Implementing First Hop Security in IPv6

Last Updated: December 1, 2011

This document provides information about configuring features that comprise first hop security functionality in IPv6.

A database table of IPv6 neighbors connected to the switch is created from information sources such as Neighbor Discovery (ND) protocol snooping. This database, or binding, table is used by various IPv6 guard features (such as IPv6 ND Inspection, per-port address limit, IPv6 device tracking) to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

IPv6 ND Inspection learns and secures bindings for stateless autoconfiguration addresses in layer 2 neighbor tables. IPv6 ND inspection analyzes neighbor discovery messages in order to build a trusted binding table database, and IPv6 neighbor discovery messages that do not conform are dropped.

Router advertisements (RAs) are used by routers to announce themselves on the link. IPv6 RA Guard analyzes these RAs and can filter out bogus ones sent by unauthorized routers.

The per-port address limit feature enables an operator to specify a maximum number of IPv6 addresses allowed on a port of the switch. This function is achieved by filtering out ND messages sourced with addresses beyond the per-port address limit.

IPv6 Device Tracking provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

The Secure Neighbor Discovery for Cisco IOS Software feature is designed to counter the threats of the ND protocol. Secure neighbor discovery (SeND) defines a set of neighbor discovery options and two neighbor discovery messages. SeND also defines a new autoconfiguration mechanism to establish address ownership. The IPv6 PACL feature adds IPv6 port-based ACL support.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Implementing First Hop Security in IPv6

- You should be familiar with the IPv6 neighbor discovery feature. For information about IPv6 neighbor discovery, see "Implementing IPv6 Addressing and Basic Connectivity".
- The SeND feature is available on crypto images because it involves using cryptographic libraries.
- In order to use IPv6 port-based access list (PACL), you must know how to configure IPv6 access lists. For information about configuring IPv6 access lists, see "Implementing Traffic Filters and Firewalls for IPv6 Security".

Restrictions for Implementing First Hop Security in IPv6

The IPv6 PACL feature is supported only in the ingress direction; it is not supported in the egress direction.

RA Guard in Cisco IOS Release 12.2(33)SXI4

- The RA guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- This feature is supported only in hardware by programming the TCAM.
- This feature can be configured only on a switchport interface in the ingress direction.
- This feature supports only host mode.
- This feature is supported only in the ingress direction; it is not supported in the egress direction.
- This feature is supported on ether channel, but not on ether channel port members.
- This feature is not supported on trunk ports with merge mode.
- This feature is supported on auxiliary VLANs and PVLANs. In case of PVLANs, primary VLAN features are inherited and merged with port features.
- Packets dropped by the RA guard feature can be spanned.
- If the **platform ipv6 acl icmp optimize neighbor-discovery** command is configured, RA guard feature configuration should not be allowed and an error message should be displayed. This command adds default global ICMP entries that will override the RA guard ICMP entries.

Information About Implementing First Hop Security in IPv6

IPv6 First-Hop Security Binding Table

A database table of IPv6 neighbors connected to the switch is created from information sources such as Neighbor Discovery (ND) protocol snooping. This database, or binding, table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

IPv6 Device Tracking

The IPv6 device tracking feature provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears. The feature tracks of the liveness of the neighbors connected through the Layer 2 switch on regular basis in order to revoke network access privileges as they become inactive.

IPv6 Port-Based Access List Support

The IPv6 PACL feature provides the ability to provide access control (permit or deny) on L2 switch ports for IPv6 traffic. IPv6 PACLs are similar to IPv4 PACLs, which provide access control on L2 switch ports for IPV4 traffic. They are supported only in ingress direction and in hardware.

PACL can filter ingress traffic on L2 interfaces based on L3 and L4 header information or non-IP L2 information.

IPv6 Global Policies

IPv6 global policies provide policy database services to features with regard to storing and accessing those policies. IPv6 ND inspection and IPv6 RA guard are IPv6 global policies features. Every time an ND inspection or RA guard is configured globally, the attributes of the policy are stored in the software policy database. The policy is then applied to an interface, and the software policy database entry is updated to include this interface to which the policy is applied.

IPv6 RA Guard

IPv6 RA guard provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network switch platform. RAs are used by routers to announce themselves on the link. The RA Guard feature analyzes these RAs and filters out bogus RAs sent by unauthorized routers. In host mode, all router advertisement and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the L2 device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

IPv6 ND Inspection

IPv6 ND inspection learns and secures bindings for stateless autoconfiguration addresses in layer 2 neighbor tables. IPv6 ND inspection analyzes neighbor discovery messages in order to build a trusted

binding table database, and IPv6 neighbor discovery messages that do not conform are dropped. SA neighbor discovery message is considered trustworthy if its IPv6-to-Media Access Control (MAC) mapping is verifiable.

This feature mitigates some of the inherent vulnerabilities for the neighbor discovery mechanism, such as attacks on duplicate address detection (DAD), address resolution, router discovery, and the neighbor cache.

Secure Neighbor Discovery in IPv6

IPv6 Neighbor Discovery Trust Models and Threats

There are three IPv6 neighbor discovery trust models, which are described as follows:

- All authenticated nodes trust each other to behave correctly at the IP layer and not to send any neighbor discovery or router discovery (RD) messages that contain false information. This model represents a situation where the nodes are under a single administration and form a closed or semiclosed group. A corporate intranet is an example of this model.
- A router trusted by the other nodes in the network to be a legitimate router that routes packets between the local network and any connected external networks. This router is trusted to behave correctly at the IP layer and not to send any neighbor discovery or RD messages that contain false information. This model represents a public network run by an operator. The clients pay the operator, have the operator's credentials, and trust the operator to provide the IP forwarding service. The clients do not trust each other to behave correctly; any other client node must be considered able to send falsified neighbor discovery and RD messages.
- A model where the nodes do not directly trust each other at the IP layer. This model is considered suitable where a trusted network operator is not available.

Nodes on the same link use ND to discover each other's presence and link-layer addresses, to find routers, and to maintain reachability information about the paths to active neighbors. ND is used by both hosts and routers. The original ND specifications used IPsec to protect ND messages. However, not many detailed instructions for using IPsec are available. The number of manually configured security associations needed for protecting ND can be very large, which makes that approach impractical for most purposes. These threats need to be considered and eliminated.

SeND Protocol

The SeND protocol counters ND threats. It defines a set of new ND options, and two new ND messages (Certification Path Solicitation [CPS] and Certification Path Answer [CPA]). It also defines a new autoconfiguration mechanism to be used in conjunction with the new ND options to establish address ownership.

SeND defines the mechanisms defined in the following sections for securing ND:

Cryptographically Generated Addresses in SeND

Cryptographically generated addresses (CGAs) are IPv6 addresses generated from the cryptographic hash of a public key and auxiliary parameters. This provides a method for securely associating a cryptographic public key with an IPv6 address in the SeND protocol.

The node generating a CGA address must first obtain a Rivest, Shamir, and Adelman (RSA) key pair (SeND uses an RSA public/private key pair). The node then computes the interface identifier part (which is the rightmost 64 bits) and appends the result to the prefix to form the CGA address.

CGA address generation is a one-time event. A valid CGA cannot be spoofed and the CGA parameters received associated to it is reused because the message must be signed with the private key that matches the public key used for CGA generation, which only the address owner will have.

A user cannot replay the complete SeND message (including the CGA address, CGA parameters, and CGA signature) because the signature has only a limited lifetime.

Authorization Delegation Discovery

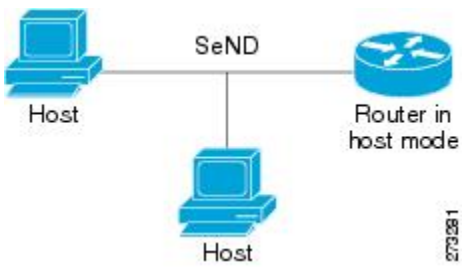
Authorization delegation discovery is used to certify the authority of routers by using a trust anchor. A trust anchor is a third party that the host trusts and to which the router has a certification path. At a basic level, the router is certified by the trust anchor. In a more complex environment, the router is certified by a user that is certified by the trust anchor. In addition to certifying the router identity (or the right for a node to act as a router), the certification path contains information about prefixes that a router is allowed to advertise in router advertisements. Authorization delegation discovery enables a node to adopt a router as its default router.

SeND Deployment Models

Host-to-Host Deployment Without a Trust Anchor

Deployment for SeND between hosts is straightforward. The hosts can generate a pair of RSA keys locally, autoconfigure their CGA addresses, and use them to validate their sender authority, rather than using a trust anchor to establish sender authority. The figure below illustrates this model.

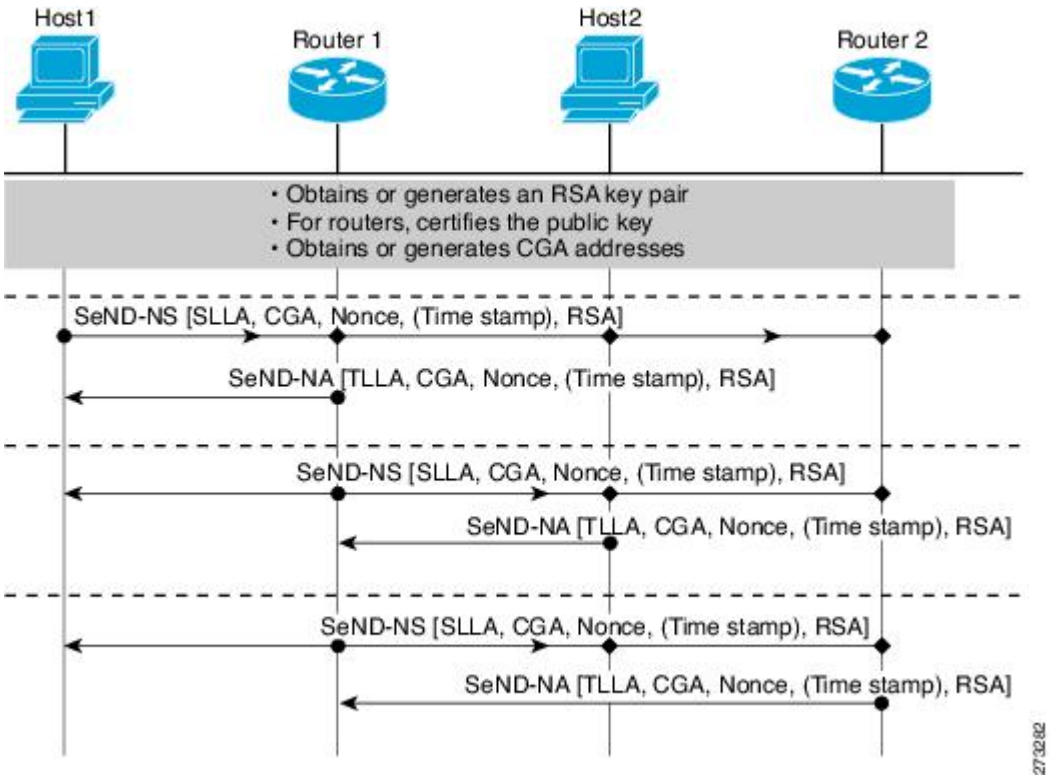
Figure 1 Host-to-Host Deployment Model



Neighbor Solicitation Flow

In a neighbor solicitation scenario, hosts and routers in host mode exchange neighbor solicitations and neighbor advertisements. These neighbor solicitations and neighbor advertisements are secured with CGA addresses and CGA options, and have nonce, time stamp, and RSA neighbor discovery options. The figure below illustrates this scenario.

Figure 2 Neighbor Solicitation Flow

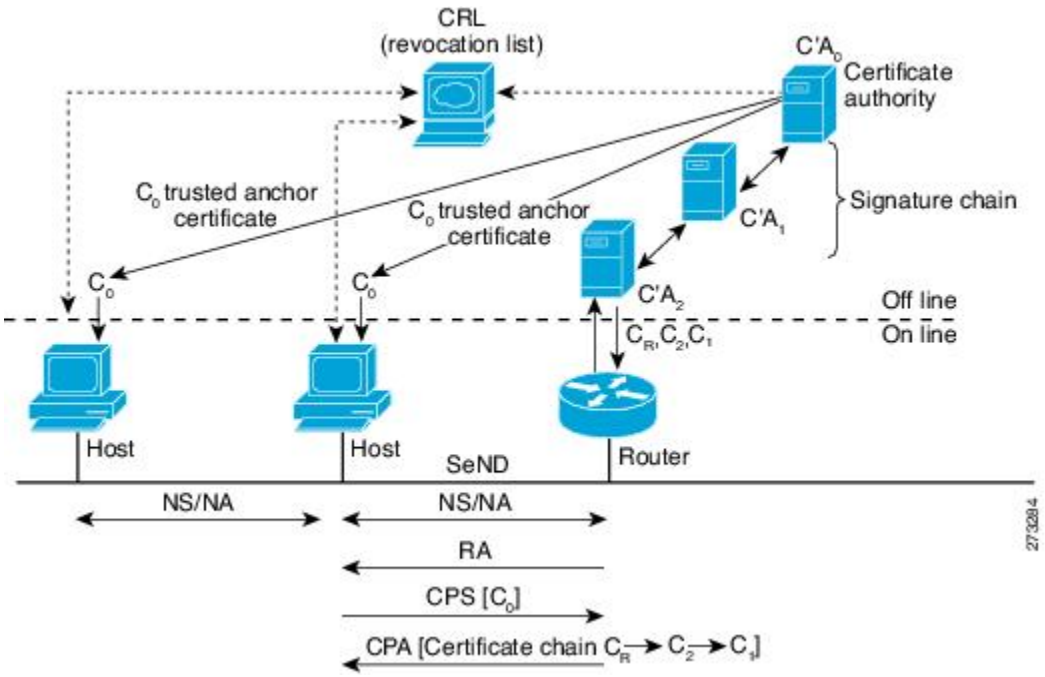


Host-Router Deployment Model

In many cases, hosts will not have access to the infrastructure that will enable them to obtain and announce their certificates. In these situations, hosts will secure their relationship using CGA, and secure their

relationship with routers using a trusted anchor. When using RAs, SeND mandates that routers are authenticated through a trust anchor. The figure below illustrates this scenario.

Figure 3 Host-Router Deployment Model

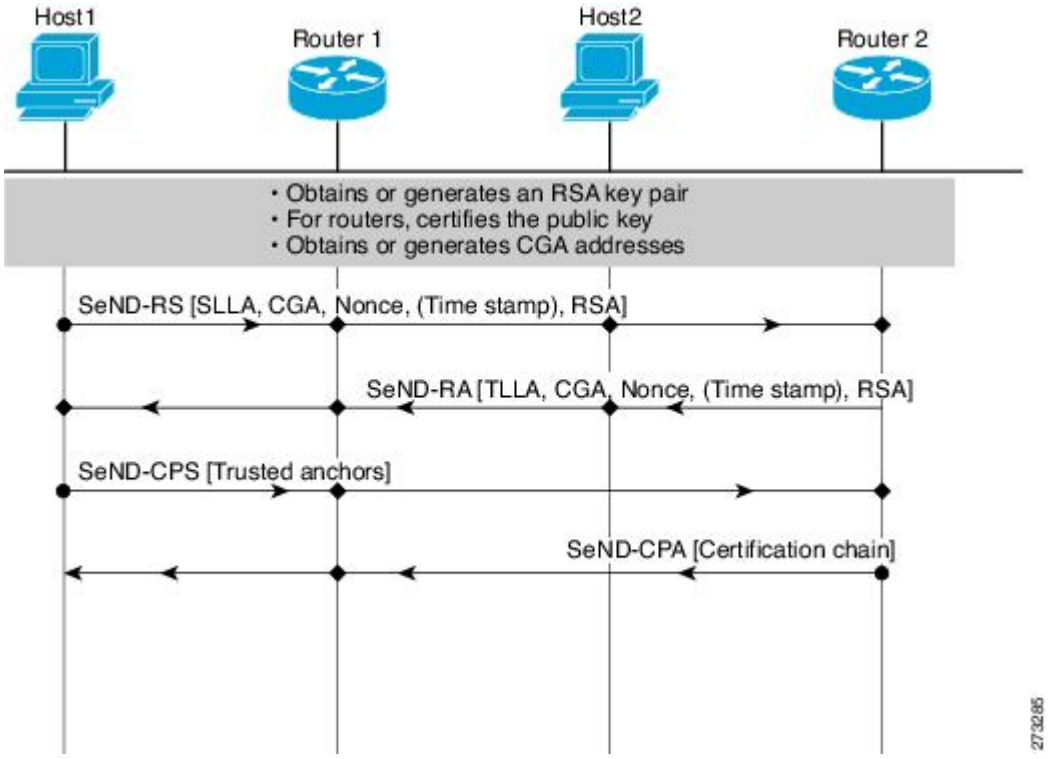


Router Advertisement and Certificate Path Flows

The figure below shows the certificate exchange performed using certification path solicitation CPS/CPA SeND messages. In the illustration, Router R is certified (using an X.509 certificate) by its own CA (certificates CR). The CA itself (CA2) is certified by its own CA (certificates C2), and so on, up to a CA (CA0) that the hosts trusts. The certificate CR contains IP extensions per RFC 3779, which describes which prefix ranges the router R is allowed to announce (in RAs). This prefix range, certified by CA2, is a subset

of CA2's own range, certified by CA1, and so on. Part of the validation process when a certification chain is received consists of validating the certification chain and the consistency of nested prefix ranges.

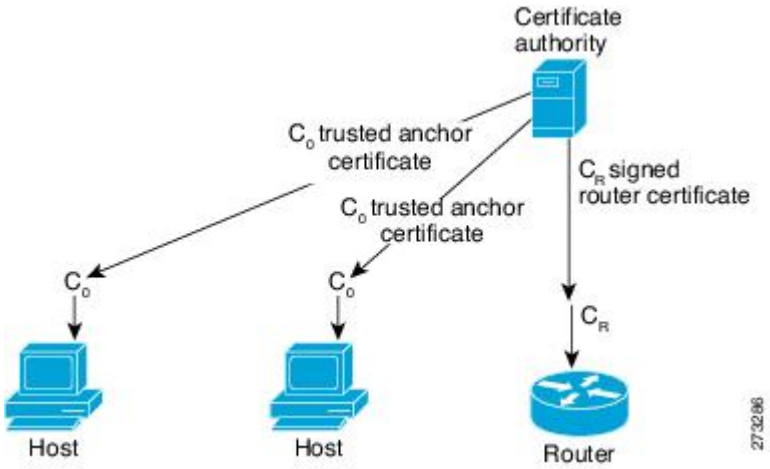
Figure 4 Router Advertisement and Certificate Path Flows



Single CA Model

The deployment model shown in the third figure above can be simplified in an environment where both hosts and routers trust a single CA such as the Cisco certification server (CS). The figure below illustrates this model.

Figure 5 Single CA Deployment Model



How to Implement First Hop Security in IPv6

Configuring the IPv6 Binding Table Content

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor binding vlan** *vlan-id* {**interface** *type number* | *ipv6-address* | *mac-address*} [**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*]
4. **ipv6 neighbor binding max-entries** *entries* [**vlan-limit** *number* | **interface-limit** *number* | **mac-limit** *number*]
5. **ipv6 neighbor binding logging**
6. **exit**
7. **show ipv6 neighbor binding** [**vlan** *vlan-id* | **interface** *type number* | **ipv6** *ipv6-address* | **mac** *mac-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ipv6 neighbor binding vlan** *vlan-id* {**interface** *type number* | *ipv6-address* | *mac-address*} [**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*] **Example:** `Router(config)# ipv6 neighbor binding reachable-entries 100`	Adds a static entry to the binding table database.
Step 4	**ipv6 neighbor binding max-entries** *entries* [**vlan-limit** *number* | **interface-limit** *number* | **mac-limit** *number*] **Example:** `Router(config)# ipv6 neighbor binding max-entries`	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 5	**ipv6 neighbor binding logging** **Example:** `Router(config)# ipv6 neighbor binding logging`	Enables the logging of binding table main events.
Step 6	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode, and places the router in privileged EXEC mode.
Step 7	**show ipv6 neighbor binding** [**vlan** *vlan-id* | **interface** *type number* | **ipv6** *ipv6-address* | **mac** *mac-address*] **Example:** `Router# show ipv6 neighbor binding`	Displays contents of a binding table.

Configuring IPv6 Device Tracking

Perform this task to provide fine grain control over the life cycle of an entry in the binding table for the IPv6 device tracking feature. This feature is available in Cisco IOS Release 12.2(50)SY. In order for IPv6 device tracking to work, the binding table needs to be populated (see the Configuring the IPv6 Binding Table Content, page 9).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 neighbor tracking** [**retry-interval** *value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 neighbor tracking** [**retry-interval** *value*] **Example:** `Router(config)# ipv6 neighbor tracking`	Tracks entries in the In order for this feat.

Configuring IPv6 ND Inspection

Configuring IPv6 ND Inspection Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd inspection policy** *policy-name*
4. **drop-unsecure**
5. **sec-level minimum** *value*
6. **device-role** {**host** | **monitor** | **router**}
7. **tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]}
8. **trusted-port**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd inspection policy** *policy-name* **Example:** `Router(config)# ipv6 nd inspection policy policy1`	Defines the ND inspection policy name and enters the router into ND inspection policy configuration mode.
Step 4	**drop-unsecure** **Example:** `Router(config-nd-inspection)# drop-unsecure`	Drops messages with no or invalid options or an invalid signature.
Step 5	**sec-level minimum** *value* **Example:** `Router(config-nd-inspection)# sec-level minimum 2`	Specifies the minimum security level parameter value when CGA options are used.
Step 6	**device-role** {**host** | **monitor** | **router**} **Example:** `Router(config-nd-inspection)# device-role monitor`	Specifies the role of the device attached to the port.
Step 7	**tracking** {**enable** [**reachable-lifetime** {*value* | **infinite**}] | **disable** [**stale-lifetime** {*value* | **infinite**}]} **Example:** `Router(config-nd-inspection)# tracking disable stale-lifetime infinite`	Overrides the default tracking policy on a port.

	Command or Action	Purpose
Step 8	**trusted-port** **Example:** `Router(config-nd-inspection)# trusted-port`	Configures a port to become a trusted port.

Applying IPv6 ND Inspection on a Specified Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd inspection** [**attach-policy** [**policy** *policy-name*] | **vlan** {**add** | **except** | **none** | **remove**| **all**} *vlan* [*vlan1, vlan2, vlan3...*]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 nd inspection** [**attach-policy** [**policy** *policy-name*]	**vlan** {**add**	**except**	**none**	**remove**| **all**} *vlan* [*vlan1, vlan2, vlan3...*]]] **Example:** `Router(config-if)# ipv6 nd inspection`	Applies the ND inspection feature on the interface.

Verifying and Troubleshooting IPv6 ND Inspection

SUMMARY STEPS

1. **enable**
2. **show ipv6 snooping capture-policy** [**interface** *type number*]
3. **show ipv6 snooping counter** [**interface** *type number*]
4. **show ipv6 snooping features**
5. **show ipv6 snooping policies** [**interface** *type number*]
6. **debug ipv6 snooping** [**binding-table** | **classifier** | **errors** | **feature-manager** | **filter** *acl* | **ha** | **hw-api** | **interface** *interface* | **memory** | **ndp-inspection** | **policy** | **vlan** *vlanid* | **switcher** | **filter** *acl* | **interface** *interface* | *vlanid*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 snooping capture-policy** [**interface** *type number*] **Example:** `Router# show ipv6 snooping capture-policy interface ethernet 0/0`	Displays snooping ND message capture policies
Step 3	**show ipv6 snooping counter** [**interface** *type number*] **Example:** `Router# show ipv6 snooping counters interface Fa4/12`	Displays information about the packets counted by the interface counter.
Step 4	**show ipv6 snooping features** **Example:** `Router# show ipv6 snooping features`	Displays information about snooping features configured on the router.
Step 5	**show ipv6 snooping policies** [**interface** *type number*] **Example:** `Router# show ipv6 snooping policies`	Displays information about the configured policies and the interfaces to which they are attached.

	Command or Action	Purpose
Step 6	**debug ipv6 snooping** [**binding-table** | **classifier** | **errors** | **feature-manager** | **filter** *acl* | **ha** | **hw-api** | **interface** *interface* | **memory** | **ndp-inspection** | **policy** | **vlan** *vlanid* | **switcher** | **filter** *acl* | **interface** *interface* | *vlanid* **Example:** `Router# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

Configuring IPv6 RA Guard

Applying IPv6 RA Guard on a Specified Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd raguard attach-policy** [*policy-name* [**vlan** {**add**| **except** | **none** | **remove** | **all**} *vlan*[*vlan1, vlan2, vlan3*...]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface Gigabit 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 nd raguard attach-policy** [*policy-name* [**vlan** {**add**| **except** | **none** | **remove** | **all**} *vlan*[*vlan1, vlan2, vlan3...*]]] **Example:** `Router(config-if)# ipv6 nd raguard attach-policy`	Applies the RA guard feature on a specified interface.

• Configuring IPv6 RA Guard in Cisco IOS Release 12.2(33)SXI4 and 12.2(54)SG, page 16

Configuring IPv6 RA Guard in Cisco IOS Release 12.2(33)SXI4 and 12.2(54)SG

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd raguard**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 nd raguard** **Example:** `Router(config-if)# ipv6 nd raguard`	Applies the IPv6 RA guard feature.

Verifying and Troubleshooting IPv6 RA Guard

SUMMARY STEPS

1. **enable**
2. **show ipv6 nd raguard policy** [*policy-name*]
3. **debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 nd raguard policy** [*policy-name*] **Example:** `Router# show ipv6 nd raguard policy raguard1`	Displays RAs guard policy on all interfaces configured with RA guard.
Step 3	**debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*] **Example:** `Router# debug ipv6 snooping raguard`	Enables debugging for snooping information in the IPv6 RA guard feature

Configuring SeND for IPv6

Certificate servers are used to grant certificates after validating or certifying key pairs. A tool for granting certificates is mandatory in any SeND deployment. Many tools are available to grant certificates, for example, Open Secure Sockets Layer (OpenSSL) on Linux. However, very few certificate servers support granting certificates containing IP extensions. Cisco IOS certificate servers support every kind of certificate including the certificates containing the IP extensions.

SeND is available in host mode. The set of available functions on a host will be a subset of SeND functionality. CGA will be fully available and the prefix authorization delegation will be supported on the host side (sending CPS and receiving CPA).

To implement SeND, configure the host with the following parameters:

- An RSA key pair used to generate CGA addresses on the interface.
- A SeND modifier that is computed using the RSA key pair.
- A key on the SeND interface.
- CGAs on the SeND interface.
- A Public Key Infrastructure (PKI) trustpoint, with minimum content; for example, the URL of the certificate server. A trust anchor certificate must be provisioned on the host.

SeND is also available in router mode. You can use the **ipv6 unicast-routing** command to configure a node to a router. To implement SeND, configure routers with the same elements as that of the host. The routers will need to retrieve certificates of their own from a certificate server. The RSA key and subject name of the trustpoint are used to retrieve certificates from a certificate server. Once the certificate has been obtained and uploaded, the router generates a certificate request to the certificate server and installs the certificate.

The following operations need to be completed before SeND is configured on the host or router:

- Hosts are configured with one or more trust anchors.
- Hosts are configured with an RSA key pair or configured with the capability to locally generate it. Note that for hosts not establishing their own authority via a trust anchor, these keys are not certified by any CA.
- Routers are configured with RSA keys and corresponding certificate chains, or the capability to obtain these certificate chains that match the host trust anchor at some level of the chain.

While booting, hosts and routers must either retrieve or generate their CGAs. Typically, routers will autoconfigure their CGAs once and save them (along with the key pair used in the CGA operation) into their permanent storage. At a minimum, link-local addresses on a SeND interface should be CGAs. Additionally, global addresses can be CGAs.

- Configuring Certificate Servers to Enable SeND, page 18
- Configuring a Host to Enable SeND, page 21
- Configuring a Router to Enable SeND, page 24
- Implementing IPv6 SeND, page 27
- Configuring SeND Parameters, page 33

Configuring Certificate Servers to Enable SeND

Hosts and routers must be configured with RSA key pairs and corresponding certificate chains before the SeND parameters are configured. Perform the following task to configure the certificate server to grant certificates. Once the certificate server is configured, other parameters for the certificate server can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki trustpoint** *name*
5. **ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*}
6. **revocation-check** {[**crl**] [**none**] [**ocsp**]}
7. exit
8. **crypto pki server** *name*
9. **grant auto**
10. **cdp-url** *url-name*
11. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip http server** **Example:** `Router(config)# ip http server`	Configures the HTTP server.
Step 4	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint CA`	(Optional) Declares the trustpoint that your certificate server should use and enters ca-trustpoint configuration mode. • If you plan to use X.509 IP extensions, use this command. To automatically generate a CS trustpoint, go to Step 8 .

	Command or Action	Purpose
Step 5	**ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*} **Example:** `Router(ca-trustpoint)# ip-extension prefix 2001:100::/32`	(Optional) Specifies that the IP extensions are included in a certificate request either for enrollment or generation of a certificate authority (CA) for the Cisco IOS CA.
Step 6	**revocation-check** {[**crl**] [**none**] [**ocsp**]} **Example:** `Router(ca-trustpoint)# revocation-check crl`	(Optional) Sets one or more methods for revocation checking.
Step 7	exit **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 8	**crypto pki server** *name* **Example:** `Router(config)# crypto pki server CA`	Configures the PKI server and places the router in server configuration mode.
Step 9	**grant auto** **Example:** `Router(config-server)# grant auto`	(Optional) Grants all certificate requests automatically.
Step 10	**cdp-url** *url-name* **Example:** `Router(config-server)# cdp-url http://209.165.202.129/CA.crl`	(Optional) Sets the URL name if the host is using a Certificate Revocation List (CRL).
Step 11	**no shutdown** **Example:** `Router(config-server)# no shutdown`	Enables the certificate server.

Configuring a Host to Enable SeND

SeND is available in host mode. Before you can configure SeND parameters in host mode, first configure the host using the following commands. Once the host has been configured, SeND parameters can be configured on it.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**][**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:**]
4. **ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}
5. **crypto pki trustpoint** *name*
6. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [pem]
7. **revocation-check {[crl] [none] [ocsp]}**
8. exit
9. **crypto pki authenticate** *name*
10. **ipv6 nd secured sec-level minimum** *value*
11. **interface** *type number*
12. **ipv6 cga rsakeypair** *key-label*
13. **ipv6 address** *ipv6-address* / *prefix-length* **link-local cga**
14. **ipv6 nd secured trustanchor** *trustanchor-name*
15. **ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value*}
16. **exit**
17. **ipv6 nd secured full-secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Host> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Host# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**][**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:**] **Example:** `Host(config)# crypto key generate rsa label SEND modulus 1024`	Configures the RSA key.
Step 4	**ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**} **Example:** `Host(config)# ipv6 cga modifier rsakeypair SEND sec-level 1`	Enables the RSA key to be used by SeND (generates the modifier).
Step 5	**crypto pki trustpoint** *name* **Example:** `Host(config)# crypto pki trustpoint SEND`	Specifies the node trustpoint and enters ca-trustpoint configuration mode.
Step 6	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [pem] **Example:** `Host(ca-trustpoint)# enrollment url http://209.165.200.254`	Specifies the enrollment parameters of a CA.
Step 7	**revocation-check {[crl] [none] [ocsp]}** **Example:** `Host(ca-trustpoint)# revocation-check none`	Sets one or more methods of revocation.
Step 8	exit **Example:** `Host(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 9	**crypto pki authenticate** *name* **Example:** `Host(config)# crypto pki authenticate SEND`	Authenticates the certification authority (by getting the certificate of the CA).

	Command or Action	Purpose
Step 10	**ipv6 nd secured sec-level minimum** *value* **Example:** `Host(config)# ipv6 nd secured sec-level minimum 1`	(Optional) Configures CGA. • You can provide additional parameters such as security level and key size. • In the example, the security level accepted by peers is configured.
Step 11	**interface** *type number* **Example:** `Host(config)# interface fastethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 12	**ipv6 cga rsakeypair** *key-label* **Example:** `Host(config-if)# ipv6 cga rsakeypair SEND`	(Optional) Configures CGA on interfaces.
Step 13	**ipv6 address** *ipv6-address* / *prefix-length* **link-local cga** **Example:** `Host(config-if)# ipv6 address FE80::260:3EFF:FE11:6770/23 link-local cga`	Configures an IPv6 link-local address for the interface and enables IPv6 processing on the interface.
Step 14	**ipv6 nd secured trustanchor** *trustanchor-name* **Example:** `Host(config-if)# ipv6 nd secured trustanchor SEND`	(Optional) Configures trusted anchors to be preferred for certificate validation.
Step 15	**ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value*} **Example:** `Host(config-if)# ipv6 nd secured timestamp delta 300`	(Optional) Configures the timing parameters.
Step 16	**exit** **Example:** `Host(config-if)# exit`	Returns to global configuration mode.

Command or Action	Purpose
Step 17 **ipv6 nd secured full-secure** **Example:** `Host(config)# ipv6 nd secured full-secure`	(Optional) Configures general SeND parameters. • In the example, secure mode is configured on SeND.

Configuring a Router to Enable SeND

SeND is available in the router mode. Perform this task before you can configure SeND parameters in router mode. Once the router has been configured, the SeND parameters can be configured on it.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**][**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}
5. **crypto pki trustpoint** *name*
6. **subject-name** [**attr** *tag*][**eq** | **ne** | **co** | **nc**] *string*
7. **rsakeypair** *key-label*
8. **revocation-check** {[**crl**][**none**][**ocsp**]}
9. exit
10. **crypto pki authenticate** *name*
11. **crypto pki enroll** *name*
12. **ipv6 nd secured sec-level minimum** *value*
13. **interface** *type number*
14. **ipv6 cga rsakeypair** *key-label*
15. **ipv6 address** *ipv6-address* / *prefix-length* **link-local cga**
16. **ipv6 nd secured trustanchor** *trustpoint-name*
17. **ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value*}
18. **exit**
19. **ipv6 nd secured full-secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**][**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*] **Example:** `Router(config)# crypto key generate rsa label SEND modulus 1024`	Configures the RSA key.
Step 4	**ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**} **Example:** `Router(config)# ipv6 cga modifier rsakeypair SEND sec-level 1`	Enables the RSA key to be used by SeND (generates the modifier).
Step 5	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint SEND`	Configures PKI for a single or multiple-tier CA, specifies the router trustpoint, and places the router in ca-trustpoint configuration mode.
Step 6	**subject-name** [**attr** *tag*][**eq** | **ne** | **co** | **nc**] *string* **Example:** `Router(ca-trustpoint)# subject-name C=FR, ST=PACA, L=Example, O=Cisco, OU=NSSTG, CN=router`	Creates a rule entry.

	Command or Action	Purpose
Step 7	**rsakeypair** *key-label* **Example:** `Router(ca-trustpoint)# rsakeypair SEND`	Binds the RSA key pair for SeND.
Step 8	**revocation-check** {[**crl**][**none**][**ocsp**]} **Example:** `Router(ca-trustpoint)# revocation-check none`	Sets one or more methods of revocation.
Step 9	exit **Example:** `host(ca-trustpoint)# exit`	Returns to global configuration mode.
Step 10	**crypto pki authenticate** *name* **Example:** `host(config)# crypto pki authenticate SEND`	Authenticates the certification authority (by getting the certificate of the CA).
Step 11	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll SEND`	Obtains the certificates for the router from the CA.
Step 12	**ipv6 nd secured sec-level minimum** *value* **Example:** `Router(config)# ipv6 nd secured sec-level minimum 1`	(Optional) Configures CGA and provides additional parameters such as security level and key size. • In the example, the minimum security level that SeND accepts from its peers is configured.
Step 13	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 14	**ipv6 cga rsakeypair** *key-label* **Example:** `Router(config-if)# ipv6 cga rsakeypair SEND`	(Optional) Configures CGA on interfaces. • In the example, CGA is generated.
Step 15	**ipv6 address** *ipv6-address* / *prefix-length* **link-local cga** **Example:** `Router(config-if)# ipv6 address fe80::link-local cga`	Configures an IPv6 link-local address for the interface and enables IPv6 processing on the interface.
Step 16	**ipv6 nd secured trustanchor** *trustpoint-name* **Example:** `Router(config-if)# ipv6 nd secured trustanchor SEND`	(Optional) Configures trusted anchors to be preferred for certificate validation.
Step 17	**ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value*} **Example:** `Router(config-if)# ipv6 nd secured timestamp delta 300`	(Optional) Configures the timing parameters.
Step 18	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.
Step 19	**ipv6 nd secured full-secure** **Example:** `Router(config)# ipv6 nd secured full-secure`	(Optional) Configures general SeND parameters, such as secure mode and authorization method. • In the example, SeND security mode is enabled.

Implementing IPv6 SeND

Creating the RSA Key Pair and CGA Modifier for the Key Pair

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:**
4. **ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:** **Example:** `Router(config)# crypto key generate rsa label SeND`	Generates RSA key pairs.
Step 4	**ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**} **Example:** `Router(config)# ipv6 cga modifier rsakeypair SeND sec-level 1`	Generates the CGA modifier for a specified RSA key, which enables the key to be used by SeND.

Configuring Certificate Enrollment for a PKI

Certificate enrollment, which is the process of obtaining a certificate from a CA, occurs between the end host that requests the certificate and the CA. Each peer that participates in the PKI must enroll with a CA. In IPv6, you can autoenroll or manually enroll the device certificate.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **subject-name** *x.500-name*]
5. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [pem]
6. **serial-number** [**none**]
7. **auto-enroll** [*percent*] [**regenerate**
8. **password** *string*
9. **rsakeypair** *key-label key-size encryption-key-size*]]
10. **fingerprint** *ca-fingerprint*
11. **ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*}
12. **exit**
13. **crypto pki authenticate** *name*
14. **exit**
15. **copy** [**/ erase**] [**/ verify** | **/ noverify**] *source-url destination-url*
16. show crypto pki certificates
17. show crypto pki trustpoints [status | label [status]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint trustpoint1`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 4	**subject-name** *x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name name1`	Specifies the subject name in the certificate request.
Step 5	**enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [pem] **Example:** `Router(ca-trustpoint)# enrollment url http://name1.example.com`	Specifies the URL of the CA on which your router should send certificate requests.
Step 6	**serial-number** [**none**] **Example:** `Router(ca-trustpoint)# serial-number`	(Optional) Specifies the router serial number in the certificate request.
Step 7	**auto-enroll** [*percent*] [**regenerate** **Example:** `Router(ca-trustpoint)# auto-enroll`	(Optional) Enables autoenrollment, allowing you to automatically request a router certificate from the CA.
Step 8	**password** *string* **Example:** `Router(ca-trustpoint)# password password1`	(Optional) Specifies the revocation password for the certificate.
Step 9	**rsakeypair** *key-label key-size encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair SEND`	Specifies which key pair to associate with the certificate.
Step 10	**fingerprint** *ca-fingerprint* **Example:** `Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E`	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

	Command or Action	Purpose
Step 11	**ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*} **Example:** `Router(ca-trustpoint)# ip-extension unicast prefix 2001:100:1://48`	Add IP extensions (IPv6 prefixes or range) to verify prefix list the router is allowed to advertise.
Step 12	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 13	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate name1`	Retrieves and authenticates the CA certificate. • This command is optional if the CA certificate is already loaded into the configuration.
Step 14	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 15	**copy** [**/ erase**] [**/ verify** | **/ noverify**] *source-url destination-url* **Example:** `Router# copy system:running-config nvram:startup-config`	(Optional) Copies the running configuration to the NVRAM startup configuration.
Step 16	show crypto pki certificates **Example:** `Router# show crypto pki certificates`	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.
Step 17	show crypto pki trustpoints [status | label [status]] **Example:** `Router# show crypto pki trustpoints name1`	(Optional) Displays the trustpoints configured in the router.

Configuring a Cryptographically Generated Address

Configuring General CGA Parameters

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd secured sec-level** [**minimum** *value*]
4. **ipv6 nd secured key-length** [[**minimum** | **maximum**] *v alue*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd secured sec-level** [**minimum** *value*] **Example:** `Router(config)# ipv6 nd secured sec-level minimum 1`	Configures the SeND security level.
Step 4	**ipv6 nd secured key-length** [[**minimum**	**maximum**] *v alue* **Example:** `Router(config)# ipv6 nd secured key-length minimum 512`	Configures SeND key-length options.

Configuring CGA Address Generation on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 cga rsakeypair** *key-label*
5. **ipv6 address** {*ipv6-address* / *prefix-length* [**cga**] | *prefix-name sub-bits*/*prefix-length*[**cga**]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 cga rsakeypair** *key-label* **Example:** `Router(config-if)# ipv6 cga rsakeypair SEND`	Specifies which RSA key pair should be used on a specified interface.
Step 5	**ipv6 address** {*ipv6-address* / *prefix-length* [**cga**] | *prefix-name sub-bits*/*prefix-length*[**cga**]} **Example:** `Router(config-if)# ipv6 address 2001:0DB8:1:1::/64 cga`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface. • The **cga** keyword generates a CGA address. **Note** The CGA link-local addresses must be configured by using the **ipv6 address link-local**command.

Configuring SeND Parameters

- Configuring the SeND Trustpoint, page 34
- Configuring SeND Trust Anchors on the Interface, page 37

- Configuring Secured and Nonsecured Neighbor Discovery Message Coexistence Mode, page 38
- Configuring SeND Parameters Globally, page 39
- Configuring the SeND Time Stamp, page 40

Configuring the SeND Trustpoint

In router mode, the key pair used to generate the CGA addresses on an interface must be certified by the CA and the certificate sent on demand over the SeND protocol. One RSA key pair and associated certificate is enough for SeND to operate; however, users may use several keys, identified by different labels. SeND and CGA refer to a key directly by label or indirectly by trustpoint.

Multiple steps are required to bind SeND to a trustpoint. First, a key pair is generated. Then the device refers to it in a trustpoint, and next the SeND interface configuration points to the trustpoint. There are two reasons for the multiple steps:

- The same key pair can be used on several SeND interfaces
- The trustpoint contains additional information, such as the certificate, required for SeND to perform authorization delegation

A CA certificate must be uploaded for the referred trustpoint. The referred trustpoint is in reality a trusted anchor.

Several trustpoints can be configured, pointing to the same RSA keys, on a given interface. This function is useful if different hosts have different trusted anchors (that is, CAs that they trust). The router can the provide each host with the certificate signed by the CA they trust.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:**
4. **ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}
5. **crypto pki trustpoint** *name*
6. **subject-name** [*x.500-name*]
7. **rsakeypair** *key-label key-size encryption-key-size*]]
8. **enrollment terminal** [**pem**
9. **ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*}
10. **exit**
11. **crypto pki authenticate** *name*
12. **crypto pki enroll** *name*
13. **crypto pki import** *name* **certificate**
14. **interface** *type number*
15. **ipv6 nd secured trustpoint** *trustpoint-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename* **:**] [**on** *devicename* **:** **Example:** `Router(config)# crypto key generate rsa label SEND`	Generates RSA key pairs.
Step 4	**ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**} **Example:** `Router(config)# ipv6 cga modifier rsakeypair SEND sec-level 1`	Generates the CGA modifier for a specified RSA key, which enables the key to be used by SeND.
Step 5	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint trustpoint1`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 6	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name name1`	Specifies the subject name in the certificate request.
Step 7	**rsakeypair** *key-label key-size encryption-key-size*]] **Example:** `Router(ca-trustpoint)# rsakeypair SEND`	Specifies which key pair to associate with the certificate.

	Command or Action	Purpose
Step 8	**enrollment terminal** [**pem** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.
Step 9	**ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress max-ipaddress*} **Example:** `Router(ca-trustpoint)# ip-extension unicast prefix 2001:100:1://48`	Adds IP extensions to the router certificate request.
Step 10	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 11	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate trustpoint1`	Authenticates the certification authority (by getting the certificate of the CA).
Step 12	**crypto pki enroll** *name* **Example:** `Router(config)# crypto pki enroll trustpoint1`	Obtains the certificates for your router from the CA.
Step 13	**crypto pki import** *name* **certificate** **Example:** `Router(config)# crypto pki import trustpoint1 certificate`	Imports a certificate manually via TFTP or as a cut-and-paste at the terminal.
Step 14	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.

Command or Action	Purpose
Step 15 **ipv6 nd secured trustpoint** *trustpoint-name* **Example:** `Router(config-if)# ipv6 nd secured trustpoint trustpoint1`	Enables SeND on an interface and specifies which trustpoint should be used.

Configuring SeND Trust Anchors on the Interface

This task can be performed only in host mode. The host must be configured with one or more trust anchors. As soon as SeND is bound to a trustpoint on an interface (see Configuring the SeND Trustpoint, page 34), this trustpoint is also a trust anchor.

A trust anchor configuration consists of the following items:

- A public key signature algorithm and associated public key, which may include parameters
- A name
- An optional public key identifier
- An optional list of address ranges for which the trust anchor is authorized

Because PKI has already been configured, the trust anchor configuration is accomplished by binding SeND to one or several PKI trustpoints. PKI is used to upload the corresponding certificates, which contain the required parameters (such as name and key).

Perform this optional task to configure a trusted anchor on the interface. It allows you to select trust anchors listed in the CPS when requesting for a certificate. If you opt not to configure trust anchors, all the PKI trustpoints configured on the host will be considered.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal** [**pem**
5. **exit**
6. **crypto pki authenticate** *name*
7. **interface** *type number*
8. **ipv6 nd secured trustanchor** *trustanchor-name*

DETAILED STEPS

Command or Action	Purpose
Step 1 **enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint anchor1`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 4	**enrollment terminal** [**pem** **Example:** `Router(ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.
Step 5	**exit** **Example:** `Router(ca-trustpoint)# exit`	Returns to global configuration.
Step 6	**crypto pki authenticate** *name* **Example:** `Router(config)# crypto pki authenticate anchor1`	Authenticates the certification authority (by getting the certificate of the CA).
Step 7	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	**ipv6 nd secured trustanchor** *trustanchor-name* **Example:** `Router(config-if)# ipv6 nd secured trustanchor anchor1`	Specifies a trusted anchor on an interface and binds SeND to a trustpoint.

Configuring Secured and Nonsecured Neighbor Discovery Message Coexistence Mode

During the transition to SeND secured interfaces, network operators may want to run a particular interface with a mixture of nodes accepting secured and unsecured neighbor discovery messages. Perform this task to configure the coexistence mode for secure and nonsecure ND messages on the same interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd secured trustpoint** *trustpoint-name*
5. **no ipv6 nd secured full-secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 nd secured trustpoint** *trustpoint-name* **Example:** `Router(config-if)# ipv6 nd secured trustpoint trustpoint1`	Enables SeND on an interface and specifies which trustpoint should be used.
Step 5	**no ipv6 nd secured full-secure** **Example:** `Router(config-if)# no ipv6 nd secured full-secure`	Provides the coexistence mode for secure and nonsecure ND messages on the same interface.

Configuring SeND Parameters Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd secured key-length** [[**minimum**| **maximum**] *value*]
4. **ipv6 nd secured sec-level minimum** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd secured key-length** [[**minimum**| **maximum**] *value*] **Example:** `Router(config)# ipv6 nd secured key-length minimum 512`	Configures the SeND key-length options.
Step 4	**ipv6 nd secured sec-level minimum** *value* **Example:** `Router(config)# ipv6 nd secured sec-level minimum 2`	Configures the minimum security level value that can be accepted from peers.

Configuring the SeND Time Stamp

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**ipv6 nd secured timestamp** {**delta** *value* | **fuzz** *value* **Example:** `Router(config-if)# ipv6 nd secured timestamp delta 600`	Configures the SeND time stamp.

Configuring IPv6 PACL

Creating an IPv6 Access List

The first task in configuring IPv6 PACL is to create an IPv6 access list. This task is described in detail in Implementing Traffic Filters and Firewalls for IPv6 Security.

Configuring PACL Mode and Applying IPv6 PACL on an Interface

Once you have configured the IPv6 access list you want to use, you must configure the PACL mode on the specified IPv6 L2 interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **access-group mode** {**prefer** {**port** | **vlan**} | **merge**}
5. **ipv6 traffic-filter** *access-list-name* {**in** | **out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/0`	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	**access-group mode** {**prefer** {**port**	**vlan**}	**merge**} **Example:** `Router(config-if)# access-group mode prefer port`	Sets the mode for the specified layer 2 interface. • The **no** form of this command sets the mode to the default value, which is merge. • The **prefer vlan** keyword combination is not supported in IPv6.
Step 5	**ipv6 traffic-filter** *access-list-name* {**in**	**out** **Example:** `Router(config-if)# ipv6 traffic-filter list1 in`	Filters incoming IPv6 traffic on an interface. **Note** The **out** keyword and therefore filtering of outgoing traffic is not supported in IPv6 PACL configuration.

Configuration Examples for Implementing First Hop Security in IPv6

Example IPv6 ND Inspection and RA Guard Configuration

This example provides information about the Ethernet 0/0 interface, on which the ND inspection and RA Guard features are configured:

```
Router# show ipv6 snooping capture interface ethernet 0/0
Hardware policy registered on Et0/0
Protocol     Protocol value   Message   Value     Action    Feature
ICMP         58               RS        85        punt      RA Guard
                                                   punt      ND Inspection
ICMP         58               RA        86        drop      RA guard
                                                   punt      ND Inspection
ICMP         58               NS        87        punt      ND Inspection
ICM          58               NA        88        punt      ND Inspection
ICMP         58               REDIR     89        drop      RA Guard
                                                   punt      ND Inspection
```

Example RA Guard Configuration

This section provides a configuration example for the RA guard feature:

```
Router(config)# interface fastethernet 3/13

Router(config-if)# ipv6 nd raguard

Router# show run interface fastethernet 3/13
Building configuration...
Current configuration : 129 bytes
!
interface FastEthernet3/13
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

Example Configuring PACL Mode and Applying IPv6 PACL on an Interface

Once you have configured the IPv6 access list you want to use, you can configure the PACL mode on a specified IPv6 switchport. This section uses an access list named list1, provides an example of how to configure PACL mode, and applies IPv6 PACL to a GigabitEthernet interface.

```
Router(config)# interface gigabitethernet 3/24
Router(config-if)# access-group mode prefer port
Router(config-if)# ipv6 traffic-filter list1 in
```

Example SeND Configuration Examples

Example Configuring Certificate Servers

The following example shows how to configure certificate servers:

```
crypto pki server CA
 issuer-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=CA0  lifetime ca-certificate
700 !
crypto pki trustpoint CA
 ip-extension prefix 2001::/16
 revocation-check crl
 rsakeypair CA
no shutdown
```

Note If you need to configure certificate servers without IP extensions, do not use the **ip-extension**command.

To display the certificate servers with IP extensions, use the **show crypto pki certificates verbose** command:

```
Router# show crypto pki certificates verbose
CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    c=FR
    st=fr
    l=example
    o=cisco
    ou=nsstg
    cn=CA0
  Subject:
    c=FR
    st=fr
    l=example
    o=cisco
    ou=nsstg
    cn=CA0
  Validity Date:
    start date: 09:50:52 GMT Feb 5 2009
    end   date: 09:50:52 GMT Jan 6 2011
  Subject Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
  Signature Algorithm: MD5 with RSA Encryption
  Fingerprint MD5: 87DB764F 29367A65 D05CEE3D C12E0AC3
  Fingerprint SHA1: 04A06602 86AA72E9 43F2DB33 4A7D40A2 E2ED3325
  X509v3 extensions:
    X509v3 Key Usage: 86000000
      Digital Signature
      Key Cert Sign
```

```
      CRL Signature
    X509v3 Subject Key ID: 75B477C6 B2CA7BBE C7866657 57C84A32 90CEFB5A
    X509v3 Basic Constraints:
        CA: TRUE
    X509v3 Authority Key ID: 75B477C6 B2CA7BBE C7866657 57C84A32 90CEFB5A
    Authority Info Access:
    X509v3 IP Extension:
        IPv6:
          2001::/16
  Associated Trustpoints: CA
```

Example Configuring a Host to Enable SeND

The following example shows how to configure a host to enable SeND:

```
enable
configure terminal
 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
 crypto pki trustpoint SEND
  enrollment url http://209.165.200.254
  revocation-check none
  exit
 crypto pki authenticate SEND
Certificate has the following attributes:
    Fingerprint MD5: FC99580D 0A280EB4 2EB9E72B 941E9BDA
    Fingerprint SHA1: 22B10EF0 9A519177 145EA4F6 73667837 3A154C53
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
 ipv6 nd secured sec-level minimum 1
 interface fastethernet 0/0
  ipv6 cga rsakeypair SEND
  ipv6 address FE80::260:3EFF:FE11:6770 link-local cga
  ipv6 nd secured trustanchor SEND
  ipv6 nd secured timestamp delta 300
  exit
 ipv6 nd secured full-secure
```

To verify the configuration use the **show running-config** command:

```
host# show running-config
Building configuration...
[snip]
crypto pki trustpoint SEND
 enrollment url http://209.165.200.225
 revocation-check none
!
interface Ethernet1/0
 ip address 209.165.202.129 255.255.255.0
 duplex half
 ipv6 cga rsakeypair SEND
 ipv6 address 2001:100::/64 cga
```

Example Configuring a Router to Enable SeND

The following example shows how to configure the router to enable SeND:

```
enable
configure terminal
 crypto key generate rsa label SEND modulus 1024
 ipv6 cga modifier rsakeypair SEND sec-level 1
 crypto pki trustpoint SEND
  subject-name C=FR, ST=PACA, L=Example, O=Cisco, OU=NSSTG, CN=router
  rsakeypair SEND
  revocation-check none
```

```
  exit
 crypto pki authenticate SEND
Certificate has the following attributes:
    Fingerprint MD5: FC99580D 0A280EB4 2EB9E72B 941E9BDA
    Fingerprint SHA1: 22B10EF0 9A519177 145EA4F6 73667837 3A154C53
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
 crypto pki enroll SEND
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: C=FR, ST=fr, L=example, O=cisco,
OU=nsstg, CN=route r % The subject name in the certificate will include: Router % Include
the router serial number in the subject name? [yes/no]: no % Include an IP address in the
subject name? [no]:
Request certificate from CA? [yes/no]: yes % Certificate request sent to Certificate
Authority % The 'show crypto pki certificate SEND verbose' commandwill show the
fingerprint.
*Feb  5 09:40:37.171: CRYPTO_PKI:  Certificate Request Fingerprint MD5:
A6892F9F 23561949 4CE96BB8 CBC85 E64
*Feb  5 09:40:37.175: CRYPTO_PKI:  Certificate Request Fingerprint SHA1:
30832A66 E6EB37DF E578911D 383F 96A0 B30152E7
*Feb  5 09:40:39.843: %PKI-6-CERTRET: Certificate received from Certificate Authority
 interface fastethernet 0/0
  ipv6 nd secured sec-level minimum 1
  ipv6 cga rsakeypair SEND
  ipv6 address fe80::link-local cga
  ipv6 nd secured trustanchor SEND
  ipv6 nd secured timestamp delta 300
  exit
 ipv6 nd secured full-secure
```

To verify that the certificates are generated, use the **show crypto pki certificates** command:

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 0x15
  Certificate Usage: General Purpose
  Issuer:
    cn=CA
  Subject:
    Name: Router
    hostname=Router
    c=FR
    st=fr
    l=example
    o=cisco
    ou=nsstg
    cn=router
  Validity Date:
    start date: 09:40:38 UTC Feb 5 2009
    end   date: 09:40:38 UTC Feb 5 2010
  Associated Trustpoints: SEND
CA Certificate
  Status: Available
  Certificate Serial Number: 0x1
  Certificate Usage: Signature
  Issuer:
    cn=CA
  Subject:
    cn=CA
  Validity Date:
    start date: 10:54:53 UTC Jun 20 2008
    end   date: 10:54:53 UTC Jun 20 2011
  Associated Trustpoints: SEND
```

To verify the configuration, use the **show running-config** command:

```
Router# show running-config
```

```
Building configuration...
[snip]
crypto pki trustpoint SEND
 enrollment url http://209.165.201.1
 subject-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=router  revocation-check
none  rsakeypair SEND !
interface Ethernet1/0
 ip address 209.165.200.225 255.255.255.0
 duplex half
 ipv6 cga rsakeypair SEND
 ipv6 address FE80:: link-local cga
 ipv6 address 2001:100::/64 cga
```

Example Configuring a SeND Trustpoint in Router Mode

The following example shows how to configure a SeND trustpoint in router mode:

```
enable
 configure terminal
 crypto key generate rsa label SEND
  Choose the size of the key modulus in the range of 360 to 2048 for your
 General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.
How many bits in the modulus [512]: 778
% Generating 778 bit RSA keys, keys will be non-exportable...[OK]
 ipv6 cga modifier rsakeypair SEND sec-level 1
 crypto pki trustpoint trstpt1
  subject-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=sa14-72b
  rsakeypair SEND
  enrollment terminal
  ip-extension unicast prefix 2001:100:1://48
  exit
 crypto pki authenticate trstpt1
 crypto pki enroll trstpt1
 crypto pki import trstpt1 certificate
 interface Ethernet 0/0
  ipv6 nd secured trustpoint trstpt1
```

Example Configuring SeND Trust Anchors in the Host Mode

The following example shows how to configure SeND trust anchors on an interface in the host mode:

```
enable
configure terminal
! Configure the location of the CS we trust !
 crypto pki trustpoint B1
  enrollment terminal
  crypto pki authenticate anchor1
  exit
! Only Query a certificate signed by the CS at B2 on this interface !
 interface Ethernet0/0
  ip address 204.209.1.54 255.255.255.0
  ipv6 cga rsakeypair SEND
  ipv6 address 2001:100::/64 cga
  ipv6 nd secured trustanchor anchor1
```

Example Configuring CGA Address Generation on an Interface

The following example shows how to configure CGA address generation on an interface:

```
enable
 configure terminal
 interface fastEthernet 0/0
  ipv6 cga rsakeypair SEND
  ipv6 address 2001:100::/64 cga
  exit
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 Neighbor Discovery	Implementing IPv6 Addressing and Basic Connectivity
ICMP in IPv6	Implementing IPv6 Addressing and Basic Connectivity
IPv6--IPv6 stateless autoconfiguration	Implementing IPv6 Addressing and Basic Connectivity
IPv6 access lists	Implementing Traffic Filters and Firewalls for IPv6 Security
IPv6 DHCP	Implementing DHCP for IPv6
Configuring certificate enrollment for a PKI	"Configuring Certificate Enrollment for a PKI" module in the *Cisco IOS Security Configuration Guide*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
All Cisco IOS commands	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3756	*IPv6 Neighbor Discovery (ND) Trust Models and Threats*
RFC 3779	X.509 Extensions for IP Addresses and AS Identifiers
RFC 3971	*Secure Neighbor Discovery (SeND)*
RFC 3972	*Cryptographically Generated Addresses (CGA)*
RFC 6105	*IPv6 Router Advertisement Guard*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Implementing First Hop Security in IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Implementing First Hop Security in IPv6

Feature Name	Releases	Feature Information
IPv6 Device Tracking	12.2(50)SY	This feature allows IPv6 host liveness to be tracked so the neighbor binding table can be immediately updated when an IPv6 host disappears. The following commands were introduced or modified: **ipv6 neighbor binding**, **ipv6 neighbor binding down-lifetime**, **ipv6 neighbor binding logging**, **ipv6 neighbor binding max-entries**, **ipv6 neighbor binding stale-lifetime**, **ipv6 neighbor binding vlan**, **ipv6 neighbor tracking**, **show ipv6 neighbor binding**.
IPv6 ND Inspection	12.2(50)SY	The IPv6 ND Inspection feature learns and secures bindings for stateless autoconfiguration addresses in layer 2 neighbor tables. The following commands were introduced: clear ipv6 snooping counters, debug ipv6 snooping, **device-role**, **drop-unsecure**, **ipv6 nd inspection**, **ipv6 nd inspection policy**, **sec-level minimum**, **show ipv6 snooping capture-policy**, s**how ipv6 snooping counters**,s**how ipv6 snooping features**, **show ipv6 snooping policies**, **tracking trusted-port**.
IPv6 PACL	12.2(54)SG 12.2(33)SXI4 12.2(50)SY	The IPv6 PACL permits or denies the movement of traffic between Layer 3 (L3) subnets and VLANs, or within a VLAN. The following commands were introduced or modified: access-group mode, ipv6 traffic-filter.

Feature Name	Releases	Feature Information
IPv6 RA Guard	12.2(54)SG 12.2(33)SXI4 12.2(50)SY	IPv6 RA guard provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network switch platform.

Feature Name	Releases	Feature Information
Secure Neighbor Discovery for Cisco IOS Software	12.4(24)T	The Secure Neighbor Discovery (SeND) protocol is designed to counter the threats of the ND protocol. SeND defines a set of neighbor discovery options and two neighbor discovery messages. SeND also defines a new autoconfiguration mechanism to establish address ownership. The following commands were introduced or modified: **auto-enroll**, **crypto key generate rsa**, **crypto pki authenticate**, **crypto pki enroll**, **crypto pki import**, **enrollment terminal (ca-trustpoint)**, **enrollment url (ca-trustpoint)**, **fingerprint**, **ip-extension**, **ip http server**, **ipv6 address**, **ipv6 address link-local**, **ipv6 cga modifier rsakeypair**, **ipv6 cga modifier rsakeypair (interface)**, **ipv6 nd secured certificate-db**, **ipv6 nd secured full-secure**, **ipv6 nd secured full-secure (interface)**, **ipv6 nd secured key-length**, **ipv6 nd secured sec-level**, **ipv6 nd secured timestamp**, **ipv6 nd secured timestamp-db**, **ipv6 nd secured trustanchor**, **ipv6 nd secured trustpoint**, **password (ca-trustpoint)**, **revocation-check**, **rsakeypair**, **serial-number (ca-trustpoint)**, **show ipv6 cga address-db**, **show ipv6 cga modifier-db**, **show ipv6 nd secured certificates**, **show ipv6 nd secured counters interface**, **show ipv6 nd secured nonce-db**, **show ipv6 nd secured timestamp-db**, **subject-name**.

Glossary

- **ACE** --access control entry
- **ACL** --access control list

- **CA** --certification authority.
- **CGA** --cryptographically generated address.
- **CPA** --certificate path answer.
- **CPR** --certificate path response.
- **CPS** --certification path solicitation. The solicitation message used in the addressing process.
- **CRL** --certificate revocation list.
- **CS** --certification server.
- **CSR** --certificate signing request.
- **DAD** --duplicate address detection. A mechanism that ensures two IPv6 nodes on the same link are not using the same address.
- **DER** --distinguished encoding rules. An encoding scheme for data values.
- **LLA** --link-layer address.
- **MAC** --media access control.
- **nonce** --An unpredictable random or pseudorandom number generated by a node and used once. In SeND, nonces are used to assure that a particular advertisement is linked to the solicitation that triggered it.
- **non-SeND node** --An IPv6 node that does not implement SeND but uses only the neighbor discovery protocol without security.
- **NUD** --neighbor unreachability detection. A mechanism used for tracking neighbor reachability.
- **PACL** --port-based access list.
- **PKI** --public key infrastructure.
- **RA** --router advertisement.
- **Router Authorization Certificate** --A public key certificate.
- **RD** --Router discovery allows the hosts to discover what routers exist on the link and what subnet prefixes are available. Router discovery is a part of the neighbor discovery protocol.
- **SeND node** --An IPv6 node that implements SeND.
- **trust anchor** --A trust anchor is an entity that the host trusts to authorize routers to act as routers. Hosts are configured with a set of trust anchors to protect router discovery.
- **ULA** --unique local addressing.