White paper
Cisco public

Cisco Catalyst 9000 Family Switch Integrated Security Features (SISF)

Contents

Introduction

The Layer 2 domain continues to be of primary importance for enterprise networks. This evolution has been creating a number of challenges, such as issues involving security and scaling. At the same time, IPv6 has been gaining momentum as the next-generation Internet Protocol, maturing over time while the IPv4 address space is nearly exhausted. One of the areas that has been heavily reengineered during the course of the IPv6 specification is IPv6 link operations, which encompass all operations occurring between end nodes in a given Layer 2 domain. Security, address assignment, address resolution, neighbor discovery, exit point discovery, etc. have been reworked and improved.

Layer 2 (and to some extent Layer 2/3) switches offer many opportunities to secure the paths between the end nodes, as well as the paths to other Layer 2/3 domains, and to optimize link operations. These switches are sometimes referred to as “first hops,” particularly when they are facing end nodes. For a long time now, Cisco has provided a set of features that run on the switches to secure and optimize Layer 2 operations with IPv4. Now that IPv6 is taking off, a similar set of capabilities is necessary.

To support large-scale Layer 2 domains with enhanced security, a number of features are necessary to address the requirements listed above. Cisco has architected an infrastructure (SISF, or Switch Integrated Security Features) built around a keystone known as the binding table.

The binding table contains information about the host’s or hosts’ IP and MAC addresses that is connected behind every switch port. This creates a physical map of the hosts that are connected. The data from the map is used to populate the source IP of dynamic Access Control Lists (ACLs), and to maintain a binding of an IP address to a security group tag. This data is also useful for inventory purposes on switches that aren’t performing routing (and so don’t have Address Resolution Protocol [ARP] tables naturally).

The binding table is updated when devices are detected or removed. Device tracking serves only as a collection point for device information and depends on probes to track the connectivity of the device. Hence device tracking is used as a common component for multiple security features.

Device tracking applies similar approaches to track IPv4 and IPv6 but using different underlying protocols.

Figure 1 shows an IPv4 and IPv6 device tracking table.

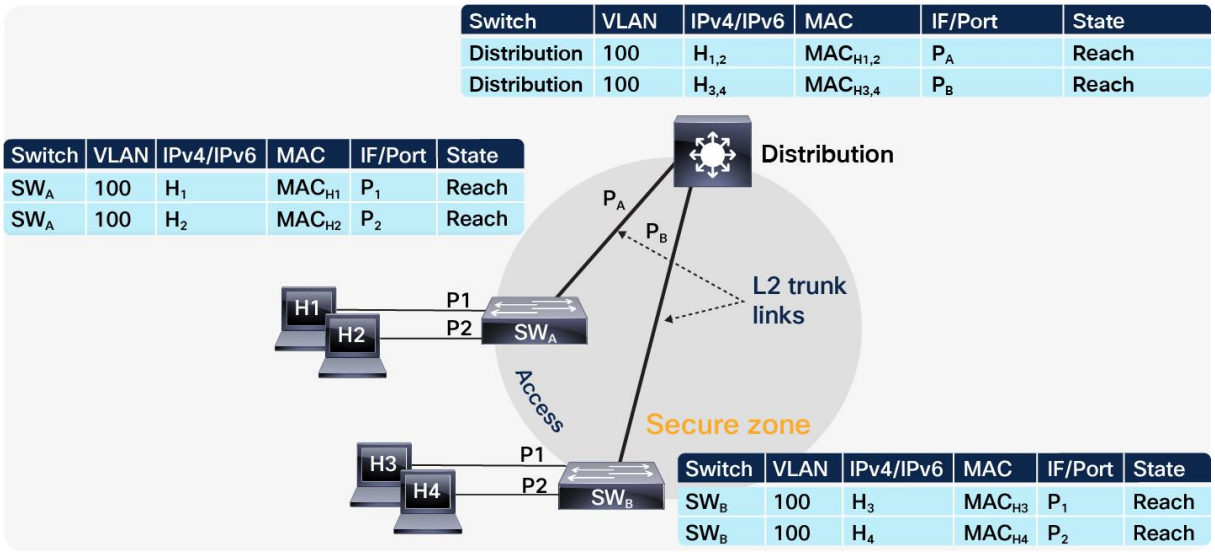

Figure 1.
IP4 and IP6 device tracking

SISF architecture

SISF provides a binding table, and there are feature clients that use the information from it. The entries in the table are populated by gleaning packets such as Dynamic Host Configuration Protocol (DHCP), ARP, and Neighbor Discovery (ND) that are tracking the host activity and help to dynamically populate the table. If there are silent hosts in the Layer 2 domain, static entries can be used to add to the SISF entries.

Figure 2 shows the SISF architecture.

