API Cross-Site Request Forgery Prevention

Table 1: Feature History

Feature Name	Release Information	Description
API Cross-Site Request Forgery Prevention	Cisco IOS XE SD-WAN Release 16.12.1b Cisco SD-WAN Release 19.2.1	This feature adds protection against Cross-Site Request Forgery (CSRF) that occurs when using Cisco SD-WAN REST APIs. This protection is provided by including a CSRF token with API requests. You can put requests on an allowed list so that they do not require protection if needed.

Cisco SD-WAN REST API Token-Based Authentication

Cisco SD-WAN release 19.2 offers token-based authentication when you use the SD-WAN REST API. This protection is provided by requiring that a token be included with API requests. Each API session uses a unique token that is valid throughout the session. If an API request does not include this token, vManage rejects the request, unless the endpoint is included on an allowed list. (For assistance with adding endpoints to an allowed list, open a case with the Cisco TAC or escalation support team.)

Token Use

The following sections describe how the token is used with the API when you use API docs or third party applications.

API Docs

Cisco vManage automatically generates a token and appends the token to every request that you send from the vManage API Docs page. This process requires no action from you, and you will not notice any difference from previous releases in how the API Docs page operates.

If there are API requests that you want to exclude from this token-based authentication, you can request that these API endpoints be included in an allowed list by opening a case with the Cisco TAC or escalation support team.

API Behaviour Scenarios

Following are the API behaviours currently supported:

1. You can login through an API request and continue using the same session until it expires. A session will expire after 30 minutes of inactivity or after 24 hours which is the total lifesapn of a session.
2. You can login through an API request and after each API call do a logout. While performing the logout make sure you have the XSRF token in the http header.

   The logout action has been explained with an example in the Third Party Application Users section below.

Third Party Application Users

If you use scripts or a third party application such as Postman, LiveAction, SolarWinds, or SevOne for vManage API requests, each request must include the token, unless the API is included in and allowed list. If an API request does not include a token and is not included in the allowed list, vManage rejects the request and returns the response code 403 (forbidden) with the message "SessionTokenFilter: Token provided via HTTP Header does not match the token generated by the server."

To request that certain API endpoints be included in an allowed list, open a case with the Cisco TAC or escalation support team

To include the token in a third party API request, follow these steps.

1. After logging in to vManage, obtain a token by making a request to the following URL, where *vManage_IP* is the IP address of your vManage server. You can obtain a token in string format or in JSON format.

   To obtain a token in string format:

   https://*vManage_IP*/dataservice/client/token

   To obtain a token in JSON format (beginning with Cisco IOS XE SD-WAN Release 16.12 and Cisco SD-WAN Release 19.2):

   https://*vManage_IP*/dataservice/client/token?json=true

   The token that these calls return is valid for the rest of your current session. The following example shows requests for obtaining a token:

   Command for obtaining a token in string format:

   ```
   sampleuser$ curl --user admin:admin https://vManage_IP/dataservice/client/token --insecure
   ```

   Output in string format:

   ```
   FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C
   ```

Command for obtaining a token in JSON format:

```
sampleuser$ curl --user admin:admin https://vManage_IP/dataservice/client/token?json=true
 --insecure
{"token":"F1E047E444DB2CA4237B0246DFE133345584B788C6E8776F04749A371B73F3C0C683043F1CDBB5E01BBBDA7D6C35F58EA37A"}
```

Output in JSON format:

```
FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C
```

2. In the header of each subsequent API request in the current session, include the key X-XSRF-TOKEN key, with a value that consists of the token that you generated.

   The following examples show a GET request and a POST request that include a generated token in the header:

   Command:

```
sampleuser$ curl "https://vManage_IP/dataservice/server/info" -H "Cookie:
JSESSIONID=pSwrx3AEWokiDO1TkFiOjgSehp-ITNdFn7Xj9PsL.c331d01e-91d7-41cc-ab90-b629c2ae6d97"
--insecure -H "X-XSRF-TOKEN=
FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C"
```

   Output:

```
{"Achitecture":"amd64","Available processors":2}
```

   Command

```
sampleuser$
"https://vManage_IP/dataservice/settings/configuration/emailNotificationSettings" -H
"Cookie:
JSESSIONID=pSwrx3AEWokiDO1TkFiOjgSehp-ITNdFn7Xj9PsL.c331d01e-91d7-41cc-ab90-b629c2ae6d97"
--insecure -H "X-XSRF-TOKEN=
FC5B19BB3521EE20CFBDCD3CEDCC48F50CB1095C9654407936029E9C0EF99FEAE50440B60E49F7CD4A0BAB5307C2855F2E0C"
-X POST --insecure -d
'{"enabled":true,"from_address":"test@mydomain.com","protocol":"smtp","smtp_server":"a.com",
"smtp_port":25,"reply_to_address":"test@test.com","notification_use_smtp_authentication":false}='
```

   Output:

```
{"data":[{"enabled":true,"protocol":"smtp","smtp_server":"a.com","from_address":"test@mydomain.com",
"smtp_port":25,"notification_use_smtp_authentication":false,"reply_to_address":"test@test.com"}]}
```

3. After each API call, you have to do a logout including the TOKEN. This is to ensure no memory gets leaked.

   The following example shows how you can perform this logout:

   Command:

```
sampleuser$ curl "https://10.48.x.y:8443/logout" -b cookies.txt  --insecure -H
"X-XSRF-TOKEN:$TOKEN"
```

   Output: