



SecMon EMS *REST API Documentation V1.0*

Table of Contents

Overview	3
1. Scope	4
2. Classification Object	8
3. Rule Object	15
4. Policy	20
5. NetFlow Collector	24

6. NetFlow Config Params	29
7. NetFlow Monitor	34
8. NetFlow CollectorSet	38
9. NetFlow Association.....	43
10. RawForward	48
11. RawForward CollectorSet.....	53
12. RawForward Association.....	58
13. SFlow Collector.....	63
14. SFlow Config Params	68
15. SFlow CollectorSet	73
16. SFlow Association.....	78

Overview

Following feature's configurations are supported via REST-

- 1) Scope and Filtering Rule configuration
- 2) NetFlow configuration
- 3) sFlow configuration
- 4) Raw Packet Forwarding configuration

=>Details of each of the above mentioned features is given in sub-sequent sections.

1. Scope

Scope defines which monitoring protocols will be supported by SecMon Agent. Protocols that are enabled for a scope will be supported by the SecMon Agents of that scope.

Note: The scope_id to be used in update, delete and show can be fetched by listing the scope and mapping the scope name for which the operation is to be performed. ID corresponding to the scope name should be used as scope_id.

Parameters	Type	Value	Description
name	String	Scope name in “ ”	Name of the scope
sflowstatus	String	enable/disable in “ ”	enable/disable sflow plugin
netflowstatus	String	enable/disable in “ ”	enable/disable netflow plugin
rawforwardstatus	String	enable/disable in “ ”	enable/disable rawforward plugin

1.1 List Scope

Returns JSON data for all scopes.

URL	/v1.0/secmon/scope/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "name": "scope1", "sflowstatus": "disable", "netflowstatus": "disable", "rawforwardstatus": "enable", "id": "a23024fc-ed37-4a2b-a93b-c02ae5d603a9" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

1.2 Show Scope

Returns JSON data for scope based on scope_id provided.

URL	/v1.0/secmon/scope/<scope_id>/ Example: /v1.0/secmon/scope/a23024fc-ed37-4a2b-a93b-c02ae5d603a9/
Method	GET
URL Params	<scope_id>: unique id of scope
Data Params	None
Success Response	Code: 200 OK Content: { "id": "a23024fc-ed37-4a2b-a93b-c02ae5d603a9", "name": "scope4", "sflowstatus": "disable", "netflowstatus": "disable", "rawforwardstatus": "enable" }

Error Response	Code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }
-----------------------	--

1.3 Create Scope

Returns JSON data for created scope.

URL	/v1.0/secmon/scope/
Method	POST
URL Params	None
Data Params	<pre>{ "name": "scope5", "sflowstatus": "disable", "netflowstatus": "disable", "rawforwardstatus": "disable" }</pre>
Success Response	Code: 201 CREATED Content: { "name": "scope5", "sflowstatus": "disable", "netflowstatus": "disable", "rawforwardstatus": "disable", "id": "fcea3c1a-f8c3-4bc9-931b-943dc40beead" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist"

	<pre> } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" } </pre>
--	--

1.4 Update Scope

Returns JSON data for updated scope based on scope_id provided.

URL	/v1.0/secmon/scope/<scope_id>/ Example: /v1.0/secmon/scope/5c0b3c6d-35ed-4be2-9911-0b5a634a514b/
Method	PUT
URL Params	None
Data Params	<pre> { "name": "scope2", "sflowstatus": "disable", "netflowstatus": "enable", "rawforwardstatus": "disable" } </pre>
Success Response	Code: 200 OK Content: { <pre> "id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "name": "scope2", "sflowstatus": "disable", "netflowstatus": "enable", "rawforwardstatus": "disable" } </pre>
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: {

	<pre> "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" } or code: 400 BAD REQUEST content: { "detail": "\"id\" update not permitted" } </pre>
--	--

1.5 Delete Scope

Deletes the scope based on scope_id provided.

URL	/v1.0/secmon/scope/<scope_id>/ Example: /v1.0/secmon/scope/5c0b3c6d-35ed-4be2-9911-0b5a634a514b/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

2. Classification Object

Classification Objects are the rules that filter the traffic based on the configuration provided. This filtered traffic is passed to the core plugin module for monitoring and sample purposes.

Note: The classificationobject_id to be used in update, delete and show can be fetched by listing the classification object and mapping the classification object name for which the operation is to be performed. ID corresponding to the classification object name should be used as classificationobject_id.

Parameter	Type	Value	Description
name	String	Classification_object name in “ ”	Name of the Classification Object
src_ip	String	Ip address in “ ”. 0.0.0.0 defines traffic from all the source ip addresses to be forwarded	Source IP Address
src_mac	String	Source mac address in “ ”. “*” can be given to forward traffic from all mac	Source mac address
src_ip_subnet	Integer	min_value =0 max_value=32	Source subnet
minimum_src_port	Integer	min_value=1 max_value=65535	Minimum source port
maximum_src_port	Integer	min_value=1 max_value=65535	Maximum source port, min-max defines the range of L4 port to be filtered, for filtering only one port, keep min-src-port and max-src-port value same
dst_ip	String	Ip address in “ ”. 0.0.0.0 defines traffic from all the destination ip addresses to be forwarded	Destination IP address
dst_mac	String	Destination mac address in “ ”. “*” can be given to forward traffic from all mac	Destination mac address
dst_ip_subnet	Integer	min_value =0 max_value=32	Destination subnet
minimum_dst_port	Integer	min_value=1 max_value=65535	minimum dst port
maximum_dst_port	Integer	min_value=1 max_value=65535	maximum dst port; min-max defines the range of L4 port to be filtered, for filtering only one port, keep min-dst-port and max-dst-port value same
protocol	Integer	1: “ICMP” 6: “TCP” 17: “UDP” 132: “SCTP”	protocol of the packet

2.1 List Classification Object

Returns JSON data for all classification objects.

URL	/v1.0/secmon/classificationobject/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "name": "col123344", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535, "protocol": 1 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

2.2 Show Classification Object

Returns JSON data for classification object based on co_id provided.

URL	/v1.0/secmon/scope/<co_id>/
------------	-----------------------------

	Example: /v1.0/secmon/classificationobject/c10a2e21-504a-47d0-a9a4-b9945be9108a/
Method	GET
URL Params	<co_id>: unique id of classification object
Data Params	None
Success Response	Code: 200 OK Content: { "id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "name": "col123344", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535, "protocol": 1 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

2.3 Create Classification Object

Returns JSON data for created classification object.

URL	/v1.0/secmon/classificationobject/
------------	------------------------------------

Method	POST
URL Params	None
Data Params	<pre>{ "name": "col2", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535, "protocol": 1 }</pre>
Success Response	<p>Code: 201 CREATED</p> <p>Content: {</p> <pre> "id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "name": "col2", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535, "protocol": 1 </pre> <p>}</p>

Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }
-----------------------	--

2.4 Update Classification Object

Returns JSON data for updated classification object based on co_id provided.

URL	v1.0/secmon/classificationobject/<co_id> Example: v1.0/secmon/classificationobject/c10a2e21-504a-47d0-a9a4-b9945be9108a/
Method	PUT
URL Params	None
Data Params	{ "name": "col2", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535,

	<pre>"protocol": 1 }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre> "id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "name": "col2", "src_ip": "0.0.0.0", "src_mac": "*", "src_ip_subnet": 0, "minimum_src_port": 1, "maximum_src_port": 65535, "dst_ip": "0.0.0.0", "dst_mac": "*", "dst_ip_subnet": 0, "minimum_dst_port": 1, "maximum_dst_port": 65535, "protocol": 1 }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p>

	<pre>"detail": "\"id\" update not permitted" }</pre>
--	--

2.5 Delete Classification Object

Deletes the classificationobject based on co_id provided.

URL	/v1.0/secmon/classificationobject/<co_id> Example: /v1.0/secmon/classificationobject/c10a2e21-504a-47d0-a9a4-b9945be9108a/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

3. Rule Object

Rule Object specify if the traffic defined by a Classification object should be dropped or forwarded.

Note: The ruleobject_id to be used in update, delete and show can be fetched by listing the rule object and mapping the rule object name for which the operation is to be performed. ID corresponding to the rule object name should be used as ruleobject_id.

Parameter	Type	Value	Description
name	String	Rule Object name in “ “	Name of Rule Object
classificationobject_id	String	Id of classification object in “ “	Classification object id for which this Rule belongs to
priority	Integer	min_value: 1 max_value: 65535	the position of the rule in the rule database to match. Rule with priority ‘1’ has highest priority and so on.

truncate_to_size	Integer	min_value:0 max_value:1600	Truncate a packet to requested size(bytes), 0 to disable truncate to size feature
action	Integer	0: Drop 1: Forward	Whether to forward the packet or drop

3.1 List Rule Object

Returns JSON data for all rule objects.

URL	/v1.0/secmon/ruleobject/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4", "name": "ruleo123", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, "action": 1 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

3.2 Show Rule Object

Returns JSON data for rule object based on ro_id provided.

URL	/v1.0/secmon/ruleobject/<ro_id> Example: /v1.0/secmon/ruleobject/6b978be7-28a2-4b57-ad50-9bcc0594f3f4/
Method	GET

URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4", "name": "ruleo123", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, "action": 1 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

3.3 Create Rule Object

Returns JSON data for created rule object.

URL	/v1.0/secmon/ruleobject/
Method	POST
URL Params	None
Data Params	{ "name": "ruleo123", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, "action": 1 }

Success Response	Code: 201 CREATED Content: { "id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4", "name": "ruleo123", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, "action": 1 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

3.4 Update Rule Object

Returns JSON data for updated ruleobject based on ro_id provided.

URL	/v1.0/secmon/ruleobject/<ro_id> Example: /v1.0/secmon/ruleobject/6b978be7-28a2-4b57-ad50-9bcc0594f3f4/
Method	PUT
URL Params	None
Data Params	{ "name": "ruleo1", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, }

	<pre>"action": 1 }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4", "name": "ruleo1", "classificationobject_id": "c10a2e21-504a-47d0-a9a4-b9945be9108a", "priority": 1, "truncate_to_size": 0, "action": 1 }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

3.5 Delete Rule Object

Deletes the ruleobject based on rule_id provided.

URL	<p>/v1.0/secmon/ruleobject/<ro_id></p> <p>Example: /v1.0/secmon/ruleobject/6b978be7-28a2-4b57-ad50-9bcc0594f3f4/</p>
------------	--

Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

4. Policy

Policy defines a collection of rule objects i.e set of filtering rules. A policy is used when multiple filtering rules are need to be applied for traffic between two VMs

Note: The policy_id to be used in update, delete and show can be fetched by listing the policy and mapping the policy name for which the operation is to be performed. ID corresponding to the policy name should be used as policy_id.

Parameter	Type	Value	Description
Name	String	Policy name in " "	Name of policy
ruleobject_id	String	List of comma separated ruleobject ids in " "	Set of rules that belong to the policy

4.1 List Policy

Returns JSON data for all policys.

URL	/v1.0/secmon/policy/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "name": "policy123", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

4.2 Show Policy

Returns JSON data for policy based on policy_id provided.

URL	/v1.0/secmon/policy/<policy_id> Example: /v1.0/secmon/policy/9b09cae0-43df-48c4-9b8b-2dc9b6ba423c/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "name": "policy123", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4" }

Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }
-----------------------	--

4.3 Create Policy

Returns JSON data for created policy.

URL	/v1.0/secmon/policy/
Method	POST
URL Params	None
Data Params	{ "name": "policy1", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4" }
Success Response	Code: 201 CREATED Content: { "name": "policy1", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4", "id": "88ab035d-73ef-4e8a-b624-397849b9035f" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: {

	<pre>"detail": " similar record name exist" }</pre>
--	--

4.4 Update Policy

Returns JSON data for updated policy based on policy_id provided.

URL	/v1.0/secmon/policy/<policy_id> Example: /v1.0/secmon/policy/88ab035d-73ef-4e8a-b624-397849b9035f/
Method	PUT
URL Params	None
Data Params	<pre>{ "name": "policy2", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4" }</pre>
Success Response	Code: 200 OK Content: { <pre> "id": "88ab035d-73ef-4e8a-b624-397849b9035f", "name": "policy2", "ruleobject_id": "6b978be7-28a2-4b57-ad50-9bcc0594f3f4" }</pre>
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { <pre> "detail": "Resource already exist" }</pre> or code: 400 BAD REQUEST content: { <pre> "detail": " similar record name exist" }</pre> or code: 400 BAD REQUEST

	content: { "detail": "\"id\" update not permitted" }
--	---

4.5 Delete Policy

Deletes the policy based on policy_id provided.

URL	/v1.0/secmon/policy/<policy_id>/ Example: /v1.0/secmon/policy/88ab035d-73ef-4e8a-b624-397849b9035f/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

5. NetFlow Collector

NetFlow is a feature that provides the ability to collect IP network traffic as it enters or exits an interface. By analyzing the data provided by NetFlow, a network administrator can determine things such as the source and destination of traffic, class of service, and the causes of congestion.

NetFlow Collector is the tool that collects and analysis the traffic.

Note: The netflowcol_id to be used in update, delete and show can be fetched by listing the netflow collector and mapping the netflow collector name for which the operation is to be performed. ID corresponding to the netflow collector name should be used as netflowcol_id.

Parameter	Type	Value	Description
Name	String	NetFlow Collector name in " "	Name of NetFlow Collector
ip_address	String	Ip address in " "	IP Address on which NetFlow Collector is running

udp_port	Integer	min_value:1 max_value: 65535	port on which NetFlow Collector listens for Netflow flow packets
col_type	String	netflow/sflow/rawforward in " "	Collector type
encapsulation_protocol	String	None	Udp/sflow, only applicable for rawforward

5.1 List NetFlow Collector

Returns JSON data for all netflow collectors.

URL	/v1.0/secmon/collector/netflow/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "169760bd-7c06-4779-a1e7-37edf1409ca7", "name": "nfcol1", "ip_address": "10.42.31.114", "udp_port": 2055, "col_type": "netflow", "encapsulation_protocol": "" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

5.2 Show NetFlow Collector

Returns JSON data for netflow collector based on netflowcol_id provided.

URL	/v1.0/secmon/collector/<netflowcol_id> Example: /v1.0/secmon/collector/169760bd-7c06-4779-a1e7-37edf1409ca7/
------------	---

Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "169760bd-7c06-4779-a1e7-37edf1409ca7", "name": "nfc01", "ip_address": "10.42.31.114", "udp_port": 2055, "col_type": "netflow", "encapsulation_protocol": "" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

5.3 Create NetFlow Collector

Returns JSON data for created NetFlow Collector.

URL	/v1.0/secmon/collector/
Method	POST
URL Params	None
Data Params	{ "name": "netflowcol1", "ip_address": "10.42.0.113", "udp_port": 1234, "col_type": "netflow", "encapsulation_protocol": "" }

Success Response	Code: 201 CREATED Content: { "name": "netflowcol1", "ip_address": "10.42.0.113", "udp_port": 1234, "col_type": "netflow", "encapsulation_protocol": "", "id": "f30bc8b6-53a6-4b31-b15a-b20037773f7b" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

5.4 Update NetFlow Collector

Returns JSON data for updated netflow collector based on nfcol_id provided.

URL	/v1.0/secmon/collector/<nfcol_id>/ Example: /v1.0/secmon/collector/f30bc8b6-53a6-4b31-b15a-b20037773f7b/
Method	PUT
URL Params	None
Data Params	{ "name": "netflowcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "netflow", }

	<pre>"encapsulation_protocol": "" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "f30bc8b6-53a6-4b31-b15a-b20037773f7b", "name": "netflowcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "netflow", "encapsulation_protocol": "" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

5.5 Delete NetFlow Collector

Deletes the scope based on nfcol_id provided.

URL	<pre>/v1.0/secmon/collector/<nfcol_id>/</pre> <p>Example: /v1.0/secmon/collector/f30bc8b6-53a6-4b31-b15a-b20037773f7b/</p>
------------	--

Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

6. NetFlow Config Params

NetFlow Config Params define configuration parameters for netflow such as refresh rate, maxflows etc. these configuration are applicable per scope.

Parameter	Type	Value	Description
scope_id	String	Id of scope in “ ”	Id of scope for which netflow specific configurations are done
active_timeout	Integer	min_value:1 max_Value:65535	active time-out to export flow, in seconds
inactive_timeout	Integer	min_value:1 max_Value:65535	flow inactive timeout rate how frequently to re-send template(seconds)
refresh_rate	Integer	min_value:1 max_Value:65535	Netflow v9 template refresh rate (i.e. no of packets), after which template need to be exported
timeout_rate	Integer	min_value:1 max_Value:65535	NetFlow template timeout rate (i.e. in minutes) after which template need to be exported
maxflows	Integer	min_value:1 max_Value:65535	Maximum number of flows that can be created

6.1 List NetFlow Config Params

Returns JSON data for all netflow config params.

URL	/v1.0/secmon/netflowconfig/
Method	GET

URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "e3a4f58e-d08b-4da7-8dfd-8ae1951b0140", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 60, "inactive_timeout": 60, "refresh_rate": 60, "timeout_rate": 60, "maxflows": 60 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

6.2 Show NetFlow Config Params

Returns JSON data for netflow config param based on netflowconf_id provided.

URL	/v1.0/secmon/netflowconfig/<netflowconf_id> Example: /v1.0/secmon/netflowconfig/e3a4f58e-d08b-4da7-8dfd-8ae1951b0140/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "e3a4f58e-d08b-4da7-8dfd-8ae1951b0140", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 60, "inactive_timeout": 60, "refresh_rate": 60, "timeout_rate": 60, "maxflows": 60 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

6.3 Create NetFlow Config Params

Returns JSON data for created netflow config params.

URL	/v1.0/secmon/netflowconfig/
Method	POST
URL Params	None
Data Params	{ "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 60, "inactive_timeout": 60, "refresh_rate": 60, "timeout_rate": 60, "maxflows": 60 }

Success Response	Code: 201 CREATED Content: { "id": "e3a4f58e-d08b-4da7-8dfd-8ae1951b0140", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 60, "inactive_timeout": 60, "refresh_rate": 60, "timeout_rate": 60, "maxflows": 60 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" }

6.4 Update Netflow Config Params

Returns JSON data for updated netflow config based on nfconfig_id provided.

URL	/v1.0/secmon/netflowconfig/<nfconfig_id>/ Example: /v1.0/secmon/netflowconfig/e3a4f58e-d08b-4da7-8dfd-8ae1951b0140/
Method	PUT
URL Params	None
Data Params	{ "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 10, "inactive_timeout": 30, "refresh_rate": 10, "timeout_rate": 20, "maxflows": 10 }

Success Response	Code: 200 OK Content: { "id": "e3a4f58e-d08b-4da7-8dfd-8ae1951b0140", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "active_timeout": 10, "inactive_timeout": 30, "refresh_rate": 10, "timeout_rate": 20, "maxflows": 10 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": "\"id\" update not permitted" }

6.5 Delete Netflow Config Params

Deletes the netflow config based on nfconfig_id provided.

URL	/v1.0/secmon/netflowconfig/<nfconfig_id> Example: /v1.0/secmon/netflowconfig/e3a4f58e-d08b-4da7-8dfd-8ae1951b0140/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None

Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }
-----------------------	---

7. NetFlow Monitor

NetFlow Monitor Configures the NetFlow Monitor configuration required to work NetFlow for particular SecMon VNF

Parameter	Type	Value	Description
scope_id	String	Id of scope in “ ”	Id of scope for which netflow specific configurations are done
match_fields	String	List of comma separated match fields in “ ” where list has: TOS, PROTOCOL, SOURCE-ADDRESS, DESTINATION-ADDRESS, SOURCE-PORT, DESTINATION-PORT, INPUT-INTERFACE, MAC-ADDRESS, VLAN	The list of match fields to look for Netflow Flow
collect_fields	String	List of comma separated collect fields in “ ” where list has: COLLECT COUNTER, FLOW ACCESS TIMESTAMP, MAC-ADDRESS, VLAN	The list of collection fields to be exported in a Flow

7.1 List NetFlow Monitor

Returns JSON data for all netflow monitors.

URL	/v1.0/secmon/netflowmonitor/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "1f7d22bb-6b60-46de-af46-fbe2461364d1", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

7.2 Show NetFlow Monitor

Returns JSON data for netflow monitor based on netflowmonitor_id provided.

URL	/v1.0/secmon/netflowmonitor/<netflowmonitor_id> Example: /v1.0/secmon/netflowmonitor/1f7d22bb-6b60-46de-af46-fbe2461364d1/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "1f7d22bb-6b60-46de-af46-fbe2461364d1", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND

	content: { "detail": "Resource not found" }
--	--

7.3 Create Netflow Monitor

Returns JSON data for created netflow monitor.

URL	/v1.0/secmon/netflowmonitor/
Method	POST
URL Params	None
Data Params	<pre>{ "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" }</pre>
Success Response	Code: 201 CREATED Content: { "id": "1f7d22bb-6b60-46de-af46-fbe2461364d1", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" }

7.4 Update Netflow Monitor

Returns JSON data for updated netflow monitor based on nfmonitor_id provided.

URL	/v1.0/secmon/netflowmonitor/<nfmonitor_id>/
------------	---

	Example: /v1.0/secmon/netflowmonitor/1f7d22bb-6b60-46de-af46-fbe2461364d1/
Method	PUT
URL Params	None
Data Params	<pre>{ "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" }</pre>
Success Response	Code: 200 OK Content: { <pre> "id": "1f7d22bb-6b60-46de-af46-fbe2461364d1", "scope_id": "554d2665-f6ab-4f82-9ed7-096b93d84460", "match_fields": "TOS,PROTOCOL", "collect_fields": "MAC-ADDRESS,VLAN" </pre> }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { <pre> "detail": "Resource already exist" </pre> } or code: 400 BAD REQUEST content: { <pre> "detail": "\"id\" update not permitted" </pre> }

7.4 Delete Netflow Monitor

Deletes the netflow monitor based on nfmonitor_id provided.

URL	/v1.0/secmon/netflowmonitor/<nfmonitor_id>/
------------	---

	Example: /v1.0/secmon/netflowmonitor/1f7d22bb-6b60-46de-af46-fbe2461364d1/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

8. NetFlow CollectorSet

NetFlow CollectorSet is set of netflow collectors that share the traffic to be monitored between them. The sharing of traffic is done based on the algorithm defined in collectorset.

Note: The netflowcolset_id to be used in update, delete and show can be fetched by listing the netflow collectorset and mapping the netflow collectorset name for which the operation is to be performed. ID corresponding to the netflow collectorset name should be used as netflowcolset_id.

Parameter	Type	Value	Description
Name	String	Collectorset name in “ ”	Name of NetFlow CollectorSet
Algorithm	Integer	0: session_based 1: round_robin 2: weighted_round_robin	Algorithm defines the way sharing of traffic is done
Collector_ids	String	List of collectors separated by comma	Set of collectors that belong to the collectorset
Weight	Integer	min_value:0 max_value:100	Weight per collector in collectorset, only applicable in case of weighted_round_robin algorithm

8.1 Show all NetFlow CollectorSet

Returns JSON data for all netflow collectorsets.

URL	/v1.0/secmon/collectorset/netflow/
------------	------------------------------------

Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "672090ff-5574-4837-8108-5c1ae89bfad3", "name": "nfcolset123", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 30}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 40}]", "col_type": "netflow", "lb_algo": "2" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

8.2 Show NetFlow CollectorSet

Returns JSON data for netflow collectorset based on netflowcolset_id provided.

URL	/v1.0/secmon/collectorset/<netflowcolset_id> Example: /v1.0/secmon/collectorset/672090ff-5574-4837-8108-5c1ae89bfad3/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "672090ff-5574-4837-8108-5c1ae89bfad3", "name": "nfcset123", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 30}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 40}]", "col_type": "netflow", "lb_algo": "2" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

8.3 Create Netflow CollectorSet

Returns JSON data for created netflow collectorset.

URL	/v1.0/secmon/collectorset/
Method	POST
URL Params	None
Data Params	{ "name": "nfcset1", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 40}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 50}]", "col_type": "netflow", "lb_algo": "2" }

Success Response	Code: 201 CREATED Content: { "name": "nfcset1", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 40}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 50}]", "col_type": "netflow", "lb_algo": 2, "id": "61183971-e4da-49c2-9488-199f1436300b" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

8.4 Update Netflow CollectorSet

Returns JSON data for updated netflow collectorset based on nfcset_id provided.

URL	/v1.0/secmon/collectorset/<nfcset_id>/ Example: /v1.0/secmon/collectorset/61183971-e4da-49c2-9488-199f1436300b/
Method	PUT
URL Params	None
Data Params	{ "name": "nfcset1", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 20}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 30}]", }

	<pre>"col_type": "netflow", "lb_algo": "2" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "61183971-e4da-49c2-9488-199f1436300b", "name": "nfcset1", "collector_ids": "[{'id': '8302786b-d3fc-4f2a-aa50-6feb0e00acea', 'weight': 20}, {'id': '855adab6-0537-42a3-be4a-351946520a6b', 'weight': 30}]", "col_type": "netflow", "lb_algo": "2" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

8.5 Delete Netflow CollectorSet

Deletes the scope based on nfcset_id provided.

URL	/v1.0/secmon/collectorset/<nfcset_id>/
------------	--

	Example: /v1.0/secmon/collectorset/61183971-e4da-49c2-9488-199f1436300b/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

9. NetFlow Association

NetFlow Association associates netflow collector/collectorset with the traffic to be monitored and also associates policy with it.

Parameter	Type	Value	Description
collector_id	String	Collector id in “ ”	The Collector ID for which this association belongs to.
direction	String	INGRESS/EGRESS/BOTH in “ ”	The traffic direction.
policy_id	String	Policy id in “ ”	The id of the policy.
scope_id	String	Scope id in “ ”	Id of the scope.

9.1 List NetFlow Association

Returns JSON data for all netflow associations.

URL	/v1.0/secmon/netflowassociation/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "75c45045-e21a-4c07-9fec-720bc0b0a5e4", "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

9.2 Show NetFlow Association

Returns JSON data for netflow association based on netflowassociation_id provided.

URL	/v1.0/secmon/netflowassociation/<netflowassociation_id> Example: /v1.0/secmon/netflowassociation/75c45045-e21a-4c07-9fec-720bc0b0a5e4/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "75c45045-e21a-4c07-9fec-720bc0b0a5e4", "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

9.3 Create NetFlow Association

Returns JSON data for created netflow association.

URL	/v1.0/secmon/netflowassociation/
Method	POST
URL Params	None
Data Params	{ "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }

Success Response	Code: 201 CREATED Content: { "id": "75c45045-e21a-4c07-9fec-720bc0b0a5e4", "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " record with same scope and collector already exist in association" }

9.4 Update NetFlow Association

Returns JSON data for updated netflow association based on nfassociation_id provided.

URL	/v1.0/secmon/netflowassociation/<nfassociation_id>/ Example: /v1.0/secmon/netflowassociation/75c45045-e21a-4c07-9fec-720bc0b0a5e4/
Method	PUT
URL Params	None
Data Params	{ "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "EGRESS",

	<pre>"policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "75c45045-e21a-4c07-9fec-720bc0b0a5e4", "collector_id": "Collector:8302786b-d3fc-4f2a-aa50-6feb0e00acea", "direction": "EGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }}</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " record with same scope and collector already exist in association" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

9.5 Delete NetFlow Association

Deletes the netflow assocaition based on nfassociation_id provided.

URL	/v1.0/secmon/netflowassociation/<nfassociation_id>/ Example: /v1.0/secmon/netflowassociation/75c45045-e21a-4c07-9fec-720bc0b0a5e4/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

10. RawForward

RawForward forwards all traffic that comes from 5Tuple filtering to the monitoring tools machine. Raw Forwarding shall not perform any sampling/monitoring to the incoming traffic. It encapsulates the traffic in UDP and Sflow header; secure it using IPsec and forward towards the configured monitoring tools for traffic analysis.

RawForward Collector is a tool that collects and analysis the the traffic.

Note: The rawcol_id to be used in update, delete and show can be fetched by listing the rawforward collector and mapping the rawforward collector name for which the operation is to be performed. ID corresponding to the rawforward collector name should be used as rawcol_id.

Parameter	Type	Value	Description
Name	String	RawForward Collector name in " "	Name of RawForward Collector
ip_address	String	Ip address in " "	IP Address on which RawForward Collector is running
udp_port	Integer	min_value:1 max_Value: 65535	port on which RawForward Collector listens for RawForward flow packets
col_type	String	"rawforward"	Collector type
encapsulation_protocol	String	udp/sflow in " "	Udp/sflow protocol type of which the packets will be forwarded.

10.1 List RawForward Collector

Returns JSON data for all rawforward collectors.

URL	/v1.0/secmon/collector/rawforward/
Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "8baef94b-23d1-40b7-bb78-b48b8c77a418", "name": "rawcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", "encapsulation_protocol": "SFLOW" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

10.2 Show RawForward Collector

Returns JSON data for rawforward collector based on rawcol_id provided.

URL	/v1.0/secmon/collector/<rawcol_id> Example: /v1.0/secmon/collector/8baef94b-23d1-40b7-bb78-b48b8c77a418/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "8baef94b-23d1-40b7-bb78-b48b8c77a418", "name": "rawcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", "encapsulation_protocol": "SFLOW" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

10.3 Create RawForward Collector

Returns JSON data for created rawforward collector.

URL	/v1.0/secmon/collector/rawforward/
Method	POST
URL Params	None
Data Params	{ "name": "rawcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", "encapsulation_protocol": "SFLOW" }

Success Response	Code: 201 CREATED Content: { "id": "8baef94b-23d1-40b7-bb78-b48b8c77a418", "name": "rawcol1", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", "encapsulation_protocol": "SFLOW" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

10.4 Update RawForward Collector

Returns JSON data for updated rawforward collector based on rawcol_id provided.

URL	/v1.0/secmon/collector/<rawcol_id>/ Example: /v1.0/secmon/collector/8baef94b-23d1-40b7-bb78-b48b8c77a418/
Method	PUT
URL Params	None
Data Params	{ "name": "rawco12", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", }

	<pre>"encapsulation_protocol": "SFLOW" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "8baef94b-23d1-40b7-bb78-b48b8c77a418", "name": "rawco12", "ip_address": "10.42.0.113", "udp_port": 6343, "col_type": "rawforward", "encapsulation_protocol": "SFLOW" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

10.5 Delete RawForward Collector

Deletes the rawforward collector based on rawcol_id provided.

URL	<pre>/v1.0/secmon/collector/<rawcol_id>/</pre> <p>Example: /v1.0/secmon/collector/8baef94b-23d1-40b7-bb78-b48b8c77a418/</p>
------------	---

Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

11. RawForward CollectorSet

RawForward CollectorSet is set of rawforward collectors that share the traffic to be monitored between them. The sharing of traffic is done based on the algorithm defined in collectorset.

Note: The rawcolset_id to be used in update, delete and show can be fetched by listing the rawforward collectorset and mapping the rawforward collectorset name for which the operation is to be performed. ID corresponding to the rawforward collectorset name should be used as rawcolset_id.

Parameter	Type	Value	Description
Name	String	Collectorset name in “ ”	Name of RawForward CollectorSet
Algorithm	Integer	0: session_based 1: round_robin 2: weighted_round_robin	Algorithm defines the way sharing of traffic is done
Collector_ids	String	List of collectors separated by comma	Set of collectors that belong to the collectorset
Weight	Integer	min_value:0 max_value:100	Weight per collector in collectorset, only applicable in case of weighted_round_robin algorithm

11.1 List RawForward CollectorSet

Returns JSON data for all rawforward collectorsets.

URL	/v1.0/secmon/collectorset/rawforward/
Method	GET
URL Params	None

Data Params	None
Success Response	Code: 200 OK Content: { "id": "07f9826f-99cb-4f6f-b286-544bcab6d8c7", "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '0'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '0'}]", "col_type": "rawforward", "lb_algo": 0 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

11.2 Show RawForward CollectorSet

Returns JSON data for rawforward collectorset based on rawcolset_id provided.

URL	/v1.0/secmon/collectorset/<rawcolset_id> Example: /v1.0/secmon/collectorset/07f9826f-99cb-4f6f-b286-544bcab6d8c7/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "07f9826f-99cb-4f6f-b286-544bcab6d8c7", "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '0'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '0'}]", "col_type": "rawforward", "lb_algo": 0 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

11.3 Create RawForward CollectorSet

Returns JSON data for created rawforward collectorset.

URL	/v1.0/secmon/collectorset/
Method	POST
URL Params	None
Data Params	<pre>{ "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '10'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '20'}]", "col_type": "rawforward", "lb_algo": "2" }</pre>

Success Response	Code: 201 CREATED Content: { "id": "07f9826f-99cb-4f6f-b286-544bcab6d8c7", "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '0'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '0'}]", "col_type": "rawforward", "lb_algo": 0 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

11.4 Update RawForward CollectorSet

Returns JSON data for updated rawforward collectorset based on rawcolset_id provided.

URL	/v1.0/secmon/collectorset/<rawcolset_id>/ Example: /v1.0/secmon/collectorset/07f9826f-99cb-4f6f-b286-544bcab6d8c7/
Method	PUT
URL Params	None
Data Params	{ "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '20'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '30'}]", "col_type": "rawforward", }

	<pre>"lb_algo": "2" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre> "name": "rawcolset1", "collector_ids": "[{'id': '8baef94b-23d1-40b7-bb78-b48b8c77a418', 'weight': '20'}, {'id': 'f7c7cae0-acee-4e84-a364-c433699d12ea', 'weight': '30'}]", "col_type": "rawforward", "lb_algo": "2" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": "\"id\" update not permitted" }</pre>

11.5 Delete RawForward CollectorSet

Deletes the rawforward collectorset based on rawcolset_id provided.

URL	<pre>/v1.0/secmon/collectorset/<rawcolset_id>/</pre> <p>Example: /v1.0/secmon/collectorset/07f9826f-99cb-4f6f-b286-544bcab6d8c7/</p>
------------	--

Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

12. RawForward Association

RawForward Association associates rawforward collector/collectorset with the traffic to be monitored and also associates policy with it.

Parameter	Type	Value	Description
collector_id	String	Collector id in “ ”	The Collector ID for which this association belongs to.
direction	String	INGRESS/EGRESS/BOTH in “ ”	The traffic direction.
policy_id	String	Policy id in “ ”	The id of the policy.
scope_id	String	Scope id in “ ”	Id of the scope.

12.2 List RawForward Association

Returns JSON data for all rawforward associations.

URL	/v1.0/secmon/rawforwardassociation/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "014aed0a-7f9f-4661-ae7f-15bc80379143", "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

12.2 Show RawForward Association

Returns JSON data for rawforward association based on rawassociation_id provided.

URL	/v1.0/secmon/rawforwardassociation/<rawassociation_id> Example: /v1.0/secmon/rawforwardassociation/014aed0a-7f9f-4661-ae7f-15bc80379143/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "014aed0a-7f9f-4661-ae7f-15bc80379143", "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

12.3 Create RawForward Association

Returns JSON data for created rawforward association.

URL	/v1.0/secmon/rawforwardassociation/
Method	POST
URL Params	None
Data Params	{ "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }

Success Response	Code: 201 CREATED Content: { "id": "014aed0a-7f9f-4661-ae7f-15bc80379143", "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } Or code: 400 BAD REQUEST content: { "detail": " record with same scope and collector already exist in association" }

12.4 Update RawForward Association

Returns JSON data for updated rawforward association based on rawassociation_id provided.

URL	/v1.0/secmon/rawforwardassociation/<rawassociation_id>/ Example: /v1.0/secmon/rawforwardassociation/014aed0a-7f9f-4661-ae7f-15bc80379143/
Method	PUT
URL Params	None
Data Params	{ "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418",

	<pre> "direction": "EGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" } </pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre> "id": "014aed0a-7f9f-4661-ae7f-15bc80379143", "collector_id": "Collector:8baef94b-23d1-40b7-bb78-b48b8c77a418", "direction": "EGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" } </pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": "Resource already exist" } </pre> <p>Or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": " record with same scope and collector already exist in association" } </pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre> "detail": "\"id\" update not permitted" } </pre>

12.5 Delete RawForward Association

Deletes the rawforward association based on rawassociation_id provided.

URL	/v1.0/secmon/rawforwardassociation/<rawassociation_id>/ Example: /v1.0/secmon/rawforwardassociation/014aed0a-7f9f-4661-ae7f-15bc80379143/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

13. SFlow Collector

SFlow provides a means for sampling of incoming packets, together with interface counters. Sampled packets/operation and counter information referred as flow sample and counter samples are sent as SFlow datagrams to a central server running software that analyzes and reports on network traffic; the SFlow collector.

Note: The sflowcol_id to be used in update, delete and show can be fetched by listing the sflow collector and mapping the sflow collector name for which the operation is to be performed. ID corresponding to the sflow collector name should be used as sflowcol_id.

Parameter	Type	Value	Description
Name	String	SFlow Collector name in " "	Name of SFlow Collector
ip_address	String	Ip address in " "	IP Address on which SFlow Collector is running
udp_port	Integer	min_value:1 max_Value: 65535	port on which SFlow Collector listens for SFlow flow packets
col_type	String	"sflow"	Collector type
encapsulation_protocol	String	udp/sflow in " "	Udp/sflow protocol type of which the packets will be forwarded.

13.1 List SFlow Collector

Returns JSON data for all sflow collectors.

URL	/v1.0/secmon/collector/sflow/
------------	-------------------------------

Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "4a9d9416-17b6-40b1-9c71-fd550f60efff", "name": "sfcol1", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", "encapsulation_protocol": "" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

13.2 Show SFlow Collector

Returns JSON data for sflow collector based on sflowcol_id provided.

URL	/v1.0/secmon/collector/<sflowcol_id> Example: /v1.0/secmon/collector/4a9d9416-17b6-40b1-9c71-fd550f60efff/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "4a9d9416-17b6-40b1-9c71-fd550f60efff", "name": "sfcol1", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", "encapsulation_protocol": "" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

13.3 Create SFlow Collector

Returns JSON data for created sflow collector.

URL	/v1.0/secmon/collector/
Method	POST
URL Params	None
Data Params	<pre>{ "id": "4a9d9416-17b6-40b1-9c71-fd550f60efff", "name": "sfcol123", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", "encapsulation_protocol": "" }</pre>

Success Response	Code: 201 CREATED Content: { "name": "sfc0l123", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", "encapsulation_protocol": "", "id": "4a9d9416-17b6-40b1-9c71-fd550f60efff" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

13.4 Update SFlow Collector

Returns JSON data for updated sflow collector based on sflowcol_id provided.

URL	/v1.0/secmon/collector/<sflowcol_id>/ Example: /v1.0/secmon/collector/4a9d9416-17b6-40b1-9c71-fd550f60efff/
Method	PUT
URL Params	None
Data Params	{ "name": "sfc0l12345", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", }

	<pre>"encapsulation_protocol": "" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "4a9d9416-17b6-40b1-9c71-fd550f60efff", "name": "sflowcol12345", "ip_address": "10.42.0.116", "udp_port": 6343, "col_type": "sflow", "encapsulation_protocol": "" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

13.5 Delete SFlow Collector

Deletes the sflow collector based on sflowcol_id provided.

URL	<pre>/v1.0/secmon/collector/<sflowcol_id>/</pre> <p>Example: /v1.0/secmon/collector/4a9d9416-17b6-40b1-9c71-fd550f60efff/</p>
------------	---

Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

14. SFlow Config Params

SFlow Config Params defines Configurations such as agent-ip, sampling rate etc. These configurations are applicable per scope.

Parameter	Type	Value	description
scope_id	String	Scope id in “ ”	Id of scope for which sflow specific configurations are done
agent_ip	String	Ip address in “ ”	Ip address of the SFlow agent that sends the data
agent_subid	Integer	min_value: 0 max_value: 65535	Subid of sflow agent, by default 0
sampling_rate	Integer	min_value:1 max_value:65535	1-in-N packet sampling
truncate_to_size	Integer	min_value:0 max_value:65535	Enter value to truncate packet to requested size(bytes), 0 not allowed

14.1 List SFlow Config Params

Returns JSON data for all sflow config params.

URL	/v1.0/secmon/sflowconfig/
------------	---------------------------

Method	GET
URL Params	None
Data Params	None
Success Response	Code: 200 OK Content: { "id": "d4e72987-28d4-4d71-a8eb-24930a2f1e39", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 0, "sampling_rate": 20, "truncate_to_size": 128 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

14.2 Show SFlow Config Params

Returns JSON data for sflow config param based on sflowconfig_id provided.

URL	/v1.0/secmon/sflowconfig/<sflowconfig_id> Example: /v1.0/secmon/sflowconfig/d4e72987-28d4-4d71-a8eb-24930a2f1e39/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "d4e72987-28d4-4d71-a8eb-24930a2f1e39", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 0, "sampling_rate": 20, "truncate_to_size": 128 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

14.3 Create SFlow Config Params

Returns JSON data for created sflow config params.

URL	/v1.0/secmon/sflowconfig/
Method	POST
URL Params	None
Data Params	{ "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 0, "sampling_rate": 20, "truncate_to_size": 128 }

Success Response	Code: 201 CREATED Content: { "id": "d4e72987-28d4-4d71-a8eb-24930a2f1e39", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 0, "sampling_rate": 20, "truncate_to_size": 128 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" }

14.4 Update SFlow Config Params

Returns JSON data for updated sflow config based on sflowconfig_id provided.

URL	/v1.0/secmon/sflowconfig/<sflowconfig_id>/ Example: /v1.0/secmon/sflowconfig/d4e72987-28d4-4d71-a8eb-24930a2f1e39/
Method	PUT
URL Params	None
Data Params	{ "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 1, "sampling_rate": 20, "truncate_to_size": 128 }

Success Response	Code: 200 OK Content: { "id": "d4e72987-28d4-4d71-a8eb-24930a2f1e39", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b", "agent_ip": "0.0.0.0", "agent_subid": 1, "sampling_rate": 20, "truncate_to_size": 128 }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": "\"id\" update not permitted" }

14.5 Delete SFlow Config Params

Deletes the sflow config based on sflowconfig_id provided.

URL	/v1.0/secmon/sflowconfig/<sflowconfig_id>/ Example: /v1.0/secmon/sflowconfig/d4e72987-28d4-4d71-a8eb-24930a2f1e39/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR

	or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }
--	---

15. SFlow CollectorSet

SFlow CollectorSet is set of sflow collectors that share the traffic to be monitored between them. The sharing of traffic is done based on the algorithm defined in collectorset.

Note: The sflowcolset_id to be used in update, delete and show can be fetched by listing the sflow collectorset and mapping the sflow collectorset name for which the operation is to be performed. ID corresponding to the sflow collectorset name should be used as sflowcolset_id.

Parameter	Type	Value	Description
Name	String	Collectorset name in “ ”	Name of SFlow CollectorSet
Algorithm	Integer	0: session_based 1: round_robin 2: weighted_round_robin	Algorithm defines the way sharing of traffic is done
Collector_ids	String	List of collectors separated by comma	Set of collectors that belong to the collectorset
Weight	Integer	min_value:0 max_value:100	Weight per collector in collectorset, only applicable in case of weighted_round_robin algorithm

15.1 List SFlow CollectorSet

Returns JSON data for all sflow collectorsets.

URL	/v1.0/secmon/collectorset/sflow/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "36d59412-48d0-41ba-bbb3-0be168ed371e", "name": "sflowset1", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", "col_type": "sflow", "lb_algo": 2 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

15.2 Show SFlow CollectorSet

Returns JSON data for sflow collectorset based on sflowcolset_id provided.

URL	/v1.0/secmon/collectorset/<sflowcolset_id>/ Example: /v1.0/secmon/collectorset/36d59412-48d0-41ba-bbb3-0be168ed371e/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "36d59412-48d0-41ba-bbb3-0be168ed371e", "name": "sfcolset1", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", "col_type": "sflow", "lb_algo": 2 }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

15.3 Create SFlow CollectorSet

Returns JSON data for created sflow collectorset.

URL	/v1.0/secmon/collectorset/
Method	POST
URL Params	None
Data Params	<pre>{ "name": "sfcolset2", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", "col_type": "sflow", "lb_algo": "2" }</pre>

Success Response	Code: 201 CREATED Content: { "id": "36d59412-48d0-41ba-bbb3-0be168ed371e", "name": "sfcolset1", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", "col_type": "sflow", "lb_algo": "2" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " similar record name exist" }

15.4 Update SFlow CollectorSet

Returns JSON data for updated sflow collectorset based on sflowcolset_id provided.

URL	/v1.0/secmon/collectorset/<sflowcolset_id>/ Example: /v1.0/secmon/collectorset/36d59412-48d0-41ba-bbb3-0be168ed371e/
Method	PUT
URL Params	None
Data Params	{ "name": "sfcolset3", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", }

	<pre>"col_type": "sflow", "lb_algo": "2" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "36d59412-48d0-41ba-bbb3-0be168ed371e", "name": "sfcolset3", "collector_ids": "[{'id': '4a9d9416-17b6-40b1-9c71-fd550f60efff', 'weight': 30}, {'id': '60ea2299-cc7e-4fed-b636-07fbd7a3cf7d', 'weight': 50}]", "col_type": "sflow", "lb_algo": "2" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " similar record name exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

15.5 Delete SFlow CollectorSet

Deletes the sflow collectorset based on sflowcolset_id provided.

URL	/v1.0/secmon/collectorset/<sflowcolset_id>/
------------	---

	Example: /v1.0/secmon/collectorset/36d59412-48d0-41ba-bbb3-0be168ed371e/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }

16. SFlow Association

SFlow Association associates sflow collector/collectorset with the traffic to be monitored and also associates policy with it.

Parameter	Type	Value	Description
collector_id	String	Collector id in “ ”	The Collector ID for which this association belongs to.
direction	String	INGRESS/EGRESS/BOTH in “ ”	The traffic direction.
policy_id	String	Policy id in “ ”	The id of the policy.
scope_id	String	Scope id in “ ”	Id of the scope.

16.1 List SFlow Association

Returns JSON data for all sflow associations.

URL	/v1.0/secmon/sflowassociation/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "a9703b49-5c7d-465d-9ae1-3289dca06049", "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

16.2 Show SFlow Association

Returns JSON data for sflow association based on sflowassociation_id provided.

URL	/v1.0/secmon/sflowassociation/<sflowassociation_id> Example: /v1.0/secmon/sflowassociation/a9703b49-5c7d-465d-9ae1-3289dca06049/
Method	GET
URL Params	None
Data Params	None

Success Response	Code: 200 OK Content: { "id": "a9703b49-5c7d-465d-9ae1-3289dca06049", "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	code: 500 INTERNAL SERVER ERROR or code: 404 NOT FOUND content: { "detail": "Resource not found" }

16.3 Create SFlow Association

Returns JSON data for created sflow association.

URL	/v1.0/secmon/sflowassociation/
Method	POST
URL Params	None
Data Params	{ "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "EGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }

Success Response	Code: 201 CREATED Content: { "id": "a9703b49-5c7d-465d-9ae1-3289dca06049", "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "INGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }
Error Response	Code: 500 INTERNAL SERVER ERROR or code: 400 BAD REQUEST content: { "detail": "Resource already exist" } or code: 400 BAD REQUEST content: { "detail": " record with same scope and collector already exist in association" }

16.4 Update SFlow Association

Returns JSON data for updated sflow association based on sflowassociation_id provided.

URL	/v1.0/secmon/sflowassociation/<sflowassociation_id>/ Example: /v1.0/secmon/sflowassociation/a9703b49-5c7d-465d-9ae1-3289dca06049/
Method	PUT
URL Params	None
Data Params	{ "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "EGRESS", }

	<pre>"policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }</pre>
Success Response	<p>Code: 200 OK</p> <p>Content: {</p> <pre>"id": "a9703b49-5c7d-465d-9ae1-3289dca06049", "collector_id": "Collector:4a9d9416-17b6-40b1-9c71-fd550f60efff", "direction": "EGRESS", "policy_id": "9b09cae0-43df-48c4-9b8b-2dc9b6ba423c", "scope_id": "5c0b3c6d-35ed-4be2-9911-0b5a634a514b" }</pre>
Error Response	<p>Code: 500 INTERNAL SERVER ERROR</p> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "Resource already exist" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": " record with same scope and collector already exist in association" }</pre> <p>or</p> <p>code: 400 BAD REQUEST</p> <p>content: {</p> <pre>"detail": "\"id\" update not permitted" }</pre>

16.5 Delete SFlow Association

Deletes the sflow association based on sflowassociation_id provided.

URL	/v1.0/secmon/sflowassociation/<sflowassociation_id>/ Example: /v1.0/secmon/sflowassociation/a9703b49-5c7d-465d-9ae1-3289dca06049/
Method	DELETE
URL Params	None
Data Params	None
Success Response	Code: 204 NO CONTENT Content: None
Error Response	Code: 500 INTERNAL SERVER ERROR or Code: 500INTERNAL SERVER ERROR Content: { "detail": "Resource cannot be deleted as dependency exist" }