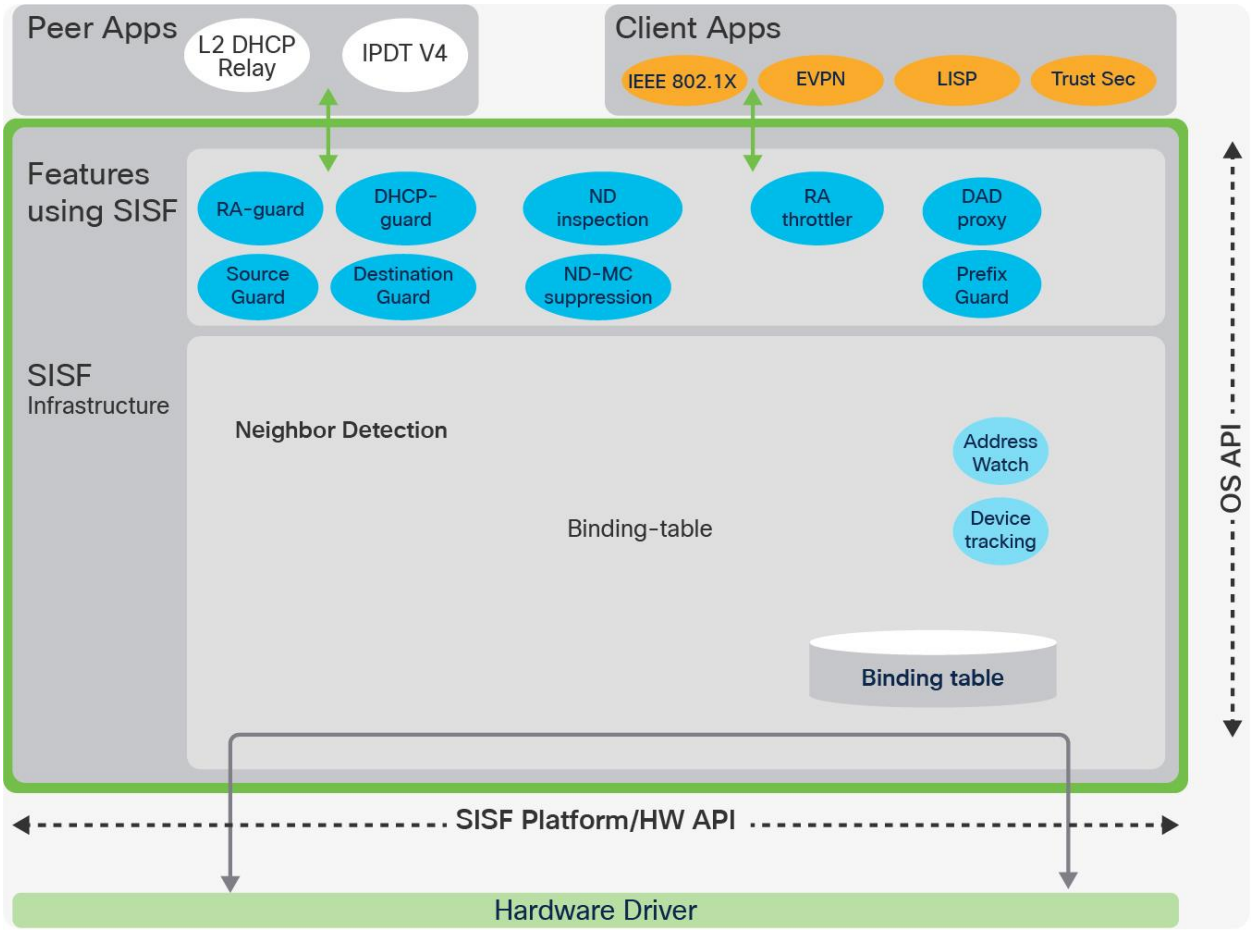

Figure 2.
SISF architecture

This architecture offers a flexible and scalable way for the binding table to be shared and reused by many feature clients.

SISF uses a policy model to configure device roles and additional settings on the switch. A single policy can be applied to the interface or at the VLAN level. If a policy is applied to a VLAN and a different policy is applied to an interface, the interface policy will take precedence over the VLAN policy. Some parameters between the policies might merge.

SISF can limit the number of hosts learned per port. When the limit is reached:

- For IPv4 no more new entries can be added to the table.
- For IPv6 no more new entries can be added to the table, and traffic from the new hosts will be dropped.

Figure 3 shows an example of SISF policy and how to read it.

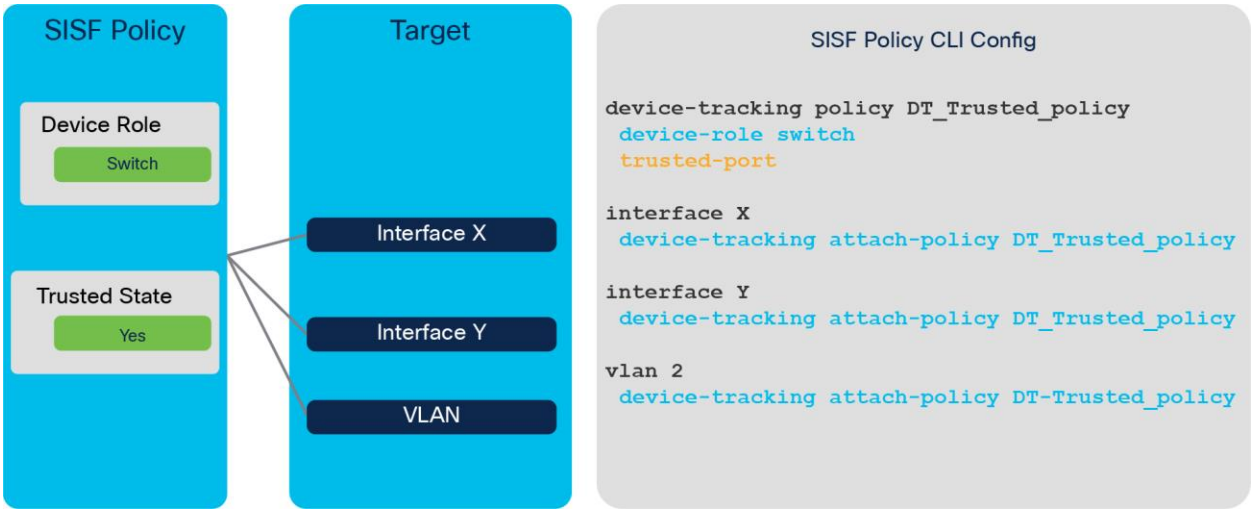

Figure 3.
SISF policy

There are two ways to enable SISF policies:

- Programmatically enabled by other features
- Manually configured by the Command-Line Interface (CLI), which will override any programmatic policy

Inside the policy, the customer has the option to select based on the address of the packets to be snooped populated into the SISF database.

```
Switch(config-device-tracking)#?
device-tracking policy configuration mode:
  data-glean            binding recovery by data traffic source address gleaning
  destination-glean     binding recovery by data traffic destination address gleaning
  device-role           Sets the role of the device attached to the port
  limit                 Specifies a limit
  prefix-glean          Glean prefixes in RA and DHCP-PD traffic
  protocol              Sets the protocol to glean (default all)
  security-level        setup security level
  tracking              Override default tracking behavior
  trusted-port          setup trusted port
```

In Cisco IOS® XE Release 16.9(x), SISF introduced policy priority. It adds options to control the updates into SISF, and if two or more clients are using the binding table, updates from the higher-priority feature will be applied. (We discuss this in more detail later.)

Finally, there is a common misunderstanding regarding ping and device-tracking entry. Many of us think ping will always create a device-tracking entry. Actually, it does not. First of all, SISF snoops only control packets, such as ARP and ND. It don't snoop Internet Control Message Protocol (ICMP) echo request/reply packets, which ping sends.

Sometimes ping will trigger an ARP (for IPv4) or ND (for IPv6) packet if the sender's ARP cache or IPv6 neighbor table doesn't have the target's IP address yet. That is when ping can result in a device-tracking entry. When the target IP is already in the ARP cache or IPv6 neighbor table, no ARP or ND packet is generated when you ping, and therefore SISF cannot learn the IP address.

Open Cisco IOS XE 16.X also introduced a distributed SISF binding table. The main benefits of the distributed table are that the distribution switch does not need to run SISF and have one huge table without affecting the secure zones.

Figure 4 shows the initial stage before SISF security features are enabled.

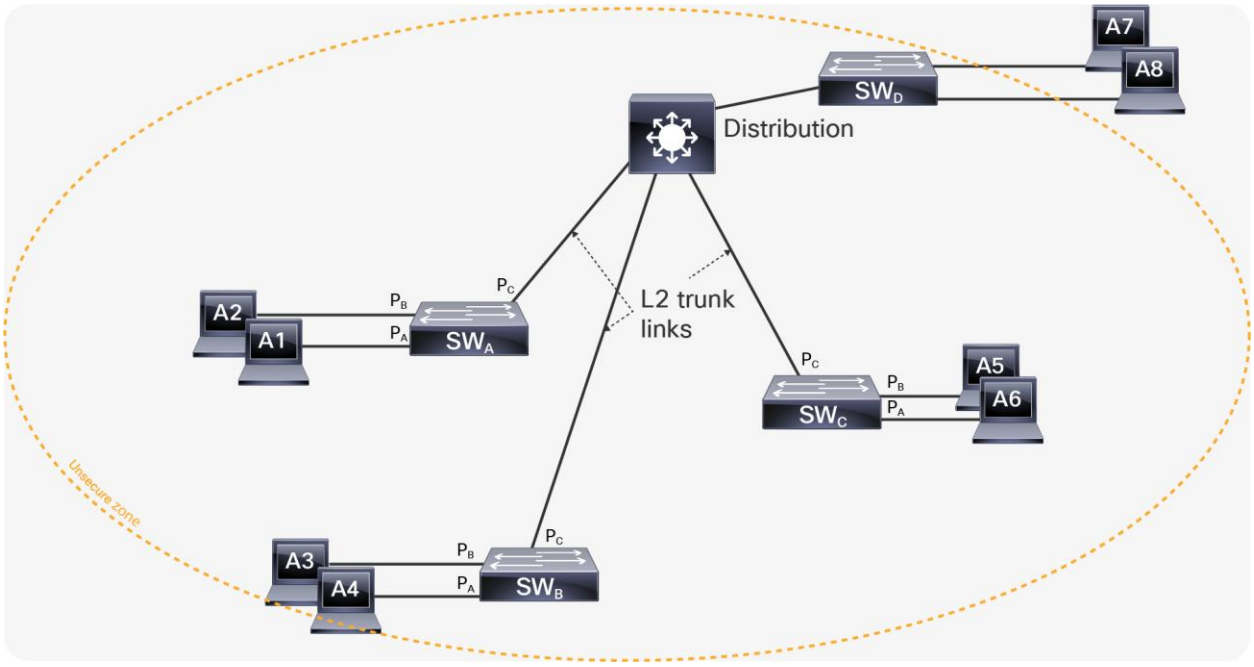

Figure 4.
Initial stage

Once SISF security features are enabled, they are isolated per switch. Figure 5 shows an isolated secure zone, with every access switch having a large SISF binding table, which is quite inefficient and consumes more memory and CPU resources.

