

Retrieve ESA User Role APIs Using the Privileges API

Contents

[Introduction](#)

[Environment](#)

[Resolution](#)

[Interpret the Response](#)

[Related Content](#)

[Issue](#)

[Cause](#)

[Expected Result](#)

[Resolution](#)

[Procedure](#)

[Example Output](#)

Introduction

This document describes how to retrieve privilege information for the authenticated ESA role using the `/esa/api/v2.0/login/privileges` endpoint.

Environment

- Python 3.x
- Python **requests** library installed
- Email Security Appliance (ESA) with the API service enabled on the management interface and with the API port configured

Resolution

1. Confirm that the ESA API service is enabled on the management interface.
2. Confirm the configured API port on the ESA and use the same port value in the script.
3. Obtain an **Authorization** header value and a valid ESA session cookie for the account used to call the API.
4. If needed, install the required dependency:

```
pip install requests
```

5. Update the script variable ESA IP address, API port, and authentication values), then run the script.

 **Note:** Use the API port configured on the ESA. Keep the port value consistent across the script and any screenshots.

Interpret the Response

The **/login/privileges** response represents the privileges granted to the authenticated session. Use the returned privilege list to determine which API resources and operations are available for the current user role.

In the JSON payload, review **data.privileges**. Each entry lists a **resource** path and the allowed HTTP **methods** for that resource.

Related Content

2) Install **requests** (it is not installed with Python by default). Run this command from a command prompt:

```
pip install requests
```

requests allows sending HTTP requests from Python.

3) **pprint** is included with Python (standard library). No installation is required.

The pprint module provides a capability to **pretty-print** arbitrary Python data structures in a form that can be used as input to the interpreter and display them in a readable format. This module is optional. The script can use print instead; however, the output is less readable.

JSON does **NOT** need to be installed separately because it is part of the standard library.

Issue

Determine which Cisco Email Security Appliance (ESA) APIs are accessible to the authenticated user role.

Cause

API access on the ESA is controlled by the privileges assigned to the authenticated session and role. The

/esa/api/v2.0/login/privileges endpoint returns the effective privilege list so you can determine which API resources and HTTP methods are permitted for the current user.

Expected Result

A successful **GET** request to **/esa/api/v2.0/login/privileges** returns HTTP **200 OK** and a JSON payload that lists the privileges (API access) available to the authenticated session. Review the returned JSON to identify the allowed API resources and operations for that role.

Resolution

Procedure

```
import requests
import json
import pprint

url = "https://192.0.2.10:6080/esa/api/v2.0/login/privileges"
headers = {
    "Authorization": "Basic <BASE64_BASIC_AUTH>",
    "Cookie": "<SESSION_COOKIE>"
}

response = requests.request("GET", url, headers=headers, verify=True)
json_string = json.loads(response.content)
pprint.pprint(json_string)
```

Example Output

Example output:

```
{
  "data": {
    "privileges": [
      {
        "resource": "/esa/api/v2.0/login/privileges",
        "methods": ["GET"]
      }
    ]
  }
}
```

Review the **privileges** list to identify which API resources and HTTP methods are permitted for the authenticated session.