Configuring IP Device Tracking

This chapter provides details about configuring IP Device Tracking (IPDT) on the IR8340 Router.

Information About IP Device Tracking

The main IPDT task is to keep track of connected hosts (association of MAC and IP address). In order to do this, it sends unicast Address Resolution Protocol (ARP) probes with a default interval of 30 seconds; these probes are sent to the MAC address of the host connected on the other side of the link, and use Layer 2 (L2) as the default source the MAC address of the physical interface out of which the ARP goes and a sender IP address of 0.0.0.0, based on the ARP Probe definition listed in RFC 5227.

In this document, the term 'ARP Probe' is used to refer to an ARP Request packet, broadcast on the local link, with an all-zero 'sender IP address'. The 'sender hardware address' MUST contain the hardware address of the interface sending the packet. The 'sender IP address' field MUST be set to all zeroes, to avoid polluting ARP caches in other hosts on the same link in the case where the address turns out to be already in use by another host. The 'target IP address' field MUST be set to the address being probed. An ARP Probe conveys both a question ("Is anyone using this address?") and an implied statement ("This is the address I hope to use.").

The purpose of IPDT is for the switch to obtain and maintain a list of devices that are connected to the switch via an IP address. The probe does not populate the tracking entry; it is simply used in order to maintain the entry in the table after it is learned through an ARP request/reply from the host.

IP ARP Inspection is enabled automatically when IPDT is enabled; it detects the presence of new hosts when it monitors ARP packets. If dynamic ARP inspection is enabled, only the ARP packets that it validates are used in order to detect new hosts for the Device Tracking table.

IP DHCP Snooping, if enabled, detects the presence or removal of new hosts when DHCP assigns or revokes their IP addresses.

IPDT is a feature that has always been available. However, on more recent Cisco IOS releases, its interdependencies are enabled by default (see Cisco bug ID CSCuj04986). It can be extremely useful when its database of IP/MAC hosts associations is used in order to populate the source IP of dynamic Access Control Lists (ACLs), or to maintain a binding of an IP address to a security group tag.

The ARP probe is sent under two circumstances:

- The link associated with a current entry in the IPDT database moves from a DOWN to an UP state, and the ARP entry has been populated.
- A link already in the UP state that is associated with an entry in the IPDT database has an expired probe interval.

Overview of SISF-Based Device Tracking

The Switch Integrated Security Features based (SISF-based) device tracking feature is part of the suite of first-hop security features.

The main role of the feature is to track the presence, location, and movement of end-nodes in the network. SISF snoops traffic received by the switch, extracts device identity (MAC and IP address), and stores them in a binding table. Many features, such as, Cisco TrustSec, IEEE 802.1X, LISP, and web authentication depend on the accuracy of this information to operate properly.

SISF-based device tracking supports both IPv4 and IPv6.

Even with the introduction of SISF-based device tracking, the legacy device tracking CLI (IP Device Tracking (IPDT) and IPv6 Snooping CLI) continues to be available. When you bootup the switch, the set of commands that is available depends on existing configuration, and only one of the following is available:

- SISF-based device tracking CLI, or
- IPDT and IPv6 Snooping CLI

SISF-based device tracking can be enabled manually (by using **device-tracking** commands), or programmatically (which is the case when providing device tracking services to other features).

Options to Enable SISF-Based Device Tracking

SISF-based device tracking is disabled by default.

You can enable it by defining a device tracking policy and attaching the policy to a specific target.

Note The target could be an interface or a VLAN.

Manually Enabling SISF-Based Device Tracking

- Option 1: Apply the **default** device tracking policy to a target.

  Enter the **device-tracking** command in the interface configuration mode or in the VLAN configuration mode. The system then attaches the **default** policy it to the interface or VLAN.

Note The **default** policy is a built-in policy with default settings; you cannot change any of the attributes of the **default** policy. In order to be able to configure device tracking policy attributes you must create a custom policy. See *Option 2: Create a custom policy with custom settings*.

- Option 2: Create a custom policy with custom settings.

  Enter the device-tracking policy command in global configuration mode and enter a custom policy name. The system creates a policy with the name you specify. You can then configure the available settings, in the device tracking configuration mode (config-device-tracking), and attach the policy to a specified target.

Programmatically Enabling SISF-Based Device Tracking

Some features rely on device tracking and utilize the trusted database of binding entries that SISF-based device tracking builds and maintains. These features, also called device tracking clients, enable device tracking programmatically (create and attach the device tracking policy).

Note The exceptions here are IEEE 802.1X, web authentication, Cisco TrustSec, and IP Source Guard (IPSG) - they also rely on device tracking, but they do not enable it. For these device tracking clients, you must enter the **ip dhcp snooping vlan** vlan command, to programmatically enable device tracking on a particular target.

Note the following about programmatically enabling SISF-based device tracking:

- A device tracking client *requires* device tracking to be enabled.

  There are several device tracking clients, therefore, multiple programmatic policies could be created. The settings of each policy differ depending on the device tracking client that creates the policy.

- The policy that is created, and its settings, are system-defined.

  Configurable policy attributes are available in the device tracking configuration mode (config-device-tracking) and vary from one release to another. If you try to modify an attribute that is not configurable, the configuration change is rejected and an error message is displayed.

How to Configure SISF-Based Device Tracking

Manually Enabling SISF-Based Device Tracking

Applying the Default Device Tracking Policy to a Target

Beginning in privileged EXEC mode, follow these steps to apply the default device tracking policy to an interface or VLAN:

Procedure

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

Step 3	Specify an interface or a VLAN.
	• **interface** *type number*
	• **vlan configuration** *vlan_list*
	Example:
	`Device(config)# interface gigabitethernet 0/1/0`
	OR
	`Device(config)# vlan configuration 100`
		interface *type number*—Specifies the interface and enters interface configuration mode. The device tracking policy will be attached to the specified interface.
		vlan configuration *vlan_list*—Specifies the VLANs and enters VLAN feature configuration mode. The device tracking policy will be attached to the specified VLAN.

Step 4	**device-tracking**
	Example:
	`Device(config-if)# device-tracking`
	OR
	`Device(config-vlan-config)# device-tracking`
		Enables SISF-based device tracking and attaches the default policy it to the interface or VLAN.
		The default policy is a built-in policy with default settings; none of the attributes of the default policy can be changed.

Step 5	**end**
	Example:
	`Device(config-if)# end`
	OR
	`Device(config-vlan-config)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.
		Exits VLAN feature configuration mode and returns to privileged EXEC mode.

Step 6	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy default`	Displays device-tracking policy configuration, and all the targets it is applied to.

Creating a Custom Device Tracking Policy with Custom Settings

Beginning in privileged EXEC mode, follow these steps to create and configure a device tracking policy:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	[**no**] **device-tracking policy** *policy_name* **Example:** `Device(config)# device-tracking policy example_policy`	Creates the policy and enters device-tracking configuration mode.

Step 4	[**data-glean** | **default** | **destination-glean** | **device-role** | **distribution-switch** | **exit** | **limit** | **no** | **prefix-glean** | **protocol** | **security-level** | **tracking** | **trusted-port** | **vpc**]
	Example:
	`Device(config-device-tracking)# security-level glean`
		Enter the question mark (?) at the system prompt to obtain a list of available options in this mode. You can configure the following for both IPv4 and IPv6:
		• (Optional) **data-glean** —Enables learning of addresses from a data packet snooped from a source inside the network and populates the binding table with the data traffic source address. Enter one of these options:
		  • log-only —Generates a syslog message upon data packet notification
		  • recovery —Uses a protocol to enable binding table recovery. Enter **NDP** or **DHCP** .
		• (Optional) **default** —Sets the policy attribute to its default value. You can set these policy attributes to their default values: **data-glean** , **destination-glean** , **device-role** , **limit** , **prefix-glean** , **protocol** , **security-level** , **tracking** , **trusted-port** .
		• (Optional) **destination-glean** —Populates the binding table by gleaning data traffic destination address. Enter one of these options:
		  • **log-only** —Generates a syslog message upon data packet notification
		  • **recovery** —Uses a protocol to enable binding table recovery. Enter **DHCP** .

	Command or Action	Purpose

		• (Optional) **device-role** —Sets the role of the device attached to the port. It can be a node or a switch. Enter one of these options:
		  • node —Configures the attached device as a node. This is the default option.
		  • switch —Configures the attached device as a switch.
		• (Optional) **distribution-switch** —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
		• **exit** —Exits the device-tracking policy configuration mode.
		• **limit address-count** —Specifies an address count limit per port. The range is 1 to 32000.
		• **no** —Negates the command or sets it to defaults.
		• (Optional) **prefix-glean** —Enables learning of prefixes from either IPv6 Router Advertisements or from DHCP-PD. You have the following option:
		  • (Optional) **only** —Gleans only prefixes and not host addresses.
		• (Optional) **protocol** —Sets the protocol to glean; by default, all are gleaned. Enter one of these options:
		  • **arp** [**prefix-list** *name*] —Gleans addresses in ARP packets. Optionally, enter the name of prefix-list that is to be matched.
		  • **dhcp4** [**prefix-list** *name*] —Glean addresses in DHCPv4 packets. Optionally, enter the name of prefix-list that is to be matched.
		  • **dhcp6** [**prefix-list** *name*] —Glean addresses in DHCPv6 packets. Optionally, enter the name of prefix-list that is to be matched.

	Command or Action	Purpose
		• **ndp** [**prefix-list** *name*] —Glean addresses in NDP packets. Optionally, enter the name of prefix-list that is to be matched. • **udp** [**prefix-list** *name*] —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect. • (Optional) **security-level** —Specifies the level of security enforced by the feature. Enter one of these options: • **glean** —Gleans addresses passively. • **guard** —Inspects and drops un-authorized messages. This is the default. • **inspect** —Gleans and validates messages. • (Optional) **tracking** —Specfies a tracking option. Enter one of these options: • **disable** [**stale-lifetime** [1-86400-seconds| **infinite**]] —Turns of device-tracking. Optionally, you can enter the duration for which the entry is kept inactive before deletion, or keep it permanently inactive. • **enable** [**reachable-lifetime** [1-86400-seconds| **infinite**]] —Turns on device-tracking. Optionally, you can enter the duration for which the entry is kept reachable, or keep it permanently reachable. • (Optional) **trusted-port** —Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.

	Command or Action	Purpose
		• (Optional) **vpc** —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.
Step 5	**end** **Example:** `Device(config-device-tracking)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 6	**show device-tracking policy** *policy-name* **Example:** `Device# show device-tracking policy default`	Displays the device-tracking policy configuration.

Attaching a Device Tracking Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach a device tracking policy to an interface:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies an interface and enters interface configuration mode.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the interface. Device tracking is also supported on EtherChannels. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.
Step 6	**show device-tracking policies** [**interface***interface-id*] **Example:** `Device# show device-tracking policies interface gigabitethernet 0/1/0`	Displays device-tracking policy configuration, and all the targets it is applied to.

Attaching a Device Tracking Policy to a VLAN

Beginning in privileged EXEC mode, follow these steps to attach a device-tracking policy to VLANs across multiple interfaces:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**vlan configuration** *vlan_list* **Example:** `Device(config)# vlan configuration 100`	Specifies an interface and enters interface configuration mode.
Step 4	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-vlan-config)# device-tracking attach-policy example_policy`	Attaches the device tracking policy to the specified VLANs across all switch interfaces. **Note** SISF based device-tracking policies can be disabled only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits device-tracking configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show device-tracking policies** [**vlan***vlan-id*] **Example:** `Device# show device-tracking policies vlan 100`	Verifies that the policy is attached to the specified VLAN, without exiting the VLAN interface configuration mode.

Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port

In a multi-switch network, SISF-based device tracking provides the capability to distribute binding table entries between switches running the feature. Binding entries are only created on the switches where the host appears on an access port. No entry is created for a host that appears over a trunk port. This is achieved by configuring a policy with the **trusted-port** and **device-role switch** options, and attaching it to the trunk port.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**device-tracking policy** *policy_name* **Example:** `Device(config)# device-tracking policy DT_trunk_policy`	Enters device-tracking policy configuration mode, for the specified policy.
Step 4	**device-role switch** **Example:** `Device(config-device-tracking)# device-role switch`	Specifies the role of the device attached to the port. Default is node. Enter the **device-role switch** option to stop the creation of binding entries for the port.
Step 5	**trusted-ports** **Example:** `Device(config-device-tracking)# trusted-port`	Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table.
Step 6	**end** **Example:** `Device(config-device-tracking)# end`	Exits device-tracking policy configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies a trunk interface and enters interface configuration mode.
Step 8	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy DT_trunk_policy`	Attaches a device tracking policy to the interface or the specified VLANs on the interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling SISF Syslogs

To enable syslogs of binding table events (such as create, delete, or modify entries), the following commands need to be executed:

```
device-tracking binding logging
```

If appropriate syslog level (6 - informational) need to be adjusted, execute:

```
logging console informational
```

To direct it to buffer:

```
logging buffered informational
```

to generate syslogs for MAC and/or IP theft events:

```
device-tracking logging theft
```

To generate syslogs for events when any of the SISF features decides to drop the packet for any reason:

```
device-tracking logging packet drop
```

to generate syslogs for events related to destination guard events:

```
device-tracking logging resolution-veto
```

the following command could be used to enable syslogs for all three event types listed above (but not to binding table events):

```
device-tracking logging
```

Example: DHCP Snooping Auto Enabling DT PROGRAMMATIC Policy

```
configure terminal
device-tracking policy Poo@12345
security-level glean
 device-role node
 limit address-count 10
tracking enable
  end
Switch(confg)#ip dhcp snooping
```

```
Switch(confg)#ip dhcp snooping vlan 100
Switch(confg)#end

configure terminal
interface Gi0/1/0
device-tracking attach-policy Poo@12345
end
```

Use the following show commands to display the status of device tracking:

```
router#show device-tracking policies
Target               Type  Policy               Feature        Target range
Gi0/1/0              PORT  Poo@12345            Device-tracking vlan all
vlan 100             VLAN  DT-PROGRAMMATIC      Device-tracking vlan all
router#
router#show device-tracking policy DT-PROGRAMMATIC
Device-tracking policy DT-PROGRAMMATIC configuration:
security-level glean
device-role node
gleaning from Neighbor Discovery
gleaning from DHCP6
gleaning from ARP
 gleaning from DHCP4
 NOT gleaning from protocol unkn
limit address-count for IPv4 per mac 1
 tracking (downlink only) enable
Policy DT-PROGRAMMATIC is applied on the following targets:
Target               Type  Policy               Feature        Target range
vlan 100             VLAN  DT-PROGRAMMATIC      Device-tracking vlan all
router#
router#show device-tracking policy Poo@12345
Device-tracking policy Poo@12345 configuration:
 security-level glean
 device-role node
 gleaning from Neighbor Discovery
gleaning from DHCP6
  gleaning from ARP
 gleaning from DHCP4
  gleaning from protocol unkn
 limit address-count 10
 tracking enable
Policy Poo@12345 is applied on the following targets:
Target               Type  Policy               Feature        Target range
Gi0/1/0              PORT  Poo@12345            Device-tracking vlan all

Router#show device-tracking database
Binding Table has 11 entries, 11 dynamic (limit 100000)
Codes: L - Local, S - Static, ND - Neighbor Discovery, ARP - Address Resolution Protocol,
DH4 - IPv4 DHCP, DH6 - IPv6 DHCP, PKT - Other Packet, API - API created
Preflevel flags (prlvl):
0001:MAC and LLA match     0002:Orig trunk            0004:Orig access
0008:Orig trusted trunk    0010:Orig trusted access   0020:DHCP assigned
0040:Cga authenticated     0080:Cert authenticated    0100:Statically assigned


 Network Layer Address                  Link Layer Address     Interface  vlan       prlvl
      age        state      Time left
ARP 100.1.1.1                               ac4a.6763.5a51         Gi0/1/10   100
0005       169s       REACHABLE  143 s try 0
DH4 100.0.0.14                              0013.0100.0004         Gi0/1/0    100
0024       53s        REACHABLE  259 s(31535947 s)
DH4 100.0.0.13                              0013.0100.0003         Gi0/1/0    100
0024       53s        REACHABLE  262 s(31535946 s)
DH4 100.0.0.12                              0013.0100.0002         Gi0/1/0    100
0024       52s        REACHABLE  250 s(31535947 s)
```