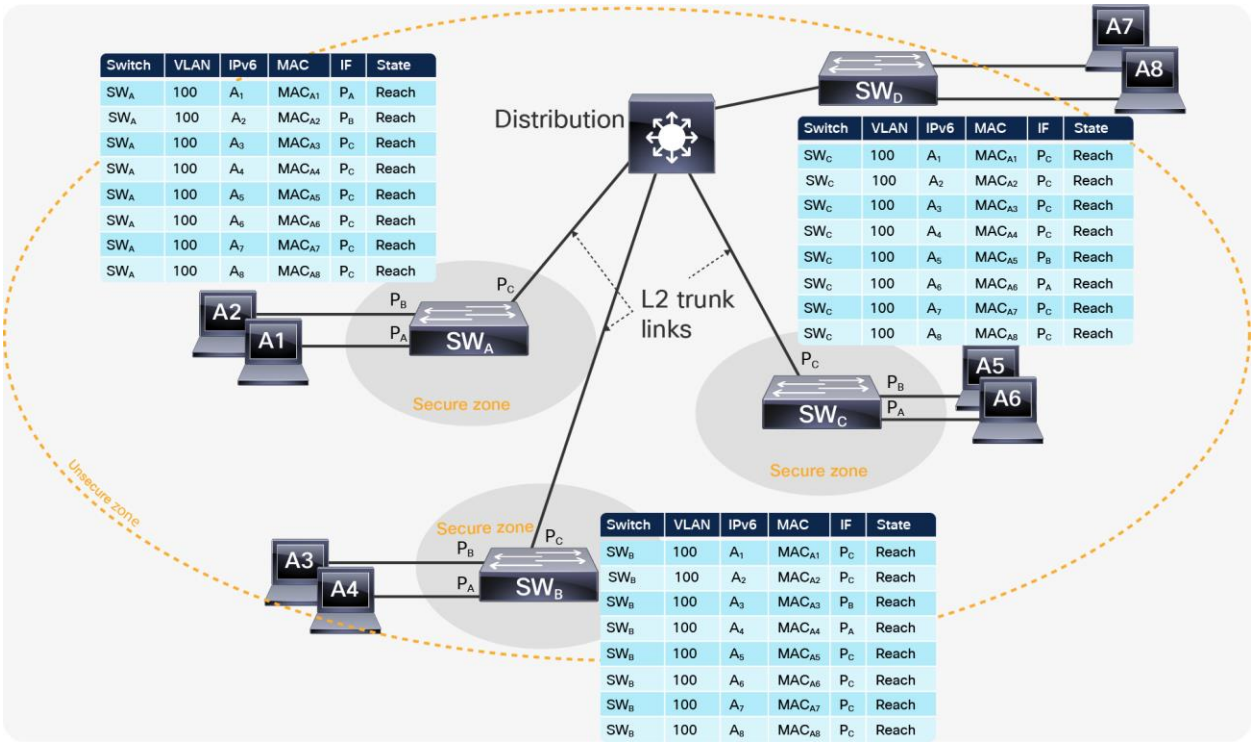

Figure 5.
Isolated secure zone

With the distributed SISF table, we can see in Figure 6 that every access switch has a very small SISF table, and the distribution switch is now part of the secure zone. That creates a very effective and optimized SISF binding table on every switch. To enable the feature, configure a “trust SISF” policy on inter-switch links.

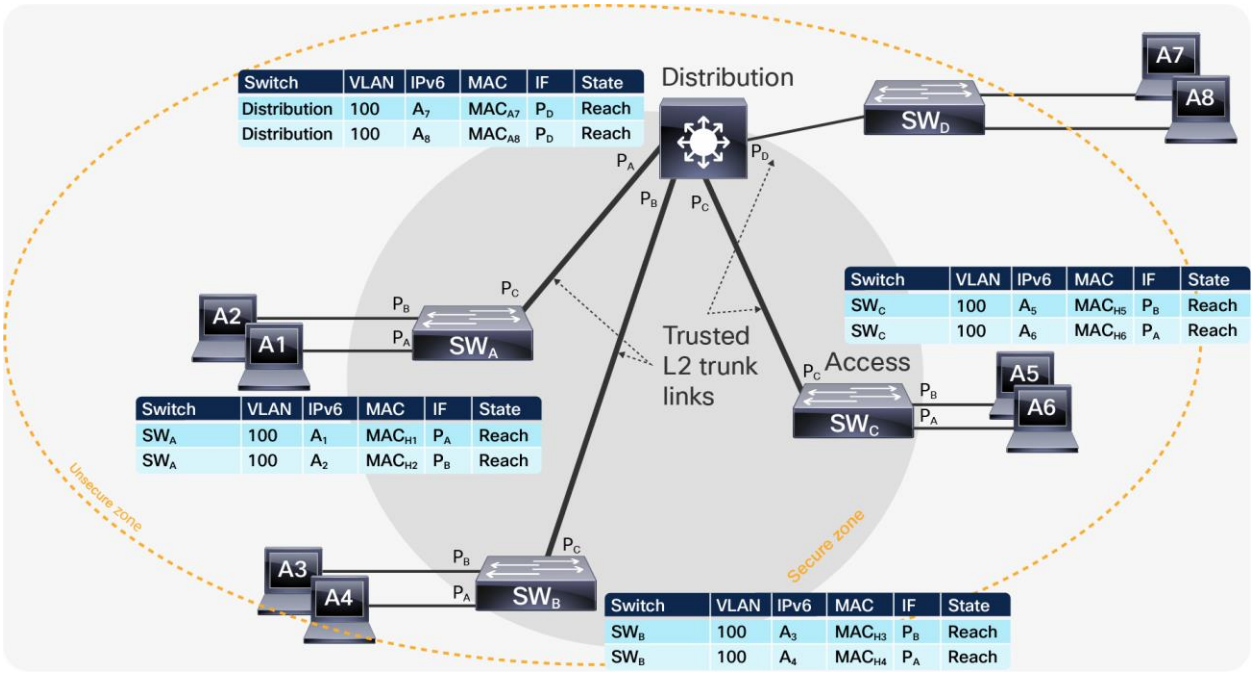

Figure 6.
Distributed SISF table

SISF use cases

This section discusses how SISF is used to increase network security. There are many use cases for SISF; here we list a few examples of how to use it to allow the IP and MA addresses so that their entry port is well known and authorized.

The **first** use case is **protection against default gateway theft**. In this scenario the attacker is trying to steal the default router IP and MAC and redirect all traffic through an intermediate device. The attacker tries to intercept the ARP/ND discovery messages for the gateway and replay them with false alternate information. Other approaches are to send a rogue Router Advertisement (RA), steal the router identity, or send gratuitous ARP messages.

To mitigate this problem, the user can configure RA guard and IPv6 snooping, which use SISF.

The **second** use case is **protection against address theft**. In this scenario, the attacker is into the same Layer 2 domain and listens for Stateless Address Aauto-Configuration (SLAAC) and DHCP and sends unsolicited neighbor advertisement messages with the same IP as the target IP and its own MAC address to redirect the traffic.

To mitigate this problem, the user can configure address watch (IPv6 snooping), which uses SISF. IPv6 snooping will:

- Glean addresses in Neighbor Discovery Protocol (NDP) and DHCP
- Log bindings <address, port, MAC, VLAN> for traceability
- Establish and enforce rules for address ownership
- Prevent address theft
- Limit the number of bindings accepted per user (define “user”)

The **third** use case is **protection against address spoofing**. This scenario involves the types of attacks listed in Table 1.

Table 1. Types of address spoofing attacks

Blind attacks	Non-blind attacks
• Single packet attacks • Flood-based denial of service (DoS) • Poisoning attack • Spoof-based worm/malware propagation • Reflective attacks • Accounting subversion	• Man-in-the-middle attacks • Third-party recon

To mitigate this problem, the user can configure port-based address binding (IP Source Guard), which uses SISF.

The **fourth** use case is **protection against a DoS attack on the neighbor cache**. In this scenario, the attacker is trying to exhaust the resources of the SISF table, cause confusion, and drop malicious changes into the SISF table.

To mitigate this problem, the user can configure IPv6 snooping, which uses SISF. IPv6 snooping will accept only known and authorized changes.

SISF behavior evolution

This section discusses how SISF capabilities have changed in different releases to help you design and use SISF.

16.6.x/16.8.x releases

Table 2 discusses the features that use SISF in Releases 16.6 and 16.8.

Table 2. Release 16.6 and 16.8 features that depend on SISF

Features that depends on SISF	Usage
802.1X / MAC Authentication Bypass (MAB) WebAuth Cisco Trustsec® IP Source Guard	Description: These features use SISF to create a table separate from the MAC address table. The SISF entries are used for verification and are changed based on authorization. SISF is enabled programmatically. Mandatory configuration: CLI “ip dhcp snooping vlan” is required for these features to work. SISF will be enabled, but DHCP snooping is not enabled, as the global DHCP snooping CLI is disabled. Optional configuration: Use a global configuration to change the limit for address count, stale-lifetime or down-lifetime.

LISP on VLAN	Description: LISP uses SISF to discover IPv6 entries and Layer 3 to Layer 2 bindings (IPv4 or IPv6/MAC).
	SISF is enabled programmatically.
	Mandatory LISP configuration to enable SISF:
	`router lisp`
	`instance-id 1`
	`service ethernet`
	`eid-table vlan 10`
	`database-mapping mac locator-set`

LISP on VLAN with DHCP snooping	Description: “LISP on VLAN” will take precedence in the settings to which it applies. The difference between LISP and DHCP snooping will be the binding time only; hence, both features will work.
	`Note: (LISP settings will be applied)`
	`Binding entry down timer: 10 minutes (*)`
	`Binding entry stale timer: 30 minutes (*)`
	`Note: (DHCP snooping will not be applied)`
	`Binding entry down timer: 24 hours (*)`
	`Binding entry stale timer: 24 hours (*)`
	SISF is enabled programmatically.

16.9 and newer releases

If a Cisco® Catalyst® 9000 family switch configuration is migrated from Release 16.6 or 16.8, the features will continue to operate, but there will be behavioral changes.

The first change is SISF policy priority. The priority cannot be changed and is automatically added by SISF. It will help to apply the merged settings between the features that use SISF: LISP/EVPN, 802.1X, WebAuth, Cisco Trustsec, DHCP snooping. In previous releases only one feature could be applied on a target (interface or VLAN), but adding a priority enables the switch to apply more than one SISF feature.

The second change is optimization for Cisco Software-Defined Access (SD-Access). SISF assists SD-Access deployments in reducing ARP/ND broadcasts by intercepting the messages/packets and converting them to unicast if the destination MAC address is known. This capability is enabled programmatically.

The third change is that ARP probes are sent by the switch as unicast in fabric environment. In earlier releases, ARP probes were sent as broadcasts, and if there were hundreds of clients in one large VLAN, that would cause an unnecessary flood of SISF probes.

The fourth change is the added ability to achieve distributed device-tracking tables in multiswitch environments. This is achieved via a custom configured policy. The policy makes the port listen but not learn or create a new entry in SISF. When a new host comes on board, the information will be distributed between all switches in the Layer 2 domain. If a switch has the same MAC entry as the new host, the switch will do a probe to confirm if the new host is legitimate and will accept or reject the host:

```
device-tracking policy DT_trunk_policy
       trusted-port
       device-role switch
## Applied typically on inter-switch links
interface <name>
       device-tracking policy DT_trunk_policy
```

Table 3 shows the priority that will be used to determine which programmatic policy to apply on the same target (VLAN or interface). Some SISF feature attributes can be “merged,” but that is not all.

Table 3. Policy priorities of SISF features

Features that enables SISF	Policy priority
LISP on VLAN with AR_RELAY	Description: The AR_RELAY feature converts a broadcast ARP packet into unicast. In this case the broadcasts are disabled in the underlay. SISF is enabled programmatically with priority 112.
LISP on VLAN without AR_RELAY	Description: In this case the broadcasts are enabled in the underlay. SISF is enabled programmatically with priority 112.
Ethernet VPN (EVPN) on VLAN	SISF is enabled programmatically with priority 96.
DHCP snooping	SISF is enabled programmatically with priority 80.

17.1(1) release

ND inspection has been deprecated in Release 17.1(1), as its functionality is already integrated via SISF. The ND CLIs will be removed in subsequent releases.

In Release 17.1, a new SISF feature was introduced to support scalability of the EVPN solution. This feature uses the device-tracking binding table to reduce the number of broadcasts.

SISF design and recommendations

This section provides key tips on how to design and configure SISF features in combination with other features.

- **Recommendation for the uplink (trunk) port**

  When the switch on the other side of a trunk port is also enabling device tracking, it is recommended to stop learning on the binding over the trunk port and apply only listening capability. That will help keep the SISF table size smaller, and it can be achieved by adding a trust policy to the interface. A sample policy is provided above with the name DT_trunk_policy.

- **Recommendation for IPv6 SVI device-tracking DAD probe on Layer 2 switches**

  When configured with "ipv6 neighbor tracking" on Layer 2 distribution switches, SISF will send a probe (every 5 minutes by default) to see if the target device is still present. The probe is using a format similar to IPv6 Duplicate Address Detection (DAD), carried by a Neighbor Solicitation (NS) packet. Unlike a normal IPv6 DAD packet, in the NS packet SISF populates the target MAC so that the probe is unicast.

  In a normal IPv6 DAD packet, the Source Address field in the IPv6 header is set to the unspecified address (0:0:0:0:0:0:0:0). An SISF probe can have two options. The order for choosing the source address in an SISF probe is:

  - Link-local address of Switch Virtual Interface (SVI), if configured
  - Use 0:0:0:0:0:0:0:0

  To configure a link-local IPv6 address on the SVI:

  ```
  interface vlan {X}
    ipv6 enable
  ```

- **Recommendation for avoiding the IPv4 "duplicate IP address" issue**

  One of the most common questions about SISF with ARP probes is the cause of duplicate IPs. The switch will send a message that looks like the following:

```
> Ethernet II, Src: 00:00:00_ff:95:ae (00:00:00:ff:95:ae), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Cisco_b7:60:83 (1c:e6:c7:b7:60:83)
    Sender IP address: 0.0.0.0
    Target MAC address: Cisco_47:7e:41 (a0:cf:5b:47:7e:41)
    Target IP address: 10.201.174.129
```

From the switch's point of view, the IP addresses used as the source in the ARP probes are not important, as the device is looking back to hear a response. Hence, this feature can be used on devices with no IP address configured at all, so the IP source of 0.0.0.0 is not relevant.

When the host receives these messages, it replies back and populates the destination IP field with the only IP address available in the received packet, which is its own IP address. This can cause false duplicate IP address alerts because the host that replies sees its own IP address as both the source and the destination of the packet.

With Microsoft Windows Vista and later versions, Microsoft introduced a new mechanism that is used to detect duplicate addresses on the network when the DHCP process occurs. This new detection flow is described in RFC 5227.

One of the triggers for this detection flow is defined in section 2.1.1 of RFC 5227:

> In addition, if during this period the host receives any ARP Probe where the packet's 'target IP address' is the address being probed for, and the packet's 'sender hardware address' is not the hardware address of any of the host's interfaces, then the host SHOULD similarly treat this as an address conflict and signal an error to the configuring agent as above. This can occur if two (or more) hosts have, for whatever reason, been inadvertently configured with the same address, and both are simultaneously in the process of probing that address to see if it can safely be used.

Causes of duplicate IP addresses

If the switch sends out an ARP probe for the client while the Microsoft Windows PC is in its duplicate address detection phase, Windows detects the probe as a duplicate IP address and presents a message that a duplicate IP address was found on the network for 0.0.0.0. The PC does not obtain an address, and the user must either manually release or renew the address, disconnect and reconnect to the network, or reboot the PC in order to gain network access.

Figure 7 shows the failed packet sequence.

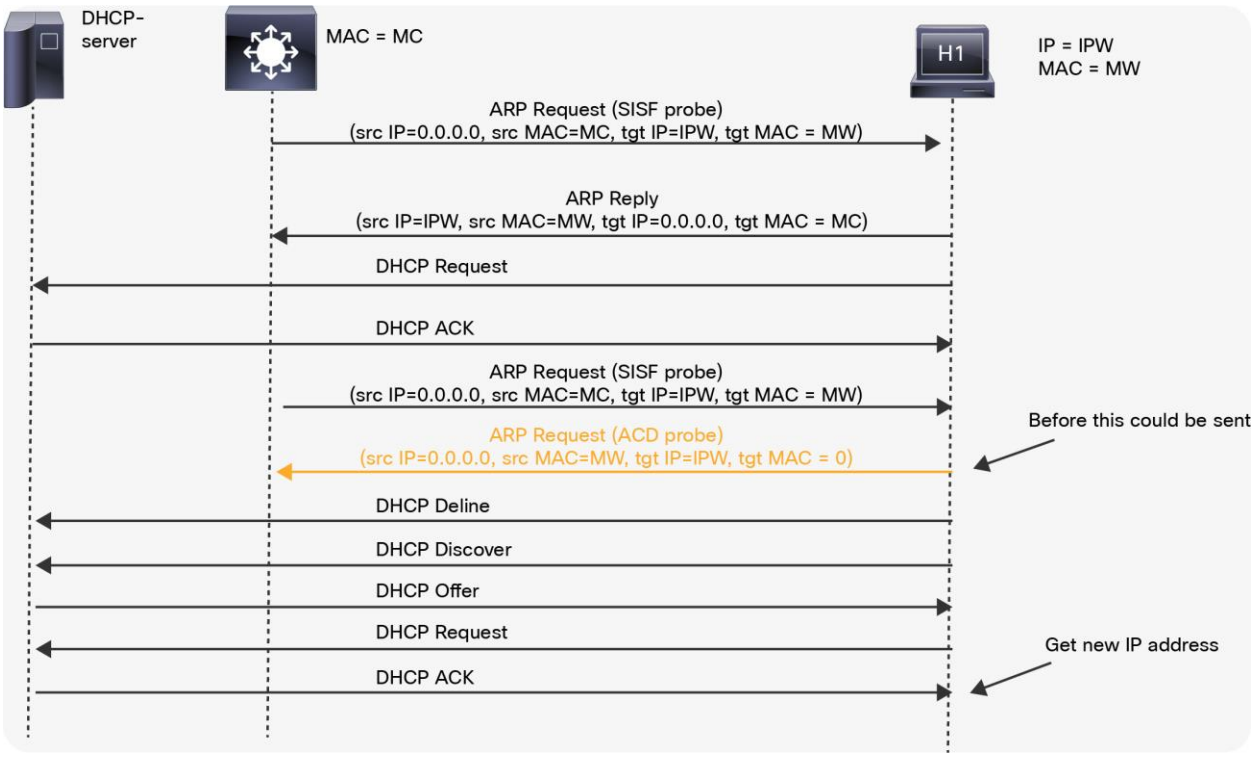

Figure 7.
Failed packet sequence

There are a few ways to solve duplicate IP address messages:

Enter the command "ip device tracking probe delay 10"

This command delays the probe for 10 seconds when it detects a link UP/flap, which minimizes the possibility that the probe will be sent while the host on the other side of the link checks for duplicate IP addresses. The RFC specifies a 10-second window for duplicate address detection, so if you delay the device-tracking probe, the issue can be solved in most cases.

If the switch sends out an ARP probe for the client while the host (for example, a Microsoft Windows PC) is in its duplicate address detection phase, the host detects the probe as a duplicate IP address and presents the user with a message that a duplicate IP address was found on the network. The PC might not obtain an address, and the user must manually release or renew the address, disconnect and reconnect to the network, or reboot the PC in order to gain network access.

During a probe delay, the delay resets itself if the switch detects a new probe from the PC/host. For example, if the probe timer has counted down to 5 seconds and detects an ARP probe from the PC/host, the timer resets back to 10 seconds.

Enter command "ip device tracking probe use-svi"

With this command, you can configure the switch to send a non-RFC-compliant ARP probe; the IP source will not be 0.0.0.0, but will be the SVI in the VLAN where the host resides. Microsoft Windows machines no longer see the probe as a probe as defined by RFC 5227 and do not flag a potential duplicate IP address.

Enter command "ip device tracking probe auto-source [fallback <host-ip> <mask>] [override]"

For customers who do not have predictable or controllable end devices, or for those who have many switches in a Layer 2–only role, the configuration of an SVI, which introduces a Layer 3 variable into the design, is not a suitable solution. The following CLI options are available for a Layer 2 design:

CLI: **ip device tracking probe auto-source**

- Set the source to the VLAN SVI if present, where this SVI is different than the default gateway and IP redirects are disabled.
- Search for a source/MAC pair in the IP host table for the same subnet.
- Send the zero IP source as in the default case.

Add the fallback option:

```
ip device tracking probe auto-source fallback 0.0.0.1 255.255.255.0
```

Compute the source IP from the destination IP with the host bit and mask provided.

Add the override option:

```
the ip device tracking probe auto-source fallback 0.0.0.1 255.255.255.0 override
```

Compute the source IP from the destination IP with the host bit and mask provided, where "override" makes you skip the search for an entry in the table.

As an example of the previous computations, assume that you probe host 192.168.1.200. With the mask and host bits provided, you generate a source address of 192.168.1.1.

If you probe entry 10.5.5.20, you would generate an ARP probe with source address 10.5.5.1, and so on.

The MAC address range for the auto-source will be the Layer 2 MAC address of the sending port.

- **Avoid a small reachable time for device-tracking binding**

  The reachability timer is refreshed at events such as host activity or discovery. If the timer expires, the reachability is affected. To avoid issues during migration from IP Device Tracking (IPDT) (older releases) to SISF, remove the below command:

  ```
  <device-tracking binding reachable-time 10>
  ```

- **SISF on a port channel**

  Device tracking on a port channel (or EtherChannel) is supported. But the configuration must be applied on the channel group, not the individual port-channel members. The only interface that shows up (and is known) from the binding standpoint is the port channel.

- **802.1X, MAB, and Cisco Trustsec**

  In Cisco IOS XE Release 16.X, SISF has to be additionally enabled if 802.1X, MAB, and Cisco Trustsec are used with downloadable ACLs or Security Ggroup Tag (SGT) assignment.

  Sample policy:

  ```
  device-tracking policy dot1x
        device-role node
        tracking enable
  interface g1/0/25
        device-tracking attach-policy dot1x
  ```

Verify SISF operation

Use these commands to verify SISF status on your device:

- **show device-tracking ...**

  This command displays interfaces where SISF is enabled and where MAC/IP/interface associations are currently tracked.

```
Switch# show device-tracking ?
   capture-policy    Message capture policy
   counters          Interface counters
   database          Show device-tracking database
   events            Display recent events history
   features          Registered device-tracking security features
   internal          Internal Information
   messages          Display recent messages history
   policies          Configured policies
   policy            Display a policy for feature device-tracking
```

- clear device tracking ...

  This command clears SISF-related entries.

```
Switch# clear device-tracking?
   Counters     Clear counters
   Database     Clear dynamic device-tracking bindings
   Events       Clear recent events history
   Messages     Clear recent messages history
```

Note: For further details on CLI commands, refer to the SISF configuration guide.

CLI migration to SISF syntax

Cisco IOS XE Release 3.X and later used a CLI syntax starting with "**ip**":

```
Switch(config)# ip device tracking
Switch(config)# ip device tracking probe count
Switch(config)# ip device tracking probe delay
Switch(config)# ip device tracking probe interval
Switch(config)# ip device tracking probe use-svi
```

Starting from Release 16.3, the CLI syntax for SISF has removed the "ip":

```
Switch(config)# ip device-tracking policy reachable-lifetime
Switch(config)# ip device-tracking policy retry-interval
Switch(config)# ip device-tracking policy <policy-name>
```

To make the migration to this new syntax easy, use the command device-tracking upgrade-cli. But if a migration is performed, it cannot be reverted.

When migrating from previous switches' IP device tracking to SISF, consider the following changes:

- SISF doesn't probe just because of a lack of recent activity unless configured otherwise. IPDT doesn't even allow the user to disable its probes.
- Any IP/MAC binding entries are refreshed when we detect "activity" packets related to the pair, such as ARP requests or replies or DHCP packets, not just for "responses" to our probes. Such responses are typically ARP replies and IPv6 neighbor advertisements. Probes are not indispensable. An important difference between SISF and IPDT is that if a host has been "silent" for a period of time, SISF only changes the state of the binding from REACHABLE to STALE rather than removing it, which further reduces the need to probe.
- Many customers are unable to configure SVIs and assign IP addresses on every Layer 2 access switch. But if SVI is configured on the Layer 2 access switch, SISF will use the SVI IP addresses as the sender's IP in its probes if they are configured, without any need to configure "use-svi," as in IPDT. It uses 0.0.0.0 as a last resort. Also, this is the only option that truly complies with the IETF standards; IPDT's "auto-source fallback" does not.
- When SISF does send a limited number of probes, it paces out successive probes using an "exponential backoff" algorithm to reduce losses due to collisions. Neither the number of probes nor the intervals between probes is configurable, unlike with IPDT. Probes will stop as soon as refresh activities are detected during the probing period, whether or not such activities are responses to the probes.
- SISF is aware of the distinction between access ports and trunk ports when a distributed SISF table is enabled (trusted inter-switch links) and doesn't send probes over trunk ports, unlike IPDT. We've seen many customers experiencing "duplicate IP address" errors caused by probes sent not by the switch directly connected to the host but by a remote switch connected via trunk ports multiple hops away.
- This can be achieved using specific policies and attaching them to targets (access and trunk ports) – which makes SISF aware of which port is a trunk vs. an access port. Without applying different policies, SISF will not make a distinction to resolve the duplicate IP address issues.
- Windows 7 hosts do not always report errors and give up their IP addresses every time they see packets that look like ARP Address Conflict Detection (ACD) (sender's IP == 0.0.0.0). They throw errors during certain vulnerable periods when they are in the process of renewing their DHCP leases. SISF's default policy dictates that SISF snoops on DHCP packets, and such DHCP lease renewal activity would refresh SISF's binding entry and thus remove the need to probe anyway. With IPDT, DHCP snooping needs to be configured.

Scale and performance

The SISF table can scale up to 100,000 entries per switch. SISF also offers optimization, which is tracked for cases where multiple IP addresses can be linked to a single MAC address to reduce the number of used entries. Such examples are on uplink interfaces where multiple IP addresses can be behind the same MAC address.

Starting from Release 16.9, the scale can change up to 4 IPv4 addresses per MAC address and 12 IPv6 addresses per MAC address based on the SISF configuration. When 802.1x, Cisco Trustsec, and IP Source Guard features are enabled, the number of IPv4 addresses per MAC address is reduced to 1.

Summary

Cisco Catalyst 9000 family switches offer flexible techniques to track device change and apply more first-hop security features. These techniques significantly increase network stability and reduce the number of attack vectors.

References

The following website offers more detailed information about the Cisco Catalyst 9000 family:

https://www.cisco.com/c/en/us/products/switches/catalyst-9000.html.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA C11-743346-00 01/20