Introduction to IPv6
DNS, DHCP, FHS, FHRP

Stefan Kollar
Consulting Systems Engineer
CCIE #10668

skollar@cisco.com
DNS for IPv6
DNS Basics

- DNS is a database managing Resource Records (RR)
  Stockage of RR from various types—IPV4 and IPV6:
    Start of Authority (SoA)
    Name Server
    Address—A and AAAA
    Pointer—PTR

- DNS is an IP application
  It uses either UDP or TCP on top of IPv4 or IPv6

- References
  RFC3596: DNS Extensions to Support IP Version 6
  RFC3363: Representing Internet Protocol Version 6 Addresses in Domain Name system (DNS)
  RFC3364: Tradeoffs in Domain Name System (DNS) Support for Internet Protocol version 6 (IPv6)
IPv6 and DNS

**Hostname to IP address**

**IPv4**

A record:

www.abc.test. A 192.168.30.1

**IPv6**

AAAA record:

www.abc.test AAAA 2001:db8:C18:1::2

**IP address to hostname**

**IPv4**

PTR record:


**IPv6**

PTR record:

2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.8.1.c.0.
8.b.d.0.1.0.0.2.ip6.arpa PTR www.abc.test.
In a dual stack case an application that:

- Is IPv4 and IPv6-enabled
- Can query the DNS for IPv4 and/or IPv6 records (A) or (AAAA) records
- Chooses one address and, for example, connects to the IPv6 address

www.example.org = * ?

192.168.0.3
2001:db8:1::1

www IN A 192.168.0.3
www IN AAAA 2001:db8:1::1
**DNS Query on Windows 7 (Dual Stack)**

### Domain name with IPv6 address only

<table>
<thead>
<tr>
<th>mSecs</th>
<th>Source</th>
<th>Destination</th>
<th>Prot</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>64.104.197.141</td>
<td>64.104.200.248</td>
<td>DNS</td>
<td>Standard query A ipv6.google.com</td>
</tr>
<tr>
<td>0.158</td>
<td>64.104.200.248</td>
<td>64.104.197.141</td>
<td>DNS</td>
<td>Standard query response CNAME ipv6.1.google.com</td>
</tr>
<tr>
<td>0.000</td>
<td>64.104.197.141</td>
<td>64.104.200.248</td>
<td>DNS</td>
<td>Standard query AAAA ipv6.google.com</td>
</tr>
<tr>
<td>0.135</td>
<td>64.104.200.248</td>
<td>64.104.197.141</td>
<td>DNS</td>
<td>Standard query response CNAME ipv6.1.google.com AAAA IPv6 address of canonical name returned</td>
</tr>
</tbody>
</table>

Initial Query over IPv4 for IPv4 A record
DNS response refers to an alias/canonical address
Host immediately sends a request for AAAA record (original FQDN)
IPv6 address of canonical name returned

### Domain name with both addresses

<table>
<thead>
<tr>
<th>mSecs</th>
<th>Source</th>
<th>Destination</th>
<th>Prot</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>64.104.197.141</td>
<td>64.104.200.248</td>
<td>DNS</td>
<td>Standard query A <a href="http://www.apnic.net">www.apnic.net</a></td>
</tr>
<tr>
<td>0.017</td>
<td>64.104.200.248</td>
<td>64.104.197.141</td>
<td>DNS</td>
<td>Standard query response A 202.12.29.211 IPv4 address returned</td>
</tr>
<tr>
<td>0.000</td>
<td>64.104.197.141</td>
<td>64.104.200.248</td>
<td>DNS</td>
<td>Standard query AAAA <a href="http://www.apnic.net">www.apnic.net</a></td>
</tr>
<tr>
<td>0.017</td>
<td>64.104.200.248</td>
<td>64.104.197.141</td>
<td>DNS</td>
<td>Standard query response AAAA 2001:dc0:2001:11::211 IPv6 address of FQDN returned</td>
</tr>
<tr>
<td>0.001</td>
<td>2001:420:1::ff:2</td>
<td>2001:dc0:2001:11::211</td>
<td>ICMPv6</td>
<td>Echo request (Unknown (0x00)) Hosts prefers IPv6 address (configurable)</td>
</tr>
<tr>
<td>0.023</td>
<td>2001:dc0:2001:11::211</td>
<td>2001:420:1::ff:2</td>
<td>ICMPv6</td>
<td>Echo reply (Unknown (0x00))</td>
</tr>
</tbody>
</table>
How do we get the configuration information and prefixes from the ISP provisioning system, to the PE, from the PE to the user CPE, and from the CPE to the end user hosts? Routes for delegated prefixes/addresses also need to be injected into the ISP’s routing system.

Prefix Delegation
Assignment of variable length prefixes
Independent of end user topology
Media independent
Additional Informations (DNS, NTP, SMTP, POP, etc)
DHCPv6 PD: RFC 3633

- **Media independence**
  - e.g., ADSL, FTTH, Cable
  - Only knows identity of requesting router

- **Leases for prefixes**

- **Flexible deployments**
  - Client/relay/server model

- **Requesting router** includes request for prefixes in DHCP configuration request

- **Delegating router** assigns prefixes in response along with other DHCP configuration information
Large Scale Deployment
Suggested Solution

ISP
ISP provisioning system

PE

(1) PE sends RADIUS request for the user

(2) RADIUS responds with user’s prefix

(3) CPE configures addresses from the prefix

(4) PE sends DHCP REPLY containing request options. Note that the CPE is configured as a DHCP client upstream, and as a DHCP server downstream. The DHCP downstream server acts as a cache, and uses the options received on the upstream interface.

(5) CPE sends DHCP REPLY

(6) Host configures addresses based on the prefixes received in the RA. As the O-bit is set to on, it sends a DHCP INFORMATION-REQUEST message, with an ORO = DNS

(7) CPE sends a DHCP REPLY containing request options. Note that the CPE is configured as a DHCP client upstream, and as a DHCP server downstream. The DHCP downstream server acts as a cache, and uses the options received on the upstream interface.

The PE can auto-configure an "uplink" address. Prefix should be different from the prefix assigned to the user.
DHCPv6 Relay Agent Notification for Prefix Delegation (PD)

1. Request for PD
2. Relay-Forward for PD
3. Relay-Reply for PD
4. Relay: Install DHCPv6 relay binding associated with PD
5. Relay: Install IPv6 static route for the PD
6. Reply for PD
DHCPv6 Relay-Chaining & Route Insertion for Prefix Delegation

Note: DHCPv6 relay bindings and corresponding IPv6 static routes are installed only on the first hop or directly connected relay agent.
Relayed Messages

1. Client Message

+- Start of SOLICIT (1) message (34 bytes)
  | transaction-id 12
  | ia-na (3) option (12 bytes)
  | (iaid 1, t1 0, t2 0)
  | client-identifier (1) option (10 bytes)
  | 00:03:00:01:01:02:03:04:05:06
+- End of SOLICIT message

2. First Relay’s Message

+- Start of RELAY-FORW (12) message (72 bytes)
  | hop-count 0,
  | link-address 4444:1::
  | peer-address fe80::302:3ff:fe04:506
  | relay-message (9) option (34 bytes)
  | +- Start of SOLICIT (1) message (34 bytes)
  |   | transaction-id 12
  |   | ia-na (3) option (12 bytes)
  |   | (iaid 1, t1 0, t2 0)
  |   | client-identifier (1) option (10 bytes)
  |   | 00:03:00:01:01:02:03:04:05:06
  | +- End of SOLICIT message
+- End of RELAY-FORW message

3. Second Relay’s Message

+- Start of RELAY-FORW (12) message (118 bytes)
  | hop-count 1,
  | link-address 4444:2::
  | peer-address fe80::217:a4ff:fe99:50e9
  | relay-message (9) option (72 bytes)
  | +- Start of SOLICIT (1) message (34 bytes)
  |   | transaction-id 12
  |   | ia-na (3) option (12 bytes)
  |   | (iaid 1, t1 0, t2 0)
  |   | client-identifier (1) option (10 bytes)
  |   | 00:03:00:01:01:02:03:04:05:06
  | +- End of SOLICIT message
+- End of RELAY-FORW message

4. Next relay or server
Configuring DHCPv6 Client, Server and Relay Agent

**DHCPv6 Server**

```
ipv6 dhcp pool test-pool
  prefix-delegation pool ipv6-pool
  address prefix FD00:2::1/64
  dns-server 2001:BD8:CAFE::5
  domain-name example.com

ipv6 local pool ipv6-pool 2001:BD8:CAFE:1::/48 64

interface GigabitEthernet1/0
description => relay facing interface
ipv6 address FD00:1::1/64
ipv6 enable
ipv6 dhcp server test-pool
```

**DHCPv6 Relay Agent**

```
interface GigabitEthernet0/0
description => client facing interface
ipv6 address FD00:2::2/64
ipv6 enable
ipv6 dhcp relay destination FD00:1::1

interface GigabitEthernet1/0
description => server facing interface
ipv6 address FD00:1::2/64
ipv6 enable
```

**DHCPv6 Client**

```
interface GigabitEthernet0/0
description => relay facing interface
ipv6 address dhcp
data enable
ipv6 dhcp client pd prefix_delegation_1
```
DHCPv6 Show Commands

DHCPv6_Relay# show ipv6 dhcp relay binding
Relay Bindings associated with default vrf:
Prefix: 2001:422:82:A0::/60 (Ethernet3/1.1)
  DUID: 00030001000574591800
  IAID: 1441793
  lifetime: 1209600
Summary:
  Total number of Relay bindings = 1
  Total number of Relay bindings added by Bulk lease = 0

DHCPv6_Relay# show ipv6 dhcp interface
Ethernet3/1.1 is in relay mode
  Relay destinations:
    2001:0:0:1::5

DHCPv6_Relay# show ipv6 dhcp
This device's DHCPv6 unique identifier(DUID):
  000300010005745D6400

IPv6_Client# show ipv6 dhcp interface
Ethernet3/1.1 is in client mode
  State is OPEN
  Renew will be sent in 3d11h
  List of known servers:
    Reachable via address: FE80::205:74FF:FE5D:6455
    DUID: 0001000113CECE8100237DE49EDC
    Preference: 0
  Configuration parameters:
    IA PD: IA ID 0x00160001, T1 302400, T2 483840
      Prefix: 2001:422:82:A0::/60
        preferred lifetime 604800, valid lifetime 1209600
        expires at Mar 23 2011 06:56 PM (1209451 seconds)
    Prefix name: prefix_delegation_1
  Rapid-Commit: disabled

IPv6_Client# show ipv6 dhcp
This device's DHCPv6 unique identifier(DUID):
  00030001000574591800

IPv6_Client#
Troubleshooting Commands

- Debug commands:
  - debug ipv6 dhcp detail
  - debug ipv6 dhcp relay
  - debug ipv6 dhcp database

- Show commands:
  - show ipv6 dhcp interface
  - show ipv6 dhcp relay binding
  - show ipv6 dhcp binding
  - show ipv6 dhcp pool <test-pool>
  - show ipv6 dhcp conflict
DHCPv6 Bulk Lease Query

- DHCPv6 supports bulk-lease query (rfc5460) which allows a relay agent to request information about DHCPv6 relay bindings from DHCPv6 server(s) in the event of router reload.

- Bulk-lease query is performed using IPv6 link address of the interface for each of the interfaces on which DHCPv6 relay is enabled.

- This functionality adds new query types and allows the bulk transfer of DHCPv6 binding data through TCP.

- Bulk-lease query is enabled by default if the DHCPv6 relay agent is enabled and can be disabled by user via CLI.
DHCPv6 Bulk Lease Query

1. DHCPv6 Relay Reloads
2. DHCPv6 Server responds to the query over TCP with DHCPv6 relay bindings
3. Originates Bulk-lease Query over TCP for each interface on which DHCPv6 relay is enabled
4. Relay: Install DHCPv6 relay binding associated with PD
5. Relay: Install IPv6 static route for the PD

Gateway Router

DHCPv6 Server
Configuring DHCPv6 Bulk-Lease Query

```
router(config)#
ipv6 dhcp-relay bulk-lease {data-timeout seconds | retry number} [disable]
```

- This command configures bulk-lease query parameters used while sending bulk-lease query to the DHCPv6 server.

- The DHCPv6 Bulk-Lease Query feature is enabled automatically when the DHCPv6 relay agent is enabled. The same can be disabled by using “disable” keyword in the above CLI.

- This command provides flexibly to the user to set the retry number and data-time out for the query depending on robustness of the network where the relay is deployed.
Limitations of DHCPv6 Bulk-Lease Query

- This feature is currently supported only by DHCPv6 relay agents running XE3.2 and above release.
- This feature is currently unsupported by Cisco IOS acting as DHCPv6 servers.
- Cisco IOS relay agents XE3.2 and above can be used in conjunction with CNR version 7.2 and above.
Default Router Selection
Default Router Selection

- Hosts maintain a default router list from which one is selected for traffic to off-link destinations and is then cached
  - “round-robin”, or “always the same” selection is implementation dependent
- RFC 4191 – two optional extensions to RA messages
  - Default Router Preferences (DRP): A very coarse preference metric for default routers
  - More-Specific Routes (MSR): More specific routes than the default route, together with a very coarse preference metric for each such route
  - DRP can be implemented without implementing MSR
Default Router Selection (Cont.)

- Default Router Selection
  - Enhances hosts’ selection mechanism from a set of default routers
  - Complementary to mechanisms to improve First Hop Redundancy (ND tuning, HSRP)
Default Router Selection
Example One

- One default router may provide much better performance than another for a destination
- It makes sense to adopt “B” as the default router
If most traffic is routed through “B”, than “B” is least likely to redirect traffic

In order to minimize redirects, it makes sense to adopt “B” as the default router
Default Router Selection

IPv6 Host

A

2 M

B

10 M

IPv6 Network

IPv6 Host

! interface Ethernet0/0
ipv6 nd router-preference High
!

! interface Ethernet0/0
ipv6 nd router-preference Low
!
Default Router Selection

IPv6 Host

A

2 M

B

10 M

IPv6 Network

IPv6 Host

R200#sh ipv6 router
Router FE80::A8BB:CCFF:FE00:CA00 on Ethernet0/0, last update 0 min
Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
HomeAgentFlag=0, Preference=High
Reachable time 15000 msec, Retransmit time 0 msec
Prefix 2001:1::/64 onlink autoconfig
Valid lifetime 2592000, preferred lifetime 604800
Router FE80::A8BB:CCFF:FE00:C900 on Ethernet0/0, last update 2 min
Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
HomeAgentFlag=0, Preference=Low
Reachable time 15000 msec, Retransmit time 0 msec
Prefix 2001:1::/64 onlink autoconfig
Valid lifetime 2592000, preferred lifetime 604800
First Hop Redundancy Protocol – HSRPv6
Hot Standby Routing Protocol (Cont.)

- HSRP for IPv4 and IPv6 have similar state-machine
- HSRP IPv4 differences
  - Host will learn the default gateway through router RA messages (no need to configure default gateway)
  - Active HSRP router will by default send RA every 200 seconds
  - Standby HSRP router will suppress its RA messages
- HSRP for IPv6 vs. IPv6 ND
  - Provides predictable IPv6 Host-to-Router redundancy and faster failover – default 10 seconds vs. default 30 seconds
  - Reduces ND traffic overhead (NS/NA messages) associated with reducing ND Reachable Time timer
Hot Standby Routing Protocol (Cont.)

- Virtual MAC addresses associated with HSRP for IPv6
  - 0005.73A0.0000 … 0005.73A0.0FFF (= 4096 available addresses)
  - HSRP group number (4096) → virtual MAC address → virtual Link Local address (modified EUI-64 derived)

- UDP port 2029 for HSRP packets

  standby version 2

  standby <group> ipv6 {autoconfig | <IPv6 address>}

  Autoconfig creates a Link Local IPv6 address derived from the virtual MAC address through modified EUI-64
HSRP for IPv6

IPv6 Host

HSRP protocol

Standby

R202

IPv6 Network

R201

IPv6 Host

UDP Port 2029

LL Scope Mcast: FF02::66

R202#sh standby
Ethernet0/0 - Group 0 (version 2)
State is Active
2 state changes, last state change 01:10:49
Virtual IP address is FE80::5:73FF:FEA0:1
Active virtual MAC address is 0005.73a0.0000
Local virtual MAC address is 0005.73a0.0000 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.748 secs
Preemption disabled
Active router is local
Standby router is FE80::A8BB:CCFF:FE00:CA00, priority 100 (expires in 8.728 sec)
Priority 200 (configured 200)
IP redundancy name is "hsrp-Et0/0-0" (default)

R200#sh ipv6 routers
Router FE80::5:73FF:FEA0:1 on Ethernet0/0, last update 2 min
   Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0,
   MTU=1500
   HomeAgentFlag=0, Preference=Medium
   Reachable time 0 msec, Retransmit time 0 msec
   Prefix 2001:1::/64 onlink autoconfig
   Valid lifetime 2592000, preferred lifetime 604800

Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Et0/0-0" (default)
HSRP for IPv6 configuration

- Configuration using Link-Local Virtual address

```plaintext
interface FastEthernet0/1
ipv6 address 2001:100:100::2/64
ipv6 cef
standby version 2
standby 1 ipv6 autoconfig
standby 1 timers msec 250 msec 800
standby 1 preempt
standby 1 preempt delay minimum 180
standby 1 authentication md5 key-string cisco
standby 1 track FastEthernet0/0
```

Mandatory before HSRPv6 can be activated

Generate link-local using virtual MAC & group ID
HSRP for IPv6 configuration

- Configuration using Global Virtual address

```conf
interface FastEthernet0/1
ipv6 address 2001:100:100::2/64
ipv6 cef
standby version 2
standby 1 ipv6 2001:100:100::3
standby 1 timers msec 250 msec 800
standby 1 preempt
standby 1 preempt delay minimum 180
standby 1 authentication md5 key-string cisco
standby 1 track FastEthernet0/0
```

Mandatory before HSRPv6 can be activated)

Using Global Virtual address – can be redistributed/injected into IGP
First Hop Security
Attack On Router Discovery

- Attacker tricks victim into accepting him as default router
- Based on rogue Router Advertisements
- The most frequent threat by non-malicious user

Node A sending off-link traffic to C

- RA
  - Src = C's link-local address
  - Dst = All-nodes
  - Data = router lifetime, autoconfig flag
  - Options = subnet prefix, slla

- RA
  - Src = B’s link-local address
  - Dst = All-nodes
  - Data = router lifetime=0
**Attack on Address Configuration**

- Attacker spoofs Router Advertisement with false on-link prefix
- Victim generates IP address with this prefix
- Access router drops outgoing packets from victim (ingress filtering)
- Incoming packets can't reach victim

Node A sourcing off-link traffic to B with BAD::A

B filters out BAD::A

- **Deprecates** X::A
  - RA
    - Src = B’s link-local address
    - Dst = All-nodes
    - Options = prefix X Preferred lifetime = 0
- **Computes** BAD::A and DAD it
  - RA
    - Src = B’s link-local address
    - Dst = All-nodes
    - Options = prefix BAD, Preferred lifetime
Attack On Address Resolution

- Attacker can claim victim's IP address

- NS
  - Dst = Solicited-node multicast address of B
  - Query = what is B’s link-layer address?

- NA
  - Src = B or any C’s IF address
  - Dst = A
  - Data = B
  - Option = link-layer address of C
Attack On DAD

- Attacker hacks any victim's DAD attempts
- Victim can't configure IP address and can't communicate

```
Src = UNSPEC
Dst = Solicited-node multicast address of A
Data = A
Query = Does anybody use A already?
```

```
NA  "it’s mine!"
```

```
Src = any C's IF address
Dst = A
Data = A
Option = link-layer address of C
```
Redirect Attack

- Attacker tricks nodes on the link into accepting him as default router
- Attacker redirects traffic to victim’s link-layer address

Node A sending off-link (to X) traffic to C
Node A sending off-link (to X) traffic to C

Redirect

Gateway

RA

Src = C’s link-local address
Dst = All-nodes
Data = router lifetime
Options = slla

Src = C link-local address
Dst = A
Data = target B, Destination X
Option = link-layer address of B
DoS Attacks On Neighbor Cache

X scanning 2^64 addresses

- **PFX::/64**
- **Gatew**

- **NS**
  - Dst = Solicited-node multicast address of PFX::a
  - Query = what is PFX::a’s link-layer address?

- **NS**
  - Dst = Solicited-node multicast address of PFX::b
  - Query = what is PFX::b’s link-layer address?

- **NS**
  - Dst = Solicited-node multicast address of PFX::z
  - Query = what is PFX::z’s link-layer address?

- **3 seconds history**
NDP Spoofing: Mitigation

- **NDP Inspection**: similar like dynamic ARP inspection for IPv4
  NDP Inspection support on 12.2(50)SY(6500-Sup2T)

- **Secure Neighbor Discovery**
  SEND = NDP + crypto
  IOS 12.4(24)T
  **But** not in Windows Vista, 2008 and 7, MAC OS
  Crypto means slower !...

- **Other Features**:
  Private VLAN works with IPv6
  Port security works with IPv6
  801.x works with IPv6
  Other mitigation methods – PACL, RA Guard, support on 4500/12.2(54)SG
  Nexus 7000/12.2(33)SXI4, 6500/12.2(50)SY
Secure Neighbor Discovery (SEND)  
RFC 3971

- Certification paths  
  Anchored on trusted parties, expected to certify the authority of the routers on some prefixes

- Cryptographically Generated Addresses (CGA)  
  IPv6 addresses whose interface identifiers are cryptographically generated

- RSA signature option  
  Protect all messages relating to neighbor and router discovery

- Timestamp and nonce options  
  Prevent replay attacks

- Requires IOS 12.4(24)T,
Cryptographically Generated Addresses
CGA RFC 3972 (Simplified)

- Each device has a RSA key pair (no need for cert)
- Ultra light check for validity
- Prevent spoofing a valid CGA address

RSA Keys
Priv Pub

Modifier
Public Key
Subnet Prefix

SHA-1

CGA Params

Signature

SEND Messages

Crypto. Generated Address
SeND in Cisco IOS 12.4(24)T
Configuration

crypto key generate rsa label SEND modulus 1024
The name for the keys will be: SEND
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

ipv6 cga modifier rsakeypair SEND sec-level 1

ipv6 nd secured sec-level minimum 1

ipv6 nd secured full-secure

interface fastethernet 0/0
    ipv6 cga rsakeypair SEND
    ipv6 address 2001:db8::/64 cga
Secure Neighbor Discovery: Caveats

- Private/public key pair on all devices for CGA
- Overhead introduced
  - Routers have to do many public/private key calculation
    (some may be done in advance of use)
  - => Potential DoS target
  - Routers need to keep more state
- Available:
  - Unix (DoCoMo)
  - Cisco IOS 12.4(24)T
- Microsoft:
  - no support in Vista, Windows 2008 and Windows7, MAC OS
FHS features in to-be-released code

- Port ACL
- ACL Based RA Guard
- ACL based DHCP Guard
- RA Guard
- NDP Inspection
- Device Tracking
RA Guard
ACL Based

Benefits:
- Simple validation criteria
- Cryptography and Certificate Free

Solution:
- RA filtered based on ingress port ACL
- Simple criteria based on IPv6 Header fields, ICMP type and code
- Configuration may be heavy
- Very Static and Less Secure
- Stateless
**DHCP Guard**

**ACL Based**

**Solution:**
- Simple criteria based on UDP port (546)
  - Configuration may be heavy
  - Very Static and Less Secure
  - Stateless

**Benefits:**
- Simple validation criteria
RA Guard Configuration Based

Benefits:

- Improved Security
- Cryptography and Certificate Free

Solution:

- Switch intercepts RAs and validates RAs using more complex criteria in configuration on switch
- Blocks RAs until verification succeeds
- Stateful and dynamic

Allowed parameters:

```
interface FastEthernet3/13
switchport access vlan 222
switchport mode access
ipv6 nd raguard
```

```
GLOBAL config:
ipv6 nd raguard policy policy-name
device-role {host | monitor | router | switch}
match ra prefix-list ipv6-prefix-list-name
match ipv6 access-list ipv6-access-list-name
hop-limit {maximum limit | minimum limit}
managed-config-flag {on | off}
other-config-flag {on | off}
router-preference maximum {high | low | medium}
trusted-port
```

```
Interface Config: (Router port):
Router(config-if)# ipv6 nd raguard attach-policy
```
Neighbor Discovery Protocol Inspection

Benefits:
- Prevents spoofing attacks due to NDP vulnerabilities
- Learns and Secures bindings for Stateless auto configuration Addresses in L2 Neighbor Cache
- Mechanism to dynamically activate authorization policies for guard features

Solution:
- Switch intercepts NDP messages
  - Verifies sender’s identity (signature) and authority (CGA and/or Trust Anchor based)
  - Filters NDP messages if verification fails
- Secures port ⇐ LLA ⇐ IPA binding in neighbor cache
NDP Spoofing: Mitigation summary

- **NDP Inspection**: similar like dynamic ARP inspection for IPv4
  NDP Inspection support on 12.2(50)SY(6500-Sup2T)

- **Secure Neighbor Discovery**
  SEND = NDP + crypto
  IOS 12.4(24)T
  **But** not in Windows Vista, 2008 and 7, MAC OS
  Crypto means slower !...

- **Other Features**:
  Private VLAN works with IPv6
  Port security works with IPv6
  801.x works with IPv6
  Other mitigation methods – PACL, RA Guard(Host) support on 4500/12.2(54)SG
  6500/12.2(33)SXI4, RA Guard(Router,Host) 6500/12.2(50)SY

  PACL/VACL support on Nexus 7000 from 4.1

  PACL support on 3750/3560 (all feature set) from 12.2(50)SE1, 3750-X(12.2(53)SE2)

  FHS features are on the roadmap for all platforms in phases, watch new features in coming releases, use feature navigator