Cisco Secure DDoS Edge Protection

Table 1: Feature History Table

Feature Name	Release Information	Description
Cisco Secure DDoS Edge Protection	Release 24.1.1	You can now efficiently block malicious traffic, safeguarding your network's performance and availability. This is achieved as we have implemented protection against distributed denial-of-service (DDoS) attacks at the network edge, strategically deployed at the ingress point where external network traffic enters. By implementing a centralized controller and network of edge detectors, you enhance your network's resilience, ensuring proactive mitigation against potential threats for uninterrupted operation of your applications.

The Cisco Secure DDoS Edge Protection software solution stops DDoS attacks at the ingress side of the network.

The DDoS Edge Protection solution helps you detect DDoS attacks and take mitigation actions on the router. To enable detection services at the core network, you need to configure the following entities:

- DDoS Edge Protection Controller: This entity manages and monitors the Detector docker application, mitigates attacks, and oversees a distributed network of edge detectors. It analyzes detection trends across the network, orchestrates cross-network visibility and mitigation, and provides complete system management for the entire service.
- DDoS Edge Protection Detector: This entity is a real-time DDoS detector application that runs as a docker-application on a router with the DDoS controller. The DDOS controller can run on a cloud, server, or customer premises and is connected to this application.

The DDoS Edge Protection supports DDoS detection of both IPv4 and IPv6 traffic. You can choose the interface on which the traffic should be monitored. When the protection software solution is implemented, it filters the IPv4/IPv6 traffic flow and detects DDoS attacks.

Once a DDoS attack is detected, the DDoS Edge Protection Controller initiates a mitigation action, specifying the necessary steps to counteract the attack. This includes enabling the deny action as part of the mitigation measures and so on.

Supported Platforms

Cisco Secure DDoS Edge Protection is supported on the following routers, router processors, and line cards:

Table 2: Supported Platforms

Routers	Route processors	Line cards
• Cisco 8111-32EH • Cisco 8101-32FH • Cisco 8102-64H • Cisco 8101-32H • Cisco 8201-32FH • Cisco 8201-24H8FH • Cisco 8202-32FH-M • Cisco 8201-SYS • Cisco 8202-SYS	• Cisco 8804-RP • Cisco 8808-RP • Cisco 8812-RP • Cisco 8818-RP	• 8800-LC-48H • 8800-36-FH • 88-LC0-36FH-M • 88-LC0-36FH • 88-LC0-34H14FH • 8800-LC-36-FH

Prerequisites for Installing DDoS Edge Protection

- Configure the management interface to reach the DDoS controller IP address.
- Manually configure the base ACL, NetFlow, and SSH configurations.

Restrictions of DDoS Edge Protection Solution

- Only IPv4 and IPv6 traffic is supported.
- The DDoS Edge Protection does not support tunnel traffic.
- Only default VRF configuration is supported and is limited to the management port. To ensure smooth communication between the Docker and the controller, make sure to set up the management port exclusively in the default VRF.

Install and Configure DDoS Edge Protection

You can install the DDoS Edge Protection application through the DDoS edge protection controller. Perform the following:

1. Install and download the DDoS Edge Protection Controller Software package from the Software Download page. You can access the user interface, when the controller installation is complete.

   Log in to the controller services instance to monitor, manage, and control the device.

2. Perform the following base configurations such as ACL, NetFlow configuration, and SSH manually on the router:

   Configure Loopback

```
RP/0/RP0/CPU0:ios(config)#interface Loopback100
RP/0/RP0/CPU0:ios(config-if)# ipv4 address 15.1.1.2 255.255.255.255
RP/0/RP0/CPU0:ios(config)#interface Loopback101
RP/0/RP0/CPU0:ios(config-if)# ipv4 address 17.1.1.2 255.255.255.255
RP/0/RP0/CPU0:ios(config-if)#
```

   Configure Netflow

   The following netflow configuration is automatically carried out by the controller.

```
//Configuring Monitor Map
RP/0/RP0/CPU0:ios(config)#flow monitor-map DetectPro_Monitor_IPV6
RP/0/RP0/CPU0:ios(config)# record ipv6 extended
RP/0/RP0/CPU0:ios(config)#exporter DetectPro_GPB
RP/0/RP0/CPU0:ios(config)# cache entries 1000000
RP/0/RP0/CPU0:ios(config)#cache entries active 1
RP/0/RP0/CPU0:ios(config)#cache entries inactive 1
RP/0/RP0/CPU0:ios(config)#cache timeout inactive 1
RP/0/RP0/CPU0:ios(config)#cache timeout rate-limit 1000000
!
RP/0/RP0/CPU0:ios(config)#flow monitor-map DetectPro_Monitor_IPV4
RP/0/RP0/CPU0:ios(config)# record ipv4 extended
RP/0/RP0/CPU0:ios(config)#exporter DetectPro_GPB
RP/0/RP0/CPU0:ios(config)# cache entries 1000000
RP/0/RP0/CPU0:ios(config)#cache entries active 1
RP/0/RP0/CPU0:ios(config)#cache entries inactive 1
RP/0/RP0/CPU0:ios(config)#cache timeout inactive 1
RP/0/RP0/CPU0:ios(config)#cache timeout rate-limit 1000000
!
//Configuring Exporter Map
RP/0/RP0/CPU0:ios(config)#flow exporter-map DetectPro_GPB
RP/0/RP0/CPU0:ios(config)#version protobuf
RP/0/RP0/CPU0:ios(config)#transport udp 5005
RP/0/RP0/CPU0:ios(config)#source TenGigE0/0/0/16
RP/0/RP0/CPU0:ios(config)#destination 15.1.1.2
!
//Configuring Sampler Map
RP/0/RP0/CPU0:ios(config)#sampler-map DetectPro_NFv9
RP/0/RP0/CPU0:ios(config)#random 1 out-of 100
!
```

   Configure ACL

```
RP/0/RP0/CPU0:ios(config)#ipv4 access-list myACL
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 1301 permit ipv4 any any
!
RP/0/RP0/CPU0:ios(config)#ipv6 access-list myACL
```

```
RP/0/RP0/CPU0:ios(config-ipv6-acl)# 1301 permit ipv6 any any
!
```

For more information on implementing access lists and prefix lists, see Understanding Access-List.

If there is any DDoS attack, the controller performs the mitigation action using the ACL rule automatically.

The following is a sample configuration to deny DDoS attacker traffic using user defined ACE rule:

```
1 deny udp any eq 19 host 45.0.0.1 eq 0 packet-length eq 128 ttl eq 64
2 deny tcp any host 45.0.0.1 eq www match-all -established -fin -psh +syn -urg
packet-length eq 60 ttl eq 64
1301 permit ipv4 any any
```

Configuration updates are sent by the controller to the router.

Configure SSH

```
RP/0/RP0/CPU0:ios(config)#ssh server v2
RP/0/RP0/CPU0:ios(config)#ssh server netconf
RP/0/RP0/CPU0:ios(config)#netconf agent tty
RP/0/RP0/CPU0:ios(config)#netconf-yang agent ssh
!
RP/0/RP0/CPU0:ios(config)#ssh timeout 120
RP/0/RP0/CPU0:ios(config)#ssh server rate-limit 600
RP/0/RP0/CPU0:ios(config)#ssh server session-limit 110
RP/0/RP0/CPU0:ios(config)#ssh server v2
RP/0/RP0/CPU0:ios(config)#ssh server vrf default
RP/0/RP0/CPU0:ios(config)#ssh server netconf vrf default
```

3. Check the device connection to the DDoS controller using the **ping** command.

```
RP/0/RP0/CPU0:ios#ping 10.105.237.54
Thu Jun  1 07:16:43.654 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.105.237.54 timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
RP/0/RP0/CPU0:Router#bash
Thu Jun  1 07:16:53.024 UTC
[Router:~]$ping 10.105.237.54
PING 10.105.237.54 (10.105.237.54) 56(84) bytes of data.
64 bytes from 10.105.237.54: icmp_seq=1 ttl=63 time=1.73 ms
64 bytes from 10.105.237.54: icmp_seq=2 ttl=63 time=1.29 ms
64 bytes from 10.105.237.54: icmp_seq=3 ttl=63 time=1.27 ms
64 bytes from 10.105.237.54: icmp_seq=4 ttl=63 time=1.75 ms
^C
--- 10.105.237.54 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.270/1.510/1.751/0.230 ms
[Router:~]$
```

4. Add device details on the controller panel and ensure that all the three indicators (Deployment, Container, and Configuration) are green.

   For more information on installing the DDoS controller, see the DDoS Edge Protection Installation guide.

For more information on the DDoS Edge Protection, see Cisco Secure DDoS Edge Protection Data Sheet.

Verify DDoS Edge Protection Application Configuration

You can also verify if the DDoS controller pushes the CLI to the device using the following **show running-config** commands on the device:

```
RP/0/RP0/CPU0:Router#show running-config appmgr
Thu Jun  1 07:33:36.741 UTC
appmgr
 application esentryd
  activate type docker source esentryd-cisco-20230431633 docker-run-opts "--env-file
/harddisk:/ENV_6478443711ac6830700d1aeb --net=host"
 !
!
```

```
RP/0/RP0/CPU0:Router#show flow monitor DetectPro_Monitor_IPV4 cache location 0/0/CPU0
Thu Nov 16 06:13:38.066 UTC
Cache summary for Flow Monitor DetectPro_Monitor_IPV4:
Cache size:                         1000000
Current entries:                          0
Flows added:                     2243884200
Flows not added:                          0
Ager Polls:                      2243884200
  - Active timeout                        0
  - Inactive timeout                      0
  - Immediate                             0
  - TCP FIN flag                          0
  - Emergency aged                        0
  - Counter wrap aged                     0
  - Total                        2243884200
Periodic export:
  - Counter wrap                          0
  - TCP FIN flag                          0
Flows exported                   2243884200


Matching entries:                         0
!
```

```
RP/0/RP0/CPU0:Router#show flow monitor DetectPro_Monitor_IPV6 cache location 0/0/CPU0
Thu Nov 16 06:13:43.734 UTC
Cache summary for Flow Monitor DetectPro_Monitor_IPV6:
Cache size:                         1000000
Current entries:                          0
Flows added:                          59971
Flows not added:                          0
Ager Polls:                           94437
  - Active timeout                    59971
  - Inactive timeout                      0
  - Immediate                             0
  - TCP FIN flag                          0
  - Emergency aged                        0
  - Counter wrap aged                     0
  - Total                             59971
Periodic export:
  - Counter wrap                          0
  - TCP FIN flag                          0
Flows exported                        59971


Matching entries:                         0
```

```
RP/0/RP0/CPU0:Router#show  flow exporter
exporter  exporter-map
```

```
RP/0/RP0/CPU0:tortin#show  flow exporter DetectPro_GPB location 0/0/CPU0
Thu Nov 16 06:13:58.059 UTC
Flow Exporter: DetectPro_GPB
Export Protocol: protobuf
Flow Exporter memory usage: 5265344
Used by flow monitors: DetectPro_Monitor_IPV4
                       DetectPro_Monitor_IPV6

Status: Disabled
Transport:   UDP
Destination: 15.1.1.2        (5005) VRF default
Source:      0.0.0.0         (54482)
Flows exported:                                  0 (0 bytes)
Flows dropped:                                   0 (0 bytes)

Templates exported:                              0 (0 bytes)
Templates dropped:                               0 (0 bytes)

Option data exported:                            0 (0 bytes)
Option data dropped:                             0 (0 bytes)

Option templates exported:                       0 (0 bytes)
Option templates dropped:                        0 (0 bytes)

Packets exported:                         20355756 (27716506821 bytes)
Packets dropped:                                 0 (0 bytes)

Total export over last interval of:
  1 hour:                                       12 pkts
                                              1879 bytes
                                                12 flows
  1 minute:                                      0 pkts
                                                 0 bytes
                                                 0 flows
  1 second:                                      0 pkts
                                                 0 bytes
                                                 0 flows

RP/0/RP0/CPU0:Router#show appmgr application-table
Thu Nov 16 06:13:58.059 UTC
Name     Type   Config State Status
-------- ------ ------------ -------------------------------------------------
esentryd Docker  Activated   Up 8 minutes
RP/0/RP0/CPU0:Router#
```