

OpenText™ Fortify Remediation Aviator

User Guide

Version 26.4

Table of Contents

User Guide	3
1 Change log	3
2 Introduction	6
2.1 Product name changes	7
2.2 Fortify Remediation Aviator model	7
2.3 Fortify CLI capabilities	8
2.4 Run connection diagnostics	8
2.5 User roles	9
2.6 Entitlement model	9
2.6.1 Limitations	10
2.7 Limitations (OpenText™ Fortify on Demand)	12
2.8 Authentication model	12
2.9 Audit tag mapping	13
2.10 Related documents	15
3 Supported languages and vulnerability categories	18
4 Getting started	21
4.1 Download fcli	21
4.2 Register the customer administrator	21
4.3 Generate tokens for individual users	21
4.4 Create application	23
4.5 Apply Fortify Remediation Aviator tags	24
4.6 Correlation of OpenText Fortify SAST and OpenText DAST results	24
4.7 Trigger SAST audit	26
4.8 Trigger DAST audit	31
4.9 Perform bulk SAST audit	33
4.10 Perform bulk DAST audit	37
4.11 Perform bulk correlation	41
4.12 Perform auto-remediation	42
5 Manage tenant information	49
5.1 Manage administrator configurations	49
5.2 Manage user tokens	50
5.3 Manage applications	51
5.4 Manage entitlements	53
6 FAQs	53

User Guide

This user guide provides guidance for users to use the OpenText™ Fortify Remediation Aviator .

This document includes the following information:

- Key concepts of Fortify Remediation Aviator ([Introduction](#))
- Downloading and using the Fortify Remediation Aviator ([Getting started](#))
- List of administrator customer tasks that you can perform using Fortify Remediation Aviator ([Manage tenant information](#))
- List of queries for the customer administrator and user ([FAQs](#))

1. Change log

The following table lists changes made to this document. Revisions to this document are published between software releases only if the changes made affect product functionality.

Software Release / Document Version	Changes
26.4.0	Added: • Trigger DAST audit • Perform bulk DAST audit • Run connection diagnostics • Added category combinations for DAST that support automatic suppression (see Supported languages and vulnerability categories) Updated: • All categories for HTML, Ruby, Solidity, Bicep, Terraform, YAML, JSON, Dart, Flutter, Objective-C++, and Objective-C are now supported with automatic suppression (see Supported languages and vulnerability categories) • Perform auto-remediation to support local FPR input, issue filtering, preview mode, source-file encoding handling, deterministic application, and explicit remediation outcomes (see Perform auto-remediation) • Quota behavior to preserve application quota when applications are deleted and later reactivated (see Limitations) • Required OpenText Application Security permissions for trigger audit (see Trigger audit) • <code>audit</code> command has been updated for SAST scan (see Trigger audit)

26.3.0	Added: • Configuration of selected OpenText Application Security applications to consume multiple entitlements (see Limitations) • Support to prevent auto-suppression for selected categories (see Audit tag mapping) • Correlation of OpenText SAST and OpenText DAST results • Perform bulk correlation • New command to add entitlement (see Manage applications) • New commands to manage OpenText Fortify SAST and OpenText Fortify DAST entitlements (see Manage entitlements) Updated: • SAST-DAST correlation description (see Introduction) • <code>tag mapping file</code> to include suppression exclusions (see Audit tag mapping) • Increased Initial Quota (see Limitations and Limitations (OpenText™ Fortify on Demand)) Removed: • Entitlement list command has been deprecated.
26.2.0	Added: • Support to customize the default folder prioritization order using command-line option or configuration based on the required priority when there is limited quota (see Trigger audit) • Options to prevent automatic consumption of quota on application versions that have more open issues than available quota for both individual and bulk audits (see Trigger audit and Perform bulk audit) • Option to select the mapping mode (see Perform bulk audit) • New options to apply auto-remediations for OpenText™ Application Security FPR (see Perform auto-remediation) • Option to apply auto-remediation for OpenText™ Fortify on Demand FPR (see Perform auto-remediation) Updated: • All categories for Apex, PHP, C/C++, ABAP, Scala, Swift, and PLSQL are now supported with automatic suppression (see Supported languages and vulnerability categories)

26.1.0	Added: - New options to refresh the application version metrics and to set the refresh timeout while auditing (see Trigger audit) Updated: - All categories for JavaScript, TypeScript, Python, Go, and Kotlin are now supported with automatic suppression (see Supported languages and vulnerability categories)
--------	--

2. Introduction

OpenText™ Fortify Remediation Aviator is a cloud-based enterprise service that audits the OpenText SAST or DAST scan results as a true positive or a false positive.

Fortify Remediation Aviator leverages Large Language Model (LLM) technology. When an issue is classified as a true positive, Fortify Remediation Aviator offers remediation recommendations, enabling users to resolve code issues quickly and accurately. It also supports auto-remediation and SAST-DAST correlation. SAST-DAST correlation uses AI to match OpenText Fortify SAST findings with OpenText Fortify DAST findings for the same application that refer to the same underlying vulnerability, giving security teams stronger, cross-validated evidence for each issue.

Fortify Remediation Aviator is accessible using OpenText SAST in an off-cloud setup and OpenText SAST through the Fortify Hosted model. In both cases, Fortify Remediation Aviator itself is a SaaS service. Regardless of the access method, you can use the open-source Fortify CLI tool to transmit scan results from the OpenText™ Application Security (Fortify Software Security Center) to Fortify Remediation Aviator for processing. The results from Fortify Remediation Aviator are stored as audit information in OpenText Application Security. Fortify Remediation Aviator is also available through OpenText Core Application Security (Fortify on Demand).

2.1. Product name changes

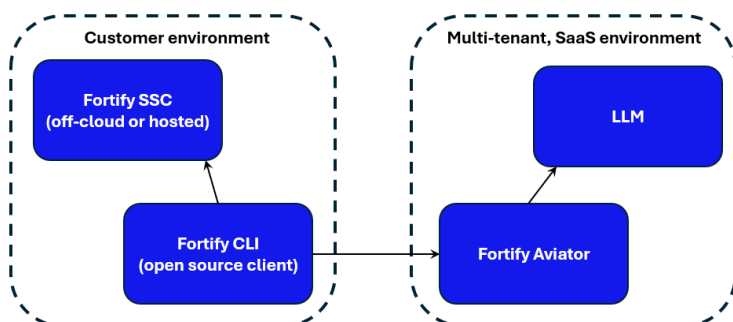
OpenText is in the process of changing the following product names:

Previous name	New name
OpenText™ Static Application Security Testing (OpenText SAST)	OpenText™ Fortify Static Application Security Testing (OpenText Fortify SAST)
Fortify Software Security Center	OpenText™ Application Security
Fortify WebInspect	OpenText™ Fortify Dynamic Application Security Testing (OpenText Fortify DAST)
OpenText™ Core Application Security	OpenText™ Fortify on Demand
OpenText™ Core Software Composition Analysis (OpenText Core SCA)	OpenText™ Fortify Software Composition Analysis (OpenText Fortify SCA)
OpenText™ Application Security Aviator	OpenText™ Fortify Remediation Aviator
Fortify Applications and Tools	OpenText™ Application Security Tools

The product names have changed on product splash pages, mastheads, login pages, and other places where the product is identified. The name changes are intended to clarify product functionality and to better align the Fortify Software products with OpenText. In some cases, such as on the documentation title page, the old name might temporarily be included in parenthesis. You can expect to see more changes in future product releases.

2.2. Fortify Remediation Aviator model

The Fortify CLI tool sends the SAST or DAST scan results, including vulnerability information and source code snippets, from the OpenText Application Security to Fortify Remediation Aviator. The Fortify Remediation Aviator sends the scan results to an LLM for auditing the scan results. The results are then returned to Fortify CLI. The vulnerability and audit information are not stored within Fortify Remediation Aviator. Fortify Remediation Aviator retains only the statistical data indicating that an audit occurred, including the number of issues identified.



Advantages of Fortify Remediation Aviator

Fortify Remediation Aviator leverages Generative Artificial Intelligence (GenAI) in the form of a Large Language Model (LLM) to generate content and provide product functionality.

- **Hybrid model:** The Fortify Remediation Aviator model enables you to use the cloud service hosted by OpenText and the existing Fortify CLI utility instead of hosting a heavy LLM.

- **Integration with OpenText Application Security:** After a regular SAST scan is performed, the results are uploaded to the OpenText Application Security. Subsequently, Fortify Remediation Aviator audits the scan results stored in OpenText Application Security. This approach prevents redundant evaluations.

2.3. Fortify CLI capabilities

In addition to auditing the SAST and DAST results using FCLI, you can perform the following operations:

- Apply the remediations proposed by Fortify Remediation Aviator to the source code automatically.
- Correlate SAST and DAST findings for an application version to identify the same underlying vulnerability across both scan types. For more information, see [Correlation of OpenText SAST and OpenText DAST results](#).
- Run Fortify Remediation Aviator connection diagnostics to troubleshoot endpoint configuration, DNS resolution, TCP connectivity, TLS handshakes, gRPC request and response reachability, and optional token validation. For more information, see [Run connection diagnostics](#).
- View entitlements and entitlement consumption. For more information, see [Entitlement model](#).
- Create new applications and view available applications.
- Create access tokens for individual users.
- List and revoke the access tokens.

2.4. Run connection diagnostics

Use the Fortify Remediation Aviator connection diagnostics command to diagnose Fortify Remediation Aviator endpoint and credential connectivity.

Runs ordered connectivity checks for Fortify Remediation Aviator endpoint configuration, DNS, TCP, optional HTTP proxy CONNECT (when a proxy is configured), TLS, gRPC, reachability, and optional credential validation and reports the result of each stage independently. Use `--url` with optional `--token`, or a saved `--aviator-session` / `--admin-config`.

```
fcli aviator connectivity diagnose --av-session <aviatorSession>| --admin-config <adminConfig>| --url <url> [-t <source>]
```

Optional parameters for command option

Optional argument	Description	Default value
<code>--timeout</code>	Timeout in seconds for TCP, proxy CONNECT, TLS, and gRPC diagnostic stages.	30

The diagnostics output includes a status indicator such as `PASS`, `FAIL`, or `WARN`, along with a brief explanation of what each stage validated. If a required stage fails, the command exits with a non-zero exit code and provides guidance on the next steps for that stage.

Diagnostic stages

The diagnostics flow checks the following stages in order:

1. Fortify Remediation Aviator endpoint configuration validation
Confirms that the Fortify Remediation Aviator endpoint configuration is available and can be used for the connection attempt. If this stage fails, verify the configured Fortify Remediation Aviator URL and the FCLI Fortify Remediation Aviator session settings.
2. DNS resolution
Confirms that the Fortify Remediation Aviator host name resolves successfully. If this stage fails, verify DNS configuration, hostname spelling, and proxy name-resolution requirements.
3. TCP connectivity
Confirms that FCLI can open a TCP connection to the Fortify Remediation Aviator endpoint. If this stage fails, verify firewall, routing, port, and proxy connectivity settings.

4. TLS handshake
Confirms that FCLI can complete the TLS handshake. If this stage fails, verify certificate trust, TLS inspection configuration, and proxy TLS settings.
5. gRPC request and response reachability
Confirms that FCLI can send a gRPC request and receive a response from Fortify Remediation Aviator. If TLS succeeds but no gRPC response is received within the timeout period, the diagnostics output explicitly reports that TLS was established but the gRPC call timed out or did not receive a response. This typically indicates a proxy, HTTP/2 forwarding, gRPC forwarding, or upstream response-path issue.
6. Token validation
Optionally confirms that the configured token is valid. If this stage fails, verify the token value, token source, and token expiration.

Review diagnostics output

Review the result of each stage independently. A later stage can fail even when earlier network stages succeed. For example, a successful TLS handshake confirms only that the encrypted connection was established. It does not confirm that Fortify Remediation Aviator received the gRPC request or that FCLI received a gRPC response.

Use machine-readable output when you need to share diagnostic results with support. Machine-readable output preserves the status, explanation, and guidance associated with each diagnostic check.

2.5. User roles

Fortify Remediation Aviator includes sequential procedures and involves different user roles in your organization. The user roles involved are:

- Customer administrator: Customer administrator can create an admin configuration, create and manage tokens, and manage applications and entitlements.
- Customer user: User can create a user session and trigger an audit.



Note

Before using the fcli, you must be registered with OpenText. OpenText Support creates and registers the tenant upon your initial purchase of Fortify Remediation Aviator.

2.6. Entitlement model

Fortify Remediation Aviator is a paid service. The following entitlement models of Fortify Remediation Aviator models are available for purchase:

- **per Application:** The basic entitlement model is **per Application**. In this model, you can run any number of audits for a single application. Therefore, the customer administrator must first create applications in the tenant. Fortify Remediation Aviator monitors the number of entitlements for each tenant every time a Fortify Project Result (FPR) is processed.
- **per Developer:** In this model, Fortify Remediation Aviator cannot monitor developers. By default, a maximum of four applications are allocated per 10 developers. If this allocation does not meet your requirements, contact your account representative.

For every new purchase of the Fortify Remediation Aviator entitlement, OpenText Support updates the number of entitlements for the respective tenant.

Application and entitlement lifecycle

- Each application consumes one entitlement by default.
- A single application can consume multiple entitlements to increase its quota.
- When an entitlement is removed from an application, its remaining quota is preserved and subject to top-up as per the Monthly Quota

For information about how Fortify Remediation Aviator preserves quota when applications are deleted and later reactivated, see [Limitations](#).

For every new purchase of a Fortify Remediation Aviator entitlement, OpenText Support updates the number of entitlements for the respective tenant.

2.6.1. Limitations

Fortify Remediation Aviator, like any LLM-powered system, has usage limitations due to the operational costs of invoking LLMs. Each application is assigned with a quota that defines the maximum number of issues it can audit. Quotas apply individually to each application and are not shared across applications or tenants. This method provides an efficient way to control audit consumption. When an application reaches its quota, issues will not be audited until the quota is assigned.

Quota consumption

When you run an audit, Fortify Remediation Aviator will audit up to the remaining quota for that application. If the audit is successful, the quota will be deducted based on the consumption. If the audit fails or crashes the quota remains unchanged.



Note

- Only one audit run per application is allowed at a time to ensure quota consistency.
- Quota usage is recorded in the audit logs for tracking.

Quota top-up

Initial Quota

When an application is created, an Initial Quota is credited as defined by the tenant configuration. The default value is 10,000.

Monthly Quota

Every month, applications receive a Monthly Quota top-up. The default value is 1,000. The total quota for an application cannot exceed the Initial Quota.

For example:

- Initial Quota = 10,000 (default)
- Monthly Quota = 1,000 (default)
- Month-1 = 500 issues are consumed and 9,500 are remaining.
- Monthly top-up = 1,000 is the default, but the total quota is limited to the Initial Quota, which is 10,000. Therefore, 500 is added (500 + 9,500).



Note

In some cases, both Initial and Monthly Quotas can be configured based on the tenant. This change affects the future and is not applied retroactively.

Application deactivation and reactivation

When a customer deletes an application that has less than the maximum available quota, the entitlement that the application used retains its remaining quota.

- The entitlement associated with the deleted application becomes available for other applications to consume.
- The available quota of the entitlement is subject to monthly top-up until it reaches 10,000 again.
- When an application consumes an entitlement, the available entitlement with the highest quota is used.



Note

When an application consumes an entitlement with less than the maximum available quota, the customer is notified of the available quota and why the application is not initialized with the maximum quota.

Multi-entitlement applications

You can configure selected applications to consume multiple entitlements. This supports large applications that regularly exceed standard quota limits and avoids creating multiple artificial applications for a single logical application.

The entitlement multiplier is configured on an existing application and defaults to 1.

- Multiplier values are positive integers (1, 2, 3, etc.).
- Each command increment adds one multiplier at a time.
- The multiplier is stored in the application entity and visible in application details.

Quota calculation

- Initial quota = Base initial quota × Entitlement multiplier
- Monthly quota = Base monthly quota × Entitlement multiplier
- Maximum quota cap = Scaled initial quota

For example,

- Multiplier = 2 → Initial Quota = 20,000; Monthly Quota = 2,000
- Multiplier = 3 → Initial Quota = 30,000; Monthly Quota = 3,000

When increasing a multiplier for an existing application, the new quota is calculated as:

`current_quota + (new_multiplier - current_multiplier) × base_initial_quota`

For example, if an application has a base initial quota of 10,000, a multiplier of 2 and a current quota of 2,000, and it is increased to a multiplier of 4:

- New quota = $2,000 + (4 - 2) \times 10,000 = 22,000$
- Maximum quota cap = $4 \times 10,000 = 40,000$

Entitlement tracking and validation

- Each multiplier increment consumes one additional tenant entitlement.
- Configuration fails if insufficient entitlements are available.
- Validation checks entitlement availability before each change.
- Error messages are displayed if limits are exceeded.
- Entitlement consumption and multiplier values appear in application details.



Note

Existing applications remain backward compatible with multiplier 1 unless explicitly updated.

Behavior when quota limit is reached

If an audit reaches the quota limit:

- The number of issues audited is based on the remaining quota. FCLI displays the status and the number of issues audited and skipped.
- The audit stops gracefully.
- For the audited issues, the FPR file is generated and uploaded to OpenText Application Security.
- This case is not treated as an error.

When you run an audit and the quota is completely utilized, FCLI displays the message, *"No quota available for application <application_name>. Your next quota update for this application will happen on <Month Date, Year>."*



Tip

The customer administrator can check the quota information using `fcli aviator app list`.

Efficient way to use the quota

When applications have a large number of issues exceeding the quota assigned and require auditing, use the following methods:

- Use OpenText Fortify SAST scan policy to reduce the number of issues. For information, see *OpenText™ Static Application Security Testing User Guide*.
- Use filter options while auditing:
 - Specify folders of your default filter set (for example, only audit Critical and High issues) to limit the number of issues processed by Fortify Remediation Aviator, or,
 - Define a custom filter set specifically to manage what gets processed by Fortify Remediation Aviator. For more information about the filter options, see [Trigger audit](#).

See also

[FAQs](#)

2.7. Limitations (OpenText™ Fortify on Demand)



Important

The limitation system described below exclusively applies to Fortify Remediation Aviator as available through OpenText™ Fortify on Demand. For the limitation mechanism in Fortify Remediation Aviator for off-cloud and Fortify Hosted customers, see [Limitations](#).

There are limitations on how Fortify Remediation Aviator audits SAST scan results containing an extremely large number of issues.

Fortify Remediation Aviator's approach to auditing data is similar to how auditing by a human works. If there are hundreds of issues, they can be audited manually. But if there are thousands, that's neither practical nor useful. The first response should be to look for patterns. A recurring bad coding pattern may cause many true positives. In that case, the code should be fixed in bulk. Alternatively, some rules in SAST may trigger massive false positives for that particular codebase. In that case, the SAST scan configuration should be adjusted. After these steps, a smaller number of remaining findings can be audited individually.

Fortify Remediation Aviator has largely been designed as an AI-powered version of a human auditor. It will not audit an unlimited number of issues. Practically, such a limit is also necessary, given the non-trivial resource consumption for every issue audited.

The following limits apply:

- For any FPR, Fortify Remediation Aviator audits at most 10,000 new issues in total.
- For any FPR, Fortify Remediation Aviator audits at most 500 new issues in a single category.

When Fortify Remediation Aviator processes an FPR that exceeds one or more of these limits, any issue beyond the limit is skipped. When a subsequent SAST analysis of the same project yields new issues, these new issues are audited if the quota is available.

For example,

- A SAST scan of project X yields 10,500 issues.
- The Fortify Remediation Aviator auditing will audit 10,000 of those, and mark 500 as "Excluded due to Limiting".
- Now, development continues. An additional 100 issues have been found, leading to an FPR with 10,600 issues.
- Fortify Remediation Aviator auditing of this new FPR will:
 - Not audit again the 10,000 previously audited issues.
 - Audit the 600 issues that were previously skipped if quota is available.

2.8. Authentication model

Fortify Remediation Aviator uses a dual authentication model.

- Customer administrators are authenticated using a private key. FCLI signs customer administrator requests with this private key. Fortify Remediation Aviator verifies the signature with the registered public key. New customer administrators and public keys can be registered through OpenText support.
- Regular users are authenticated with an access token. Customer administrators can create these tokens for users. FCLI includes this access token in user requests.

2.9. Audit tag mapping

The Fortify Remediation Aviator algorithm predicts issues to be true positives or false positives and, in some cases, unsure.

To provide audit results to OpenText Application Security, the audit result must be set as a tag value, along with the decision to suppress an issue or not. By default, the Analysis tag is used to store audit results and includes a predefined set of possible values. However as an administrator, you might have customized the available Analysis tag values in OpenText Application Security resulting in deviations from the default values. For these reasons, Fortify Remediation Aviator has the functionality to map its predictions to a provided tag value and also decide when an issue should be suppressed by default in OpenText Application Security. Additionally, it allows configuration of the storage location for audit results, in case a different tag, other than the default Analysis tag is preferred.

To configure this mapping, Fortify Remediation Aviator separates and organizes across two tiers of support. For more information, see [Supported languages and vulnerability categories](#). Considering there are two tiers and three different Fortify Remediation Aviator outcomes (true positive (TP), false positive (FP), unsure), there are six different cases in total. These cases must be mapped to an OpenText Application Security tag value and a suppression status. The following is the default mapping performed by Fortify Remediation Aviator :

Tier	Tier configuration name	Outcome	Analysis tag value	Suppressed
Supported with automatic suppression	tier_1	TP	Exploitable	No
Supported with automatic suppression	tier_1	FP	Not an issue	Yes
Supported with automatic suppression	tier_1	Unsure	Not set	No
Supported without automatic suppression	tier_2	TP	Suspicious	No
Supported without automatic suppression	tier_2	FP	Not an issue	No
Supported without automatic suppression	tier_2	Unsure	Not set	No

To override the default tag mapping, use the `--tag-mapping` argument when you run an audit.

```
fcli aviator ssc audit --av <application_version_name:id> --tag-mapping=<file.yaml>
```

The following tag mapping file is the default one that implements the mapping described in the aforementioned table. Use this mapping file as a basis to configure your required mapping. In addition to changing analysis tag values and suppression status, you can also select a different analysis tag altogether.

```
# Set the SSC tag to use to store Aviator results. Optional.
# If not set, defaults to "87f2364f-dcd4-49e6-861d-f8d3f351686b"
tag_id: "87f2364f-dcd4-49e6-861d-f8d3f351686b"
# Map Aviator results to SSC tag values. "tier_1" are high-confidence cases
# that by default are suppressed automatically. "tier_2" are the remaining issues.
# "value" is a String attribute that maps to a tag value in SSC. It may be omitted.
# "suppress" is a Boolean attribute that defaults to "false"
# Optional: Issues matching these exclusions are never auto-suppressed by Aviator.
# Matching is exact and case-insensitive.
# suppression_exclusions:
#   - categories:
#     - "Privacy Violation"
mapping:
  tier_1:
    fp:
      value: "Not an Issue"
      suppress: true
    tp:
      value: "Exploitable"
      suppress: false
    unsure:
      suppress: false
  tier_2:
    fp:
      value: "Not an Issue"
      suppress: false
    tp:
      value: "Suspicious"
      suppress: false
    unsure:
      suppress: false
```

The tag used to save the audit results can be changed by specifying the tag GUID in the `tag_id` field of the mapping file. To see the available tags and their respective GUID's, run `fcli ssc tag ls` or `fcli ssc custom-tag ls`. The list of GUIDs and their tags is displayed, for example:

Guid	Name	Value type
87f2364f-dcd4-49e6-861d-f8d3f351686b	Analysis	LIST
ACB05E55-E74D-468C-8501-52E1FDC27D71	Auditor Status	LIST
C2D6EC66-CCB3-4FB9-9EE0-0BB02F51008F	Aviator prediction	LIST
FB7B0462-2C2E-46D9-811A-DCC1F3C83051	Aviator status	LIST
831d8692-1b9a-4e6b-bddd-8e0f6552fe3e	Correlation Tag	TEXT
7A3B5C9D-1E2F-4A8B-9C0D-E1F2A3B4C5D6	DAST correlation status	TEXT
013cc66f-8651-4e39-bacb-beb918c5ef65	Expected Analysis	LIST
cc8b7810-2b61-42c6-9ab2-deec19d07234	xv	LIST

Prevent auto-suppression for selected categories

You can prevent Fortify Remediation Aviator from automatically suppressing specific vulnerability categories by defining suppression exclusions in the tag mapping file.

When Fortify Remediation Aviator is about to suppress a finding, it checks the suppression exclusions first:

- If the finding's category matches an exclusion, the finding is not suppressed.
- If there is no match, suppression follows the defined mapping rules.

For example, if you specify categories such as "Privacy Violation" and "SQL Injection," any findings in these categories will not be automatically suppressed, even if suppression is enabled in the mapping. In the tag mapping file, it can be specified as:

```
suppression_exclusions:
```

```
- categories:
```

```
- "Privacy Violation"
```

```
- "SQL Injection"
```

2.10. Related documents

This topic describes documents that provide information about OpenText Application Security Software products.



Note

Most guides are available in both PDF and HTML formats. Product help is available within the Fortify License and Infrastructure Manager (LIM) and the OpenText Fortify DAST.

All products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Product Documentation website for each product.

Document / file name	Description
<i>About OpenText Application Security Software Documentation</i> appsec-docs-n-<version>.pdf	This paper provides information about how to access OpenText Application Security Software product documentation.  Note This document is included only with the product download.
<i>OpenText™ Application Security Software System Requirements</i> appsec-sr-<version>.pdf	This document provides the details about the environments and products supported for this version of OpenText Application Security Software.
<i>What's New in OpenText Application Security Software <version></i> appsec-wn-<version>.pdf	This document describes the new features in OpenText Application Security Software products.
<i>OpenText Application Security Software Release Notes</i> appsec-rn-<i><version></i>.pdf	This document provides an overview of the changes made to OpenText Application Security Software for this release and important information not included elsewhere in the product documentation.

Fortify ScanCentral SAST

The following document provides information about Fortify ScanCentral SAST. This document is available on the Product Documentation website at <https://www.microfocus.com/documentation/fortify-software-security-center>.

Document / file name	Description
<i>OpenText™ Fortify ScanCentral SAST Installation, Configuration, and Usage Guide</i> sc-sast-ugd-<version>.pdf	This document provides information about how to install, configure, and use Fortify ScanCentral SAST to streamline the static code analysis process. It is written for anyone who intends to install, configure, or use Fortify ScanCentral SAST to offload the resource-intensive translation and scanning phases of their OpenText Fortify SAST process.

OpenText Application Security

The following document provides information about OpenText Application Security. This document is available on the Product Documentation website at <https://www.microfocus.com/documentation/fortify-software-security-center>.

Document / file name	Description
<i>OpenText™ Applicaton Security User Guide</i> sssc-ugd-<version>.pdf	This document provides OpenText Application Security users with detailed information about how to deploy and use OpenText Application Security. It provides all the information you need to deploy, configure, and use OpenText Application Security. It is intended for use by system and instance administrators, database administrators (DBAs), enterprise security leads, development team managers, and developers. OpenText Application Security provides security team leads with a high-level overview of the history and status of a project.

OpenText Fortify SAST

The following documents provide information about OpenText SAST (Fortify Static Code Analyzer). Unless otherwise noted, these documents are available on the Product Documentation website at

<https://www.microfocus.com/documentation/fortify-static-code>.

Document / file name	Description
<i>OpenText™ Static Application Security Testing User Guide</i> sast-ugd-<version>.pdf	This document describes how to install and use OpenText Fortify SAST to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.
<i>OpenText™ Static Application Security Testing Custom Rules Guide</i> sast-cr-ugd-<version>.zip	This document provides the information that you need to create custom rules for OpenText Fortify SAST. This guide includes examples that apply rule-writing concepts to real-world security issues.  Note This document is included only with the product download.
<i>OpenText™ Fortify License and Infrastructure Manager Installation and Usage Guide</i> lim-ugd-<version>.pdf	This document describes how to install, configure, and use the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.

OpenText Application Security Tools

The following documents provide information about OpenText Application Security Tools. These documents are available on the Product Documentation website at <https://www.microfocus.com/documentation/fortify-static-code-analyzer-and-tools>.

Document / file name	Description
<i>OpenText™ Application Security Tools Guide</i> sast-tgd-<version>.pdf	This document describes how to install application security tools. It provides an overview of the applications and command-line tools that enable you to scan your code with OpenText Fortify SAST, review analysis results, work with analysis results files, and more.
<i>OpenText™ Fortify Audit Workbench User Guide</i> awb-ugd-<version>.pdf	This document describes how to use Fortify Audit Workbench to scan software projects and audit analysis results. This guide also includes how to integrate with bug trackers, produce reports, and perform collaborative auditing.
<i>OpenText™ Fortify Plugin for Eclipse User Guide</i> ep-udg-<version>.pdf	This document provides information about how to install and use the Fortify Plugin for Eclipse to analyze and audit your code.
<i>OpenText™ Fortify Analysis Plugin for IntelliJ IDEA and Android Studio User Guide</i> iap-udg-<version>.pdf	This document describes how to install and use the Fortify Analysis Plugin for IntelliJ IDEA and Android Studio to analyze your code and optionally upload the results to OpenText Application Security.
<i>OpenText™ Fortify Extension for Visual Studio User Guide</i> vse-ugd-<version>.pdf	This document provides information about how to install and use the Fortify Extension for Visual Studio to analyze, audit, and remediate your code to resolve security-related issues in solutions and projects.

3. Supported languages and vulnerability categories

Fortify Remediation Aviator is verified by OpenText to maximize accuracy. The extent to which a particular vulnerability category in a certain programming language is supported by Fortify Remediation Aviator might differ based on the amount of verification and optimization that has already been performed. There are three classes.

Class	Description
Supported with automatic suppression	Cases with a high degree of confidence. By default, Fortify Remediation Aviator performs automatic suppression of false positives.
Supported without automatic suppression	Cases where confidence is yet to be established to the same standard. By default, Fortify Remediation Aviator does not perform automatic suppression.
Not supported	A small set of cases that cannot be handled by Fortify Remediation Aviator.

The underlying LLM used by Fortify Remediation Aviator evolves over time. Because not every LLM version is immediately available in all cloud hosting locations used by Fortify Remediation Aviator, different instances of Fortify Remediation Aviator may use different LLM versions at any point. The LLM version in use determines the classification of cases. Generally, on newer LLMs, more classes can be moved to “automatic suppression”.

The following overview lists how language or category combinations are classified in the current version of Fortify Remediation Aviator for off-cloud and hosted customers.

Supported DAST categories with automatic suppression

The following DAST vulnerability categories are supported with automatic suppression:

- HTTP Verb Tampering
- Often Misused: HTTP Method Override
- JavaScript Hijacking: JSONP
- Setting Manipulation: Character Set
- Web Server Misconfiguration: Insecure Content-Type Setting
- Privacy Violation: HTTP GET
- Web Server Misconfiguration: Unprotected File
- Privacy Violation: Autocomplete
- Insecure Transport
- Often Misused: Login
- JSON Web Token: Missing Protection Claims
- Weak Cryptographic Signature: Insufficient Key Size
- HTML5: CORS Unsafe Methods Allowed
- Weak Cryptographic Hash
- Web Server Misconfiguration: Server Error Message
- Information Discovery: JWT Header Claims
- System Information Leak: Internal IP
- Compliance Failure: Missing Privacy Policy
- Poor Error Handling: Unhandled Exception
- HTML5: Overly Permissive CORS Policy
- Cross-Frame Scripting
- Cross-Site Scripting: Reflected
- Hidden Field
- Session Management: Easy-to-Guess Session Identifier Name
- Information Discovery: Session Token
- Cookie Security: Missing SameSite Attribute
- SQL Injection
- Privacy Violation: Shoulder Surfing
- Web Server Misconfiguration: Weak Authentication
- Web Server Misconfiguration: OPTIONS HTTP Method
- NoSQL Injection: MongoDB
- Dangerous File Inclusion: Remote
- Password Management: Insecure Submission
- Insecure Deployment: Unpatched Application
- Dynamic Code Evaluation: Code Injection
- Dangerous File Inclusion: Local
- Password Management: Weak Password Policy
- API Discovery
- HTML5: Missing Content Security Policy

