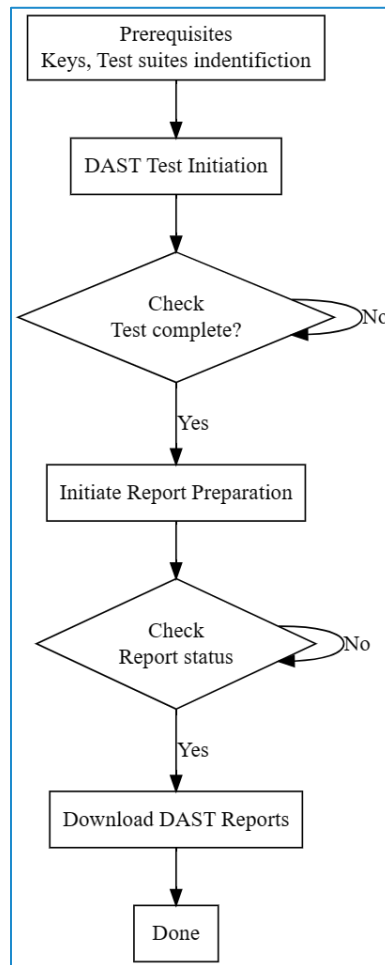




DAST CI/CD Integration

DAST CI/CD Integration.....	1
Dast test flow.....	2
1. Prerequisites:	2
2. Dast Test Initiation:	2
3. Dast Test Monitoring:	2
4. Dast Start Report preparation	3
5. Report Download and Artifact Creation:.....	3
REST API to Initiate a Dast Test.....	3
REST API to Create/update a Dast Policy/configuration.....	4
REST API to Check Dast Test Status:	5
Dast Report Start.....	5
Dast Report Availability	6
Download report APIs.....	6
DAST CI/CD Integration with Jenkins	7
DAST test flow	7
Prerequisites:	8
Steps to initiate DAST scan from Jenkins:	8
Downloading Jenkins Plugin:.....	8
Generating APIKEY and UserKey:.....	8

Dast test flow



1. Prerequisites:

Need configuration values like apikey, org, app, url of dast api, and testConf with include test suites to include and exclude.

2. Dast Test Initiation:

- Sends a POST request to the Dast API to initiate Dast Test with all the required test suites and other configurations.
- The Dast API initiates the Dast test based on the provided configuration.

3. Dast Test Monitoring:

- Use the check dast status Api to retrieve the status of the Dast test.
- Poll the API at regular intervals until the test finishes.

4. Dast Start Report preparation

- Initiate the report preparation
- Check the status of report preparation

5. Report Download and Artifact Creation:

- Upon completion, checks the availability of different Dast reports.
- Wait and check until all the reports are ready
- Downloads the desired reports (API Coverage, Dast Scan, Dast Scan Summary) if available using Download Dast reports API

REST API to Initiate a Dast Test

This API allows you to initiate a Dast test via the AppSentinels REST API.

Method: POST

URL: /api/v1/{org}/{application_id}/dast/test

Replace {application_id} with your actual application ID. **Replace {org}** with your actual organization ID.

Request Body (JSON):

```
{
  "test_config": { //mandatory
    "test_suite_groups": {"Test suite Group name": ["Test suite names"]}, // mandatory
    "target_url": "https://example.com", //mandatory
    "allowed_endpoints": [], //optional
    "auth_config": {} // optional
  }
}
```

Headers:

* apikey: Your API key

* x-user-key: Your user key // Optional

* Content-Type: application/json

Response (JSON):

```
{
  "test_id": "123456",
  "start_time": "2023-10-26T15:00:00Z",
  "status": "IN_PROGRESS",
}
```

```
"status_detail": "Test is currently running."
}
```

Additional Notes:

- Refer to the AppSentinels API documentation for details on additional parameters and error codes.
- A valid API key is required for access.

REST API to Create/update a Dast Policy/configuration

This API allows you to configure a Dast policy via the AppSentinels REST API.

Method: POST

URL: /api/v1/{org}/{application_id}/dast/policy

Replace {application_id} with your actual application ID. **Replace {org}** with your actual organization ID.

Request Body (JSON):

```
{
  "policy_data": {
    "application_config": {},
    "auth_config": {}
  }
}
```

Headers:

- * apikey: Your API key
- * x-user-key: Your user key // Optional
- * Content-Type: application/json

Response (JSON):

```
{
  "message": "Check the request status by calling '/api/dast/v1/req_status/<app  
lication_id>/<request_id>' API after sometime.", "request_id": "cca50480b19e4  
84b9877a48521954b42",
  "status": "pending"
}
```

Additional Notes:

- Refer to the AppSentinels API documentation for details on additional parameters and error codes.
- A valid API key is required for access.

REST API to Check Dast Test Status:

GET /api/v1/{org}/{application_id}/dast/test/{test_id}

Parameters:

- **application_id:** String (required). The ID of the application containing the Dast test.
- **test_id:** String (required). The ID of the Dast test to check the status of.
- **org:** String (required). The ID of the organization containing the application.

Headers:

- **apikey:** Your API key
- **x-user-key:** Your user key // optional
- **Content-Type:** String (optional). Defaults to "application/json".

Response:

```
{
  "test_id": "123456",
  "start_time": "2023-10-26T15:00:00Z",
  "end_time": null,
  "status": "IN_PROGRESS",
  "status_detail": "Test is currently running.",
  "test_cases_executed": 10,
  "application_issue_found": 2,
  "duplicate_alerts": 0,
  "policy_last_update": "2023-10-26T14:00:00Z",
  "training_last_update": "2023-10-25T13:00:00Z",
  "test_suites": ["api-testing"]
}
```

Notes:

- Replace {application_id} and {test_id} with the actual values.
- The response contains details like start time, end time (if finished), status, executed test cases, identified issues, and other information.

Dast Report Start

Method POST

URL: /api/v1/{org}/{application_id}/dast/test/{test_id}/report

Parameters:

- **application_id:** String (required) - The ID of the application containing the Dast test.
- **test_id:** String (required) - The ID of the Dast test to check the report availability for.
- **org:** String (required) - The ID of the organization containing the application.

Headers:

* **apikey:** Your API key

* **x-user-key:** Your user key //optional

* **Content-Type:** application/json

Request Body (JSON):

```
{}
```

Dast Report Availability

Method: GET

URL: /api/v1/{org}/{application_id}/dast/test/{test_id}/report/status

Parameters:

- **application_id:** String (required) - The ID of the application containing the Dast test.
- **test_id:** String (required) - The ID of the Dast test to check the report availability for.
- **org:** String (required) - The ID of the organization containing the application.

Headers:

- **apikey:** Your API key
- **x-user-key:** Your user key //optional
- **Content-Type:** String (optional) - Defaults to "application/json".

Response (JSON):

```
{  
  "available_reports": ["api-coverage", "dast-scan", "dast-scan-summary"]  
}
```

Download report APIs

API Coverage Report:

- **Method:** GET

- URL:
/api/v1/{org}/{application_id}/dast/test/{test_id}/report?report_name=dast_api_coverage

Dast Scan Report:

- Method: GET
- URL:
/api/v1/{org}/{application_id}/dast/test/{test_id}/report?report_name=dast_scan

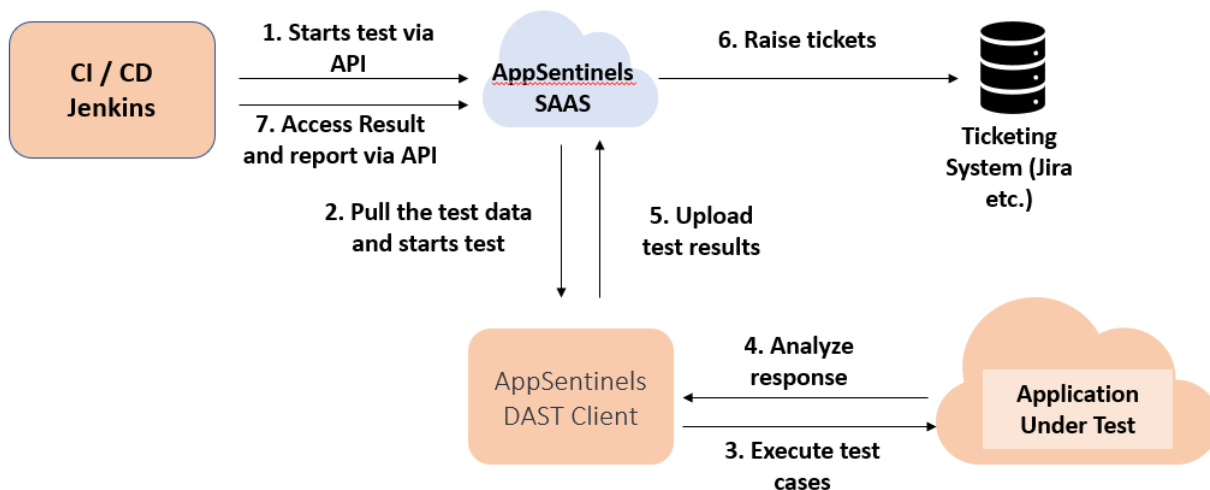
Dast Scan Summary Report:

- Method: GET
- URL:
/api/v1/{org}/{application_id}/dast/test/{test_id}/report?report_name=dast_scan_summary

These APIs allow you to download specific reports based on your needs. Remember to replace the placeholder values with your actual application and test IDs.

DAST CI/CD Integration with Jenkins

DAST test flow



Prerequisites:

- Plugin from AppSentinels needs to be installed in the Jenkins server
- Enable and Configure Email notifications in Jenkins, this will help in getting DAST test reports delivered over email after test completion(optional)

Steps to initiate DAST scan from Jenkins:

- Create a new 'Pipeline' Project from the Jenkins Dashboard.
- In the Configuration page, update the Script section with the script shared by AppSentinels – this script will contain values like APIKEY to authenticate with AppSentinels, DAST scan details and Email forward and report attachment details.
- If you want the job to be executed on a scheduled basis, provide the periodic execution details in the page

Downloading Jenkins Plugin:

AppSentinels Jenkins Plugin can be downloaded from the below link:

https://sample-config.appsentinels.ai/appsentinels-deployment/DAST/jenkins/Appsentinels_Jenkins_Plugin-2.0.hpi

Generating APIKEY and UserKey:

- Login to AppSentinels UI as Org Administrator. Navigate to Settings > API Keys Management page. Create an API Key with a name and allowed IP Addresses.
- Login to the AppSentinels UI as Application user, navigate to Settings > My Settings page. Click on 'Generate Key'. Use this key as UserKey in the API calls (UserKey is optional in the API calls)

For more datils, refer to the link below: <https://docs.appsentinels.ai/> -> Organization -> API Key Management

