

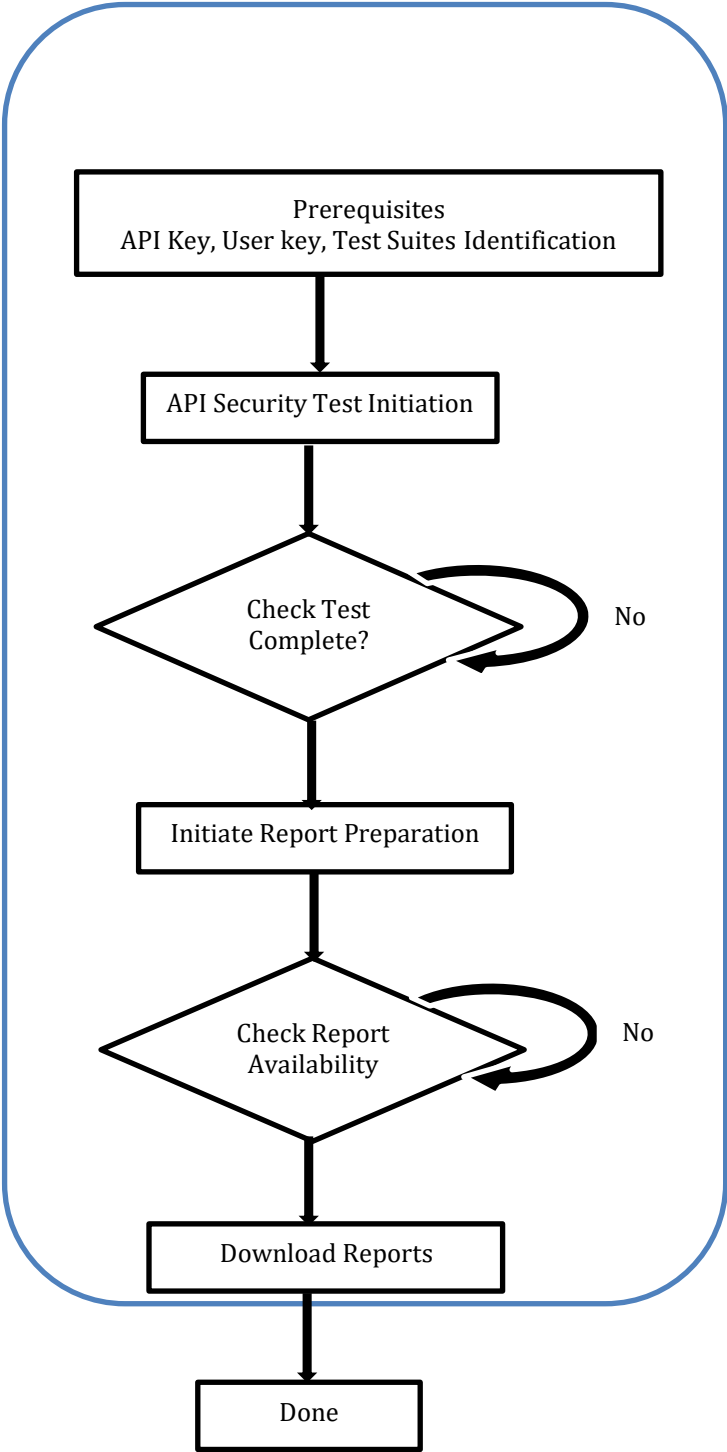
DAST CI/CD Integration

Contents:

DAST CI/CD Integration.....	1
Test flow	2
1. Prerequisites:	3
2. Test Initiation:.....	3
3. Authentication and Authorization for API Calls:.....	3
4. Test Monitoring:.....	3
5. Start Report preparation	3
6. Report Download and Artifact Creation:.....	4
7. REST API to Initiate a Test	4
7.1 Example API call for Test run on an API Group	6
7.2 Example API call for Test run on one API and BUA Test Suite	6
8. REST API to Check Test Status	7
8.1 Example API call to check the status of Test run	7
9. Generate Report	8
9.1 Example API call to generate reports for Test run	8
10. Verify Reports Availability	9
10.1 Example API call to check Reports availability	9
11. Download Reports as Json.....	10
11.1 Example API call to download Reports as Json	10
CI/CD Integration with Jenkins.....	11
Test flow.....	11
Generating APIKEY and User-Key:	11
Appendix A	12
List of Test Suites.....	12

Test

flow



1. Prerequisites:

Need configuration values like apikey, userkey, org-name, application-name, AppSentinels Server-url and Test-config which mentions test suites, apis part of the test run

Configuration Keys	Description
Org-name	Organization name as in AppSentinels UI
Application-name	Application created under the Org, on this application tests will be executed
API Key	Generate the API Key from AppSentinels UI, this will be used for Authentication
User Key	Generate the User Key from AppSentinels UI, from the application on which Tests are being executed. This will be used for Authentication
AppSentinels Server URL	AppSentinels Server URL, to call the APIs

2. Test Initiation:

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

3. Authentication and Authorization for API Calls:

API authentication is performed using the provided API Key and User Key. These keys must be generated from the AppSentinels UI. The API Key is created at the organization level, while the User Key is unique to each user. API execution is controlled based on the user's assigned access privileges

4. Test Monitoring:

- Use the [check status API](#) to retrieve the status of the test run.
- Poll the API at regular intervals until the test is finished.

5. Start Report preparation

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

6. Report Download and Artifact Creation:

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

Report Name	Description
API Coverage Report	Detailed report which covers api level testcases and its status along with brief description about the testcase
Scan Report	This report will contain the description about the vulnerabilities observed in the test run
Scan Summary Report	Summary of All Vulnerabilities observed in the Test run
Auth Coverage Report	Report will be generated only in BOLA test runs, this will contain details about api relationships, tested and excluded relationships between the apis

7. REST API to Initiate a Test

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

Method: *POST*

URL: */api/v1/{org}/{application_name}/dast/test*

Parameters:

Parameter Name	Location	Description	Mandatory(M)/ Optional(O)
org	Path	The name of the organization containing the application	M
application_name	Path	The name of the application	M
apikey	Header	Generated org level api key	M
x-user-key	Header	Generated User's user key	M
Content-type	Header	Json type api(application/json)	O
test_config	Body	Provide the config values for the test run	M

Request Body (JSON):

```
{
  "test_config": {
    "test_case_groups": [
      "intrusive_testcases"
    ],
    "api_filters": {
      "allowed_api_groups": [
        "grp1"
      ],
      "allowed_endpoints": [
        "POST /api/v1/test"
      ],
      "blocked_api_groups": [],
      "blocked_endpoints": []
    }
  },
  "test_suites": [
    "Broken User Authentication"
  ]
}
```

Note: These parameters can be used in the api call based on the need. Only *test_config* is mandatory body payload in the api

Response (JSON):

```
{
  "position_in_queue": 1,
  "status": "Test Queued",
  "test_id": "1234-56789"
}
```

7.1 Example API call for Test run on an API Group

```
curl -X 'POST' \
'https://preprod.int.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test' \
-H 'accept: application/json' \
-H 'timezone: Asia/Calcutta' \
-H 'apikey: <api-key>' \
-H 'x-user-key: <user-key>' \
-H 'Content-Type: application/json' \
-d '{
  "test_config": {
    "api_filters": {
      "allowed_api_groups": [
        "cards-apis"
      ]
    }
  }
}'
```

7.2 Example API call for Test run on one API and BUA Test Suite

```
curl -X 'POST' \
'https://preprod.int.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test' \
-H 'accept: application/json' \
-H 'timezone: Asia/Calcutta' \
-H 'apikey: <api-key>' \
-H 'x-user-key: <user-key>' \
-H 'Content-Type: application/json' \
-d '{
  "test_config": {
    "api_filters": {
      "allowed_endpoints": [
        "POST /api/v1/test/data"
      ]
    }
  },
  "test_suites": [
    "Broken User Authentication"
  ]
}'
```

8. REST API to Check Test Status

Method: *GET*

URI: */api/v1/{org}/{application_name}/dast/test/{test_id}*

Parameters:

Parameter Name	Location	Description	Mandatory(M)/Optional(O)
org	Path	The name of the organization containing the application	M
application_name	Path	The name of the application	M
apikey	Header	Generated org level api key	M
x-user-key	Header	Generated User's user key	M
Content-type	Header	Json type api(application/json)	O
Test_id	query	Executes Test's test-id	M

8.1 Example API call to check the status of Test run

```
curl -X 'GET' \ 'https://preprod.int.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test/1234-56789' \  
-H 'accept: application/json' \  
-H 'apikey: <api-key>' \  
-H 'x-user-key: <user-key>'
```

Notes:

- The response contains details like start time, end time (if finished), status, executed test cases, identified issues, and other information.

9. Generate Report

Method *POST*

URL: */api/v1/{org}/{application_name}/dast/test/{test_id}/report?regenerate_report=false*

Parameters:

Parameter Name	Location	Description	Mandatory(M)/Optional(O)
org	Path	The name of the organization containing the application	M
application_name	Path	The name of the application	M
apikey	Header	Generated org level api key	M
x-user-key	Header	Generated User's user key	M
Content-type	Header	Json type api(application/json)	O
test_id	query	Executes Test's test-id	M
regenerate_report	Query	Pass as true if you want to regenerate the already generated report	O

9.1 Example API call to generate reports for Test run

```
curl -X 'POST' \
  'https://preprod.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test/1234-56789/report?&regenerate_report=false' \
  -H 'accept: application/json' \
  -H 'apikey: <api-key>' \
  -H 'x-user-key: <user-key>' \
  -d ''
```

10. Verify Reports Availability

Method: *GET*

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

Parameters:

Parameter Name	Location	Description	Mandatory(M)/Optional(O)
org	Path	The name of the organization containing the application	M
application_name	Path	The name of the application	M
apikey	Header	Generated org level api key	M
x-user-key	Header	Generated User’s user key	M
Content-type	Header	Json type api(application/json)	O
test_id	path	Executes Test’s test-id	M

10.1 Example API call to check Reports availability

```
curl -X 'GET' \
'https://preprod.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test/1234-56789/report/status' \
-H 'accept: application/json' \
-H 'apikey: <api-key>' \
-H 'x-user-key: <user-key>'
```

Response (JSON):

```
{
  "dast_api_coverage": "available",
  "dast_scan": "available",
  "dast_scan_summary": "available"
}
```

11. Download Reports as Json

Method: *GET*

URL:

/api/v1/{org}/{application_name}/dast/test/{test_id}/resultdata?report_name=dast_api_coverage

Parameters:

Parameter Name	Location	Description	Mandatory(M)/Optional(O)
org	Path	The name of the organization containing the application	M
application_name	Path	The name of the application	M
apikey	Header	Generated org level api key	M
x-user-key	Header	Generated User's user key	M
Content-type	Header	Json type api(application/json)	O
test_id	path	Executes Test's test-id	M
report_name	Query	Name of the report	M

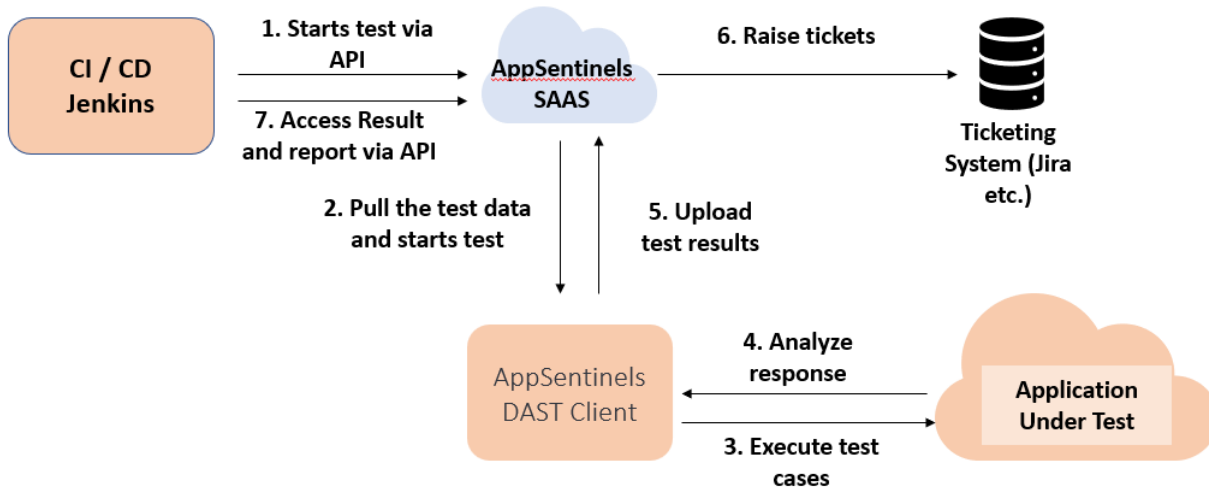
11.1 Example API call to download Reports as Json

```
curl -X 'GET' \
'https://preprod.appsentinels.ai/api/v1/testorg/testorg_app1/dast/test/1234-56789/resultdata?report_name=dast_scan' \
-H 'accept: application/json' \
-H 'apikey: <api-key>' \
-H 'x-user-key: <user-key>'
```

Name of the reports: dast_scan, dast_scan_summary, dast_api_coverage

CI/CD Integration with Jenkins

Test flow



- If you want the job to be executed on a scheduled basis, provide the periodic execution details in the page

Generating APIKEY and User-Key:

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

Appendix A

List of Test Suites

Sl. No	Test Suite Names
1	API Call Coverage Test
2	API Hygiene, Misconfiguration and Governance
3	Broken Function Level Authorization
4	Broken Object Level Authorization
5	Broken User Authentication
6	Command Injection
7	Cross Site Scripting Attack
8	Delete Attack
9	Denial of Service Attack
10	HTTP Tests
11	LDAP Injection
12	Local File Inclusion
13	Mass Assignment
14	NoSQL Injection
15	Open Redirect
16	Persistent Injection Response Validator
17	Resource Limits Check

18	SQL Injection
19	Server Side Request Forgery
20	Stateful API Fuzzer
21	XML External Entity Injection

Document History

Revision	Date Modified	Author	Comments
1.0	24-Nov-2025		Initial Draft
1.1	25-Nov-2025		Reformatted Document
1.2	17-Dec-2025		Reformatted Document