- Cache Management: Headers
- Cookie Security: HTTPOnly not Set
- Cache Management: Insecure Policy
- HTML5: Form Validation Turned Off
- Web Server Misconfiguration: Directory Listing
- Path Manipulation: Special Characters
- HLI: Detected Libraries : jQuery v1.8.2
- Insecure Transport: HSTS not Set
- Insecure Transport: Weak SSL Cipher
- HLI: Detected Libraries : jQuery UI v1.8.23
- Privacy Violation: Credit Card Number
- Privacy Violation: Social Security Number
- HLI: Detected Libraries : jQuery v1.7.2
- Often Misused: Price-Related Fields
- HLI: Detected Libraries : Ext JS v4.1.1.1
- Cross-Site Request Forgery
- Session Management: Insufficient Session ID Entropy
- HLI: Detected Libraries : jQuery v1.6.4
- Web Server Misconfiguration: HTTP Basic Authentication
- Open Redirect
- Session Management: Insufficient Session ID Length
- HTML5: Deprecated Header
- HLI: Detected Libraries : Moment.js v2.29.1
- HLI: Detected Libraries : core-js vcore-js-pure@2.6.11; core-js-pure@2.6.11
- HLI: Detected Libraries : jQuery v3.5.1
- HLI: Detected Libraries : React
- HLI: Detected Libraries : Bootstrap v4.5.3

Supported language or category combinations with automatic suppression

- Java, JavaScript, TypeScript, Python, Go, Kotlin, Apex, PHP, C/C++, ABAP, Scala, Swift, PL/SQL, HTML, Ruby, Solidity, Bicep, Terraform, YAML, JSON, Dart, Flutter, Objective-C++, Objective-C
 - All categories except explicitly non-supported ones.
- .NET
 - All categories except Code Correctness, Dead Code, Poor Error Handling, and Unsafe Native Invoke.



Note

The categories Code Correctness, Dead Code, Poor Error Handling, and Unsafe Native Invoke are "supported without automatic suppression".

Supported without automatic suppression

- All other language or category combinations supported by OpenText SAST, except explicitly excluded cases.

Not supported

- The following vulnerability category is explicitly not-supported:
 - Privilege Management: Unnecessary Permission.

**Note**

The verification of this category's issues requires access to the complete source code at once, which is not compatible with the way Fortify Remediation Aviator functions.

4. Getting started

Perform the following steps to get started with Fortify Remediation Aviator:

1. [Download fcli](#)
2. [Register customer administrator](#)
3. [Generate tokens for individual users](#)
4. [Create application](#)
5. [Apply Aviator tag](#)
6. [Correlation of OpenText SAST and OpenText DAST results](#)
7. [Trigger SAST audit](#) / [Trigger DAST audit](#) / [Perform bulk SAST audit](#) / [Perform bulk DAST audit](#) / [Perform bulk correlation](#)
8. [Perform auto-remediation](#)

4.1. Download fcli

Download fcli v3.28.0 or later from <https://github.com/fortify/fcli> and extract it into a directory from where you will run the commands.

4.2. Register the customer administrator

Prerequisite

Ensure that the tenant is registered. Contact OpenText Support for details.

Register the customer administrator

To register a customer administrator:

1. Create a 4096-bit RSA key pair in the PEM format using a cryptographic tool available in your organization.
 - (Optional) Use OpenSSL (<https://www.openssl.org/>) to generate the RSA key pair.

1. Generate a 4096-bit RSA private key (private_key.pem):

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -out private_key.pem
```

2. Extract the public key (public_key.pem) from the private key:

```
openssl rsa -in private_key.pem -pubout -out public_key.pem
```

2. Save the private key (private_key.pem) for later use.
3. Share the public key (public_key.pem) and other necessary details with OpenText Support to register the customer administrator.
You will be notified after the registration is complete.

4.3. Generate tokens for individual users

Prerequisite

You must be a registered customer administrator.

Generate tokens for individual users

To create an access token for a specified user:

1. At the command prompt, navigate to the path where fcli has been extracted.
2. (Optional) View the various administrator and user operations available:

```
fcli aviator -h
```

The following table lists the available command options.

Argument	Description
<code>-h</code> , <code>--help</code>	Shows the help message and exits.
<code>admin-config</code>	Manages the Fortify Remediation Aviator administrator configurations (start here).
<code>session</code>	Manages the Fortify Remediation Aviator user sessions (start here).
<code>app</code>	Manages the Fortify Remediation Aviator applications.
<code>connection</code>	Diagnoses Fortify Remediation Aviator connectivity.
<code>entitlement</code>	Manages the Fortify Remediation Aviator entitlements.
<code>ssc</code>	Uses OpenText™ Fortify Remediation Aviator with OpenText™ Application Security.
<code>token</code>	Manages Fortify Remediation Aviator access tokens.



Note

- Use `admin-config` to view entitlement, manage applications, and generate tokens and `session` to audit.
- Ensure to create an `admin-config` before performing administrator tasks and a user `session` before auditing.
- Use `connection` to troubleshoot connectivity issues between FCLI and Fortify Remediation Aviator. It is intended for situations where Fortify Remediation Aviator commands fail due to connectivity-related errors, such as when Fortify Remediation Aviator is unreachable or a connection cannot be established. For more information, see [Run connection diagnostics](#).

3. Create an administrator configuration for interacting with Fortify Remediation Aviator:

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-key  
<path_to_private_key.pem>
```



Note

The `--private-key` can be a file containing the key or the key itself. Ensure that the key is in the PEM format.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

An admin session is created.

4. Generate the user access token.

```
fcli aviator token create --email <user_email_id> --save-token <output_file>
```

Optional arguments	Description	Default value
<code>--name</code>	Specifies the token name.	Derived from <code>--email</code>
<code>--save-token</code>	Saves the generated raw token string to the specified file. By default, the string is in json format.	Not applicable
<code>-o</code> , <code>--output</code>	Specifies the token format. The available formats are csv, table, expr, json, xml, and yaml.	Not applicable
<code>--to-file</code>	Specifies a file to save the output.	Not applicable
<code>--end-date</code>	Specifies the token expiration date in the YYYY-MM-DD format.	30 days from the date of creation



Note

OpenText recommends using the `--save-token` optional argument to ease the user experience when using the access token.

4.4. Create application

Prerequisites

- You must be a registered customer administrator.
- Ensure to have a valid entitlement. See [Entitlement model](#) section.

Create application

To create an application:

1. Create an administrator configuration for interacting with Fortify Remediation Aviator:

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-key <path_to_private_key.pem>
```



Note

The `--private-key` can be a file containing the key or the key itself. Ensure that the key is in the PEM format.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

2. Create an application:

```
fcli aviator app create <aviator_application_name>
```

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

An application is created and assigned to the first available entitlement.

4.5. Apply Fortify Remediation Aviator tags

If the OpenText Application Security application does not have the Fortify Remediation Aviator specific tags applied, apply the Fortify Remediation Aviator tags to the OpenText Application Security filter templates.

Apply tag

To apply Fortify Remediation Aviator specific tags:

1. Log in to your OpenText™ Application Security session:

```
fcli ssc session login --url <ssc_url> -u <user_name> -p <ssc_password>
```

2. Apply the required template. Ensure to specify at least one option:



Note

Applying the Fortify Remediation Aviator tags ensures that the Fortify Remediation Aviator specific custom tags exists in OpenText Application Security and is associated with the specified issue templates and/or application versions. This prevents OpenText Application Security from removing the Fortify Remediation Aviator audit data when processing the uploaded FPR files.

```
fcli aviator ssc prepare --all-avs | --all-issue-templates | --av=<specific_app_VersionNameOrId> | --appversion=<specific_app_VersionNameOrId> | --issue-template=<specific_app_templateNameOrId>
```

Optional argument	Description	Default value
<code>--ssc-session</code>	Name of the OpenText Application Security session.	default



Note

This command may change in a future fcli version to re-use generic tag update functionality that is planned for the `fcli ssc` module.

4.6. Correlation of OpenText Fortify SAST and OpenText DAST results

OpenText Fortify SAST and OpenText Fortify DAST scans of the same application can independently detect the same underlying vulnerability. A OpenText Fortify SAST scan reports the defect anchored to a source code location, while a OpenText Fortify DAST scan reports it anchored to an HTTP request and URL. Fortify Remediation Aviator uses AI-based

SAST-DAST correlation to identify these matching pairs, providing security teams with stronger, cross-validated evidence for each vulnerability.

When you run the correlation command for an application version, Fortify Remediation Aviator downloads both the SAST FPR and the DAST FPR from OpenText Application Security. It groups findings by vulnerability category and uses a two-stage AI process to identify and confirm correlated pairs. Confirmed correlations are reported back to OpenText Application Security and are visible in the issue view for both SAST and DAST findings. Only unsuppressed findings are submitted for correlation, and previously correlated pairs are retained across re-runs.

To correlate multiple application versions in one command, see [Perform bulk correlation](#).

Prerequisites

- You must be a customer user.
- Ensure the following:
 - The application version in OpenText Application Security has both a processed SAST FPR and a processed DAST FPR.
 - Source code is included in the SAST FPR. Do not use the `--disable-source-bundling` option or the `com.fortify.sca.FPRDisableSourceBundling` property during the SAST scan.
 - The application exists in Fortify Remediation Aviator and is linked to the OpenText Application Security application version.
 - Have a valid user access token.
 - Have an active OpenText Application Security session.
 - User session is created.

Correlate OpenText Fortify SAST and OpenText Fortify DAST results

To correlate OpenText Fortify SAST and OpenText Fortify DAST findings:

1. Run the SAST-DAST correlation command:

This command downloads both the SAST FPR and the DAST FPR from the specified OpenText Application Security application version. Fortify Remediation Aviator identifies correlated pairs using AI and uploads the results back to OpenText Application Security.

```
fcli aviator ssc correlate-sast-dast --av <application_name:version> --app <aviator_application_name>
```

Optional argument	Description	Default value
<code>--app</code>	Name of the Fortify Remediation Aviator session to associate with the correlation.	SAST/FPR Build ID of the OpenText Application Security application
<code>--av</code> , <code>--appversion</code>	ID of the application version or <code><application>:<version></code> name.	Not applicable
<code>--ssc-session</code>	Name of the OpenText Application Security session	default

FCLI displays a structured result containing:

- OpenText Application Security application name and version
- Artifact ID of the uploaded DAST FPR
- Status summarizing the outcome. The following are the status values:
 - `CORRELATED` : All submitted SAST findings have at least one confirmed correlated DAST pair.
 - `PARTIALLY_CORRELATED` : Some findings are correlated. This may occur if quota is exhausted before processing all findings.

- **SKIPPED** : No shared vulnerability categories between SAST and DAST findings; no correlation performed.
- **FAILED** : The correlation run encountered an unrecoverable error.

After a successful upload, correlated SAST and DAST issues in OpenText Application Security are marked as correlated. Correlation links are available in the OpenText Application Security issue view and through the REST API for both SAST and DAST records. Subsequent correlation runs preserve previously identified pairs. SAST findings that are already correlated with a DAST URL are not reprocessed for that URL.

4.7. Trigger SAST audit

The instructions provided in this topic are for auditing a single application. If you want to audit all applications in OpenText Application Security, or a subset of applications having a certain attribute, see the [Perform bulk SAST audit](#) section.

Prerequisites

- You must be a customer user.
- Ensure the following:
 - At least one application is assigned to the entitlement.
 - You have a valid Fortify Remediation Aