPv6 Neighbor Discovery 섹션의 ASA IPv6 Configuration 가이드에서 확인할 수 있습니다 *.Cisco ASA Series General Operations CLI 컨피그레이션 가이드, 9.4* 또는 RFC 4861에 있습니다.

현재 대부분의 IPv6 관련 문제 해결에는 ND, 라우팅 또는 서브넷 구성 문제가 포함됩니다.이는 IPv4와 IPv6의 주요 차이점이기도 하기 때문입니다. NAT는 ARP와 다르게 작동하며 내부 네트워크 주소 지정도 매우 다릅니다. IPv6에서는 NAT를 사용하는 것이 매우 금지되어 있으며 개인 주소 지정은 IPv4에서 IPv4(RFC 1918 이후)에서 사용하던 방식을 더 이상 활용하지 않기 때문입니다. 이러한 차이점을 이해하고 L2/L3 문제가 해결되면 TCP/UDP 및 상위 레이어 프로토콜은 기본적으로 동일하기 때문에(사용되는 IP 버전에 관계없이) 레이어 4(L4) 이상의 문제 해결 프로세스는 IPv4에 사용되는 것과 기본적으로 동일합니다.

L2 연결 문제 해결(ND)

IPv6와의 L2 연결 문제를 해결하는 데 사용되는 가장 기본적인 명령은 **show ipv6 neighbor [nameif]** 명령이며, 이는 IPv4용 **show arp**와 같습니다.

다음은 출력의 예입니다.

```
ASAv(config)# show ipv6 neighbor outside
IPv6 Address Age Link-layer Addr State Interface
fd02::1                                   0 c471.fe93.b516  REACH   outside
fe80::c671:feff:fe93:b516                32 c471.fe93.b516  DELAY   outside
fe80::e25f:b9ff:fe3f:1bbf 101 e05f.b93f.1bbf STALE outside
fe80::b2aa:77ff:fe7c:8412 101 b0aa.777c.8412 STALE outside
fe80::213:c4ff:fe80:5f53 101 0013.c480.5f53 STALE outside
fe80::a64c:11ff:fe2a:60f4 101 a44c.112a.60f4 STALE outside
fe80::217:fff:fe17:af80 99 0017.0f17.af80 STALE outside
ASAv(config)#
```

이 출력에서는 MAC 주소가 **c471.fe93.b516인** 디바이스에 속하는 fd02**::1**의 IPv6 주소에 대한 성공적인 확인**을** 볼 수 있습니다.

> **참고:**동일한 라우터 인터페이스 MAC 주소가 이전 출력에 두 번 나타나는 것을 확인할 수 있습니다. 라우터에도 이 인터페이스에 대해 자체 할당된 링크-로컬 주소가 있기 때문입니다.링크-로컬 주소는 직접 연결된 네트워크의 통신에만 사용할 수 있는 장치별 주소입니다.라우터는 링크-로컬 주소를 통해 패킷을 전달하지 않고 직접 연결된 네트워크 세그먼트에서의 통신에만 사용됩니다.많은 IPv6 라우팅 프로토콜(예: OSPFv3)은 L2 세그먼트에서 라우팅 프로토콜 정보를 공유하기 위해 링크-로컬 주소를 사용합니다.

ND 캐시를 지우려면 clear ipv**6 neighbors** 명령**을** 입력합니다.특정 호스트에 대해 ND가 실패하면 debug ipv**6 nd** 명령을 입력하고 패킷 캡처를 수행하고 syslogs를 확인하여 L2 레벨에서 발생하는 것을 확인할 수 있습니다.IPv6 ND는 IPv6 주소에 대한 MAC 주소를 확인하기 위해 ICMPv6 메시지를 사용합니다.

IPv4 ARP 대 IPv6 ND

IPv4용 ARP 및 IPv6용 ND의 다음 비교 표를 고려하십시오.

IPv4 ARP	IPv6 ND
ARP 요청(10.10.10.1은?)	인접 디바이스 요청
ARP 회신(10.10.10.1은 dead.dead.dead)	네이버 광고

다음 시나리오에서는 ND가 외부 인터페이스에 있는 *fd02:1* 호스트의 MAC 주소를 확인하지 못합니다.

ND 디버그

다음은 debug ipv6 nd 명령의 출력입니다.

```
ICMPv6-ND: Sending NS for fd02::1 on outside

!--- "Who has fd02::1"

ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: INCMP deleted: fd02::1
ICMPv6-ND: INCMP -> DELETE: fd02::1
ICMPv6-ND: DELETE -> INCMP: fd02::1
ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: Sending NA for fd02::2 on outside

!--- "fd02::2 is at dead.dead.dead"

ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: INCMP deleted: fd02::1
ICMPv6-ND: INCMP -> DELETE: fd02::1
ICMPv6-ND: DELETE -> INCMP: fd02::1

!--- Here is where the ND times out.

ICMPv6-ND: Sending NS for fd02::1 on outside
ICMPv6-ND: Sending NS for fd02::1 on outside
```

이 디버그 출력에서는 fd02::2의 네이버 광고**가** 수신되지 않는 *것으로 나타납니다*.패킷 캡처를 확인하여 실제 사례인지 확인할 수 있습니다.

ND 패킷 캡처

> **참고:**ASA 릴리스 9.4(1)부터는 IPv6 패킷 캡처에 여전히 액세스 목록이 필요합니다.Cisco 버그 ID CSCtn09836을 사용하여 이를 추적하기 위해 개선 요청이 제출되었습니다.

ACL(Access Control List) 및 패킷 캡처를 구성합니다.

```
ASAv(config)# access-list test_ipv6 extended permit ip any6 any6
ASAv(config)# cap capout interface outside access-list test_ipv6
```

ASA에서 fd02::1에 대한 ping을 시작합니다.

```
ASAv(config)# show cap capout
....
23: 10:55:10.275284 fd02::2 > ff02::1:ff00:1: icmp6: neighbor sol: who has
fd02::1 [class 0xe0]
24: 10:55:10.277588 fd02::1 > fd02::2: icmp6: neighbor adv: tgt is fd02::1
[class 0xe0]
26: 10:55:11.287735 fd02::2 > ff02::1:ff00:1: icmp6: neighbor sol: who has
fd02::1 [class 0xe0]
27: 10:55:11.289642 fd02::1 > fd02::2: icmp6: neighbor adv: tgt is fd02::1
[class 0xe0]
28: 10:55:12.293365 fd02::2 > ff02::1:ff00:1: icmp6: neighbor sol: who has
fd02::1 [class 0xe0]
29: 10:55:12.298538 fd02::1 > fd02::2: icmp6: neighbor adv: tgt is fd02::1
[class 0xe0]
32: 10:55:14.283341 fd02::2 > ff02::1:ff00:1: icmp6: neighbor sol: who has
fd02::1 [class 0xe0]
33: 10:55:14.285690 fd02::1 > fd02::2: icmp6: neighbor adv: tgt is fd02::1
[class 0xe0]
35: 10:55:15.287872 fd02::2 > ff02::1:ff00:1: icmp6: neighbor sol: who has
fd02::1 [class 0xe0]
36: 10:55:15.289825 fd02::1 > fd02::2: icmp6: neighbor adv: tgt is fd02::1
[class 0xe0]
```

패킷 캡처에서 볼 수 있듯이 fd02::1의 네이버 광고가 수신됩니다.그러나 디버그 출력에 표시된 것처럼 일부 이유로 광고가 처리되지 않습니다. 추가 검사를 위해 syslogs를 볼 수 있습니다.

ND Syslogs

다음은 ND syslog의 예입니다.

```
May 13 2015 10:55:10: %ASA-7-609001: Built local-host identity:fd02::2
May 13 2015 10:55:10: %ASA-6-302020: Built outbound ICMP connection for faddr
ff02::1:ff00:1/0 gaddr fd02::2/0 laddr fd02::2/0(any)
May 13 2015 10:55:10: %ASA-3-325003: EUI-64 source address check failed.  Dropped
packet from outside:fd02::1/0 to fd02::2/0 with source MAC address c471.fe93.b516.
May 13 2015 10:55:10: %ASA-3-313008: Denied IPv6-ICMP type=136, code=0 from fd02::1
on interface outside
May 13 2015 10:55:11: %ASA-3-325003: EUI-64 source address check failed.  Dropped
packet from outside:fd02::1/0 to fd02::2/0 with source MAC address c471.fe93.b516.
May 13 2015 10:55:11: %ASA-3-313008: Denied IPv6-ICMP type=136, code=0 from fd02::1
on interface outside
May 13 2015 10:55:12: %ASA-3-325003: EUI-64 source address check failed.  Dropped
packet from outside:fd02::1/0 to fd02::2/0 with source MAC address c471.fe93.b516.
May 13 2015 10:55:12: %ASA-3-313008: Denied IPv6-ICMP type=136, code=0 from fd02::1
on interface outside
May 13 2015 10:55:14: %ASA-3-325003: EUI-64 source address check failed.  Dropped
packet from outside:fd02::1/0 to fd02::2/0 with source MAC address c471.fe93.b516.
May 13 2015 10:55:14: %ASA-3-313008: Denied IPv6-ICMP type=136, code=0 from fd02::1
on interface outside
May 13 2015 10:55:15: %ASA-3-325003: EUI-64 source address check failed.  Dropped
packet from outside:fd02::1/0 to fd02::2/0 with source MAC address c471.fe93.b516.
May 13 2015 10:55:15: %ASA-3-313008: Denied IPv6-ICMP type=136, code=0 from fd02::1
on interface outside
```

이러한 syslog 내에서 ISR(fd02::1)의 ND 네이버 광고 패킷이 EUI(Modified Extended Unique Identifier) 64(Modified EUI-64) 형식 검사 실패로 인해 삭제된 것을 확인할 수 있습니다.

팁:이 특정 문제에 대한 자세한 내용은 이 문서*의 Modified EUI-64 Address Encoding* 섹션을 참조하십시오.이 문제 해결 로직은 특정 인터페이스에서 ICMPv6를 허용하지 않거나 uRPF(Unicast Reverse Path Forwarding) 검사 실패가 발생하는 경우 등 모든 종류의 삭제 이유에도 적용할 수 있습니다. 이 경우 둘 다 IPv6에서 L2 연결 문제를 일으킬 수 있습니다.

기본 IPv6 라우팅 문제 해결

IPv6을 사용할 때 라우팅 프로토콜에 대한 트러블슈팅 절차는 기본적으로 IPv4를 사용할 때와 동일합니다.**debug** 및 **show 명령**과 패킷 캡처를 사용하면 라우팅 프로토콜이 예상대로 작동하지 않는 이유를 확인하는 데 유용합니다.

IPv6용 라우팅 프로토콜 디버그

이 섹션에서는 IPv6에 유용한 디버그 명령을 제공합니다.

전역 IPv6 라우팅 디버그

debug ipv6 라우팅 디버그를 사용하여 모든 IPv6 라우팅 테이블 변경 사항을 해결할 수 있습니다.

```
ASAv# clear ipv6 ospf 1 proc

Reset OSPF process? [no]: yes
ASAv# IPv6RT0: ospfv3 1, Route update to STANDBY with epoch: 2 for
2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ospfv3 1, Delete 2001:aaaa:aaaa:aaaa::/64 from table
IPv6RT0: ospfv3 1, Delete backup for fd02::/64
IPv6RT0: ospfv3 1, Route update to STANDBY with epoch: 2 for ::/0
IPv6RT0: ospfv3 1, Delete ::/0 from table
IPv6RT0: ospfv3 1, ipv6_route_add_core for 2001:aaaa:aaaa:aaaa::/64 [110/10],
next-hop :: nh_source :: via interface outside route-type 2
IPv6RT0: ospfv3 1, Add 2001:aaaa:aaaa:aaaa::/64 to table
IPv6RT0: ospfv3 1, Added next-hop :: over outside for 2001:aaaa:aaaa:aaaa::/64,
[110/10]
IPv6RT0: ospfv3 1, ipv6_route_add_core Route update to STANDBY with epoch: 2 for
2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ipv6_route_add_core: input add 2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ipv6_route_add_core: output add 2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ospfv3 1, ipv6_route_add_core for fd02::/64 [110/10], next-hop ::
nh_source :: via interface outside route-type 2
IPv6RT0: ospfv3 1, ipv6_route_add_core for ::/0 [110/1], next-hop
fe80::c671:feff:fe93:b516
nh_source fe80::c671:feff:fe93:b516 via interface outside route-type 16
IPv6RT0: ospfv3 1, Add ::/0 to table
IPv6RT0: ospfv3 1, Added next-hop fe80::c671:feff:fe93:b516 over outside for ::/0,
[110/1]
IPv6RT0: ospfv3 1, ipv6_route_add_core Route update to STANDBY with epoch: 2 for ::/0
IPv6RT0: ipv6_route_add_core: input add ::/0
IPv6RT0: ipv6_route_add_core: output add ::/0
IPv6RT0: ospfv3 1, ipv6_route_add_core for 2001:aaaa:aaaa:aaaa::/64 [110/10],
next-hop :: nh_source :: via interface outside route-type 2
IPv6RT0: ospfv3 1, Route add 2001:aaaa:aaaa:aaaa::/64 [owner]
IPv6RT0: ospfv3 1, ipv6_route_add_core Route update to STANDBY with epoch: 2 for
2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ipv6_route_add_core: input add 2001:aaaa:aaaa:aaaa::/64
IPv6RT0: ipv6_route_add_core: output add 2001:aaaa:aaaa:aaaa::/64
```

```
IPv6RT0: ospfv3 1, ipv6_route_add_core for fd02::/64 [110/10], next-hop ::
nh_source :: via interface outside route-type 2
IPv6RT0: ospfv3 1, Reuse backup for fd02::/64, distance 110
IPv6RT0: ospfv3 1, ipv6_route_add_core for ::/0 [110/1], next-hop
fe80::c671:feff:fe93:b516 nh_source fe80::c671:feff:fe93:b516 via interface outside
route-type 16
IPv6RT0: ospfv3 1, Route add ::/0 [owner]
IPv6RT0: ospfv3 1, ipv6_route_add_core Route update to STANDBY with epoch: 2 for ::/0
IPv6RT0: ipv6_route_add_core: input add ::/0
IPv6RT0: ipv6_route_add_core: output add ::/0
```

OSPFv3 디버그

OSPFv3 문제를 해결하기 위해 debug ipv6 ospf 명령을 사용할 수 있습니다.

```
ASAv# debug ipv6 ospf ?

adj OSPF adjacency events
database-timer OSPF database timer
events OSPF events
flood OSPF flooding
graceful-restart OSPF Graceful Restart processing
hello OSPF hello events
ipsec OSPF ipsec events
lsa-generation OSPF lsa generation
lsdb OSPF database modifications
packet OSPF packets
retransmission OSPF retransmission events
spf OSPF spf
```

다음은 OSPFv3 프로세스가 다시 시작된 후 활성화된 모든 디버그의 출력 예입니다.

```
ASAv# clear ipv6 ospf 1
OSPFv3: rcv. v:3 t:1 l:44 rid:192.168.128.115
aid:0.0.0.0 chk:a9ac inst:0 from outside
OSPFv3: Rcv hello from 192.168.128.115 area 0 from outside fe80::217:fff:fe17:af80
interface ID 142
OSPFv3: End of hello processingpr
OSPFv3: rcv. v:3 t:1 l:44 rid:14.38.104.1
aid:0.0.0.0 chk:bbf3 inst:0 from outside
OSPFv3: Rcv hello from 14.38.104.1 area 0 from outside fe80::c671:feff:fe93:b516
interface ID 14
OSPFv3: End of hello processingo
ASAv# clear ipv6 ospf 1 process

Reset OSPF process? [no]: yes
ASAv#
OSPFv3: Flushing External Links
Insert LSA 0 adv_rtr 172.16.118.1, type 0x4005 in maxage
OSPFv3: Add Type 0x4005 LSA ID 0.0.0.0 Adv rtr 172.16.118.1 Seq 80000029 to outside
14.38.104.1 retransmission list
....

!--- The neighbor goes down:

OSPFv3: Neighbor change Event on interface outside
OSPFv3: DR/BDR election on outside
OSPFv3: Elect BDR 14.38.104.1
OSPFv3: Elect DR 192.168.128.115
OSPFv3: Schedule Router LSA area: 0, flag: Change
OSPFv3: Schedule Router LSA area: 0, flag: Change
OSPFv3: Schedule Prefix DR LSA intf outside
```

```
OSPFv3: Schedule Prefix Stub LSA area 0
OSPFv3: 14.38.104.1 address fe80::c671:feff:fe93:b516 on outside is dead, state DOWN
....

!--- The neighbor resumes the exchange:

OSPFv3: Rcv DBD from 14.38.104.1 on outside seq 0xd09 opt 0x0013 flag 0x7 len 28
 mtu 1500 state EXSTART
OSPFv3: First DBD and we are not SLAVE
OSPFv3: rcv. v:3 t:2 l:168 rid:14.38.104.1
     aid:0.0.0.0 chk:5aa3 inst:0 from outside
OSPFv3: Rcv DBD from 14.38.104.1 on outside seq 0x914 opt 0x0013 flag 0x2 len 168
 mtu 1500 state EXSTART
OSPFv3: NBR Negotiation Done. We are the MASTER
OSPFv3: outside Nbr 14.38.104.1: Summary list built, size 0
OSPFv3: Send DBD to 14.38.104.1 on outside seq 0x915 opt 0x0013 flag 0x1 len 28
OSPFv3: rcv. v:3 t:2 l:28 rid:192.168.128.115
     aid:0.0.0.0 chk:295c inst:0 from outside
OSPFv3: Rcv DBD from 192.168.128.115 on outside seq 0xfeb opt 0x0013 flag 0x7 len 28
 mtu 1500 state EXSTART
OSPFv3: NBR Negotiation Done. We are the SLAVE
OSPFv3: outside Nbr 192.168.128.115: Summary list built, size 0
OSPFv3: Send DBD to 192.168.128.115 on outside seq 0xfeb opt 0x0013 flag 0x0 len 28
OSPFv3: rcv. v:3 t:2 l:28 rid:14.38.104.1
     aid:0.0.0.0 chk:8d74 inst:0 from outside
OSPFv3: Rcv DBD from 14.38.104.1 on outside seq 0x915 opt 0x0013 flag 0x0 len 28
 mtu 1500 state EXCHANGE
....

!--- The routing is re-added to the OSPFv3 neighbor list:

OSPFv3: Add Router 14.38.104.1 via fe80::c671:feff:fe93:b516, metric: 10
 Router LSA 14.38.104.1/0, 1 links
   Link 0, int 14, nbr 192.168.128.115, nbr int 142, type 2, cost 1
     Ignore newdist 11 olddist 10
```

EIGRP(Enhanced Interior Gateway Routing Protocol)

ASA의 EIGRP는 IPv6 사용을 지원하지 않습니다. *CLI* Book *1*의 EIGRP에 대한 지침 섹션을 참조하십시오.*자세한 내용은 Cisco ASA Series General Operations CLI 컨피그레이션 가이드, 9.4*를 참조하십시오.

BGP(Border Gateway Protocol)

IPv6을 사용할 때 BGP를 트러블슈팅하기 위해 이 debug 명령을 사용할 수 있습니다.

```
ASAv# debug ip bgp ipv6 unicast  ?

X:X:X:X::X IPv6 BGP neighbor address
keepalives BGP keepalives
updates BGP updates
<cr>
```

IPv6에 대한 유용한 Show 명령

IPv6 문제를 해결하기 위해 다음 **show** 명령을 사용할 수 있습니다.

- **show ipv6 route**

- **show ipv6 interface brief**
- **show ipv6 ospf <프로세스 ID>**
- **show ipv6 트래픽**
- **show ipv6 인접 디바이스**
- **show ipv6 icmp**

IPv6를 사용하는 패킷 추적기

IPv4와 동일한 방식으로 ASA에서 IPv6에 내장된 패킷 추적기 기능을 사용할 수 있습니다. 다음은 패킷 추적기 기능을 사용하여 fd03**::2**의 내부 호스트를 시뮬레이션하는 데 사용되는 예입니다. 이는 **881 인터페이스를 통해 학습된 기본 경로**를 사용하여 인터넷에 있는 555555::1:1의 웹 서버 연결을 시도하는 예입니다. ospf:

```
ASAv# packet-tracer input inside tcp fd03::2 10000 5555::1 80 detailed

Phase: 1
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
 Forward Flow based lookup yields rule:
 in  id=0x7fffd59ca0f0, priority=1, domain=permit, deny=false
       hits=2734, user_data=0x0, cs_id=0x0, l3_type=0xdd86
       src mac=0000.0000.0000, mask=0000.0000.0000
       dst mac=0000.0000.0000, mask=0100.0000.0000
       input_ifc=inside, output_ifc=any

Phase: 2
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop fe80::c671:feff:fe93:b516 using egress ifc  outside

Phase: 3
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:
 Forward Flow based lookup yields rule:
 in  id=0x7fffd589cc30, priority=1, domain=nat-per-session, deny=true
       hits=1166, user_data=0x0, cs_id=0x0, reverse, use_real_addr, flags=0x0,
protocol=6
       src ip/id=::/0, port=0, tag=any
       dst ip/id=::/0, port=0, tag=any
       input_ifc=any, output_ifc=any

<<truncated output>>
```

```
Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: outside
output-status: up
output-line-status: up
Action: allow

ASAv#
```

이그레스 MAC 주소는 881 인터페이스의 링크-로컬 주소입니다.앞에서 설명한 것처럼, 많은 동적 라우팅 프로토콜에서 라우터는 링크-로컬 IPv6 주소를 사용하여 인접성을 설정합니다.

IPv6 관련 ASA 디버깅 전체 목록

다음은 IPv6 문제를 해결하는 데 사용할 수 있는 디버그입니다.

```
ASAv# debug ipv6 ?

dhcp IPv6 generic dhcp protocol debugging
dhcprelay IPv6 dhcp relay debugging
icmp ICMPv6 debugging
interface IPv6 interface debugging
mld IPv6 Multicast Listener Discovery debugging
nd IPv6 Neighbor Discovery debugging
ospf OSPF information
packet IPv6 packet debugging
routing IPv6 routing table debugging
```

일반적인 IPv6 관련 문제

이 섹션에서는 가장 일반적인 IPv6 관련 문제를 해결하는 방법을 설명합니다.

잘못 구성된 서브넷

IPv6의 기능에 대한 일반적인 지식이 부족하거나 관리자가 IPv4별 프로세스를 사용하여 IPv6를 구현하려는 시도로 인해 많은 IPv6 TAC 케이스가 생성됩니다.

예를 들어, TAC에서 관리자가 ISP(인터넷 서비스 공급자)에 의해 IPv6 주소 블록 \56을 할당한 경우를 확인했습니다. 그런 다음 관리자는 ASA 외부 인터페이스에 주소 및 전체 \56 서브넷을 할당하고 내부 서버에 사용할 일부 내부 범위를 선택합니다.그러나 IPv6에서는 모든 내부 호스트에서도 라우팅 가능한 IPv6 주소를 사용해야 하며 필요에 따라 IPv6 주소 블록을 더 작은 서브넷으로 분할해야 합니다.이 시나리오에서는 할당된 \56 블록의 일부로 많은 \64 개의 서브넷을 만들 수 있습니다.

> **팁**:자세한 내용은 RFC 4291을 참조하십시오.

수정된 EUI 64 인코딩

수정된 EUI-64 인코딩 IPv6 주소를 요구하도록 ASA를 구성할 수 있습니다.EUI는 RFC 4291에 따라 호스트가 고유한 64비트 IPv6 인터페이스 식별자(EUI-64)를 자신에게 할당할 수 있도록 합니다.

이 기능은 IPv4보다 유리하며, IPv6 주소 할당에 DHCP를 사용해야 하는 요구 사항을 제거합니다.

ASA가 ipv6 enforce-eui64 nameif 명령을 통해 이 개선 사항을 요구하도록 구성된 경우 로컬 서브넷의 다른 호스트에서 많은 네이버 검색 요청 및 광고를 삭제할 가능성이 높습니다.

팁:자세한 내용은 IPv6 EUI-64비트 주소 이해 Cisco 지원 커뮤니티 문서를 참조하십시오.

클라이언트는 기본적으로 임시 IPv6 주소를 사용합니다.

기본적으로 Microsoft Windows 버전 7 및 8, Macintosh OS-X 및 Linux 기반 시스템과 같은 많은 클라이언트 OS(Operating Systems)는 IPv6 SLAAC(Stateless Address Autoconfiguration)를 통해 개인 정보를 확대하기 위해 자체 할당의 *임시* IPv6 주소를 사용합니다.

Cisco TAC에서는 호스트에서 정적으로 할당된 주소가 아니라 임시 주소에서 트래픽을 생성하므로 이 문제가 환경에서 예기치 않은 문제를 일으킨 경우가 있습니다.따라서 ACL 및 호스트 기반 경로는 트래픽이 삭제되거나 잘못 라우팅되어 호스트 통신이 실패할 수 있습니다.

이러한 상황을 해결하기 위해 사용되는 두 가지 방법이 있습니다.이 동작은 클라이언트 시스템에서 개별적으로 비활성화하거나 ASA 및 Cisco IOS® 라우터에서 이 동작을 비활성화할 수 있습니다.ASA 또는 라우터 측에서 이 동작을 트리거하는 RA(Router Advertisement) 메시지 플래그를 수정해야 합니다.

개별 클라이언트 시스템에서 이 동작을 비활성화하려면 다음 섹션을 참조하십시오.

Microsoft Windows

Microsoft Windows 시스템에서 이 동작을 비활성화하려면 다음 단계를 완료하십시오.

1. Microsoft Windows에서 관리자 권한으로 명령 프롬프트를 엽니다.
2. 임의의 IP 주소 생성 기능을 비활성화하려면 이 명령을 입력한 다음 Enter를 **누릅니다**.

```
netsh interface ipv6 set global randomizeidentifiers=disabled
```

3. Microsoft Windows에서 EUI-64 표준을 사용하도록 하려면 다음 명령을 입력합니다.

```
netsh interface ipv6 set privacy state=disabled
```

4. 변경 사항을 적용하려면 시스템을 재부팅합니다.

Macintosh OS-X

터미널에서 다음 재부팅될 때까지 호스트에서 IPv6 SLAAC를 비활성화하려면 다음 명령을 입력합니다.

```
sudo sysctl -w net.inet6.ip6.use_tempaddr=0
```

컨피그레이션을 영구적으로 유지하려면 다음 명령을 입력합니다.

```
sudo sh -c 'echo net.inet6.ip6.use_tempaddr=0 >> /etc/sysctl.conf'
```

리눅스

터미널 셸에서 다음 명령을 입력합니다.

```
sysctl -w net.ipv6.conf.all.use_tempaddr=0
```

ASA에서 전역적으로 SLAAC 비활성화

이 동작을 해결하기 위해 사용되는 두 번째 방법은 ASA에서 클라이언트로 전송되는 RA 메시지를 수정하는 것입니다. 그러면 SLAAC 사용이 트리거됩니다.RA 메시지를 수정하려면 *인터페이스 컨피그레이션* 모드에서 다음 명령을 입력합니다.

```
ASAv(config)# interface gigabitEthernet 1/1
ASAv(config-if)# ipv6 nd prefix 2001::db8/32 300 300 no-autoconfig
```

이 명령은 A-bit 플래그가 설정되지 않고 클라이언트가 임시 IPv6 주소를 생성하지 않도록 ASA에서 보내는 RA 메시지를 수정합니다.

팁:자세한 내용은 RFC 4941을 참조하십시오.

IPv6 FAQ

이 섹션에서는 IPv6 사용과 관련하여 자주 묻는 몇 가지 질문에 대해 설명합니다.

동일한 인터페이스에서 IPv4 및 IPv6에 대한 트래픽을 동시에 전달할 수 있습니까?

예.인터페이스에서 IPv6를 활성화하고 인터페이스에 IPv4 및 IPv6 주소를 모두 할당해야 하며, 두 트래픽 유형을 동시에 처리해야 합니다.

동일한 인터페이스에 IPv6 및 IPv4 ACL을 모두 적용할 수 있습니까?

버전 9.0(1) 이전 버전의 ASA에서 이 작업을 수행할 수 있습니다. ASA 버전 9.0(1)부터 ASA의 모든 ACL이 *통합됩니다*. 즉, ACL은 동일한 ACL에서 IPv4 및 IPv6 엔트리의 조합을 모두 지원합니다.

ASA 버전 9.0(1) 이상에서는 ACL이 단순히 통합되고 **access-group** 명령을 통해 단일 통합 ACL이 인터페이스에 적용됩니다.

ASA에서 IPv6에 대한 QoS를 지원합니까?

예.ASA는 IPv4와 동일한 방식으로 IPv6에 대한 폴리싱 및 우선 순위 큐잉을 지원합니다.

ASA 버전 9.0(1)부터 ASA의 모든 ACL이 *통합됩니다*. 즉, ACL은 동일한 ACL에서 IPv4 및 IPv6 엔트리의 조합을 모두 지원합니다.따라서 ACL과 일치하는 클래스 맵에서 작동하는 모든 QoS 명령은 IPv4 및 IPv6 트래픽에서 모두 작업을 수행합니다.

IPv6에 NAT를 사용해야 합니까?

ASA에서 IPv6에 대해 NAT를 구성할 수 있지만, 전 세계적으로 라우팅 가능한 IPv6 주소가 거의 무한대에 달하는 상황에서 IPv6에서 NAT를 사용하는 것은 매우 바람직하지 않으며 불필요합니다.

IPv6 시나리오에서 NAT가 필요한 경우 CLI *Book 2*의 IPv6 NAT Guidelines 섹션에서 NAT를 구성하는 방법에 대한 자세한 내용을 확인할 수 있습니다.*Cisco ASA Series Firewall CLI 컨피그레이션 가이드, 9.4*.

> **참고:**IPv6로 NAT를 구현할 때 고려해야 할 몇 가지 지침 및 제한 사항이 있습니다.

show failover 명령 출력에 link-local IPv6 주소가 표시되는 이유는 무엇입니까?

IPv6에서 ND는 L2 주소 확인을 수행하기 위해 링크-로컬 주소를 사용합니다.따라서 show failover 명령 출력에서 모니터링되는 인터페이스의 IPv6 주소는 인터페이스에 구성된 전역 IPv6 주소가 아니라 링크-로컬 주소를 표시합니다.이는 예상 동작입니다.

알려진 주의 사항/개선 요청

다음은 IPv6 사용과 관련하여 알려진 몇 가지 주의 사항입니다.

- Cisco 버그 ID CSCtn09836 ASA *8.x 캡처 "match" 절은 IPv6 트래픽을 포착하지 않습니다.*
- Cisco 버그 ID CSCuq85949 *ENH:WCCP를 위한 ASA IPv6 지원*
- Cisco 버그 ID CSCut78380 ASA IPv6 ECMP *라우팅은 트래픽을 로드 밸런싱하지 않습니다.*

관련 정보

- RFC 2460 IPv6(Internet Protocol, 버전 6) 사양
- RFC 4291 IP 버전 6 주소 지정 아키텍처
- RFC 4861 IP 버전 6(IPv6)용 인접 디바이스 검색
- CLI Book 1:Cisco ASA Series General Operations CLI 컨피그레이션 가이드, 9.4 IPv6
- AnyConnect SSL over IPv4+IPv6 to ASA 컨피그레이션
- 기술 지원 및 설명서 Cisco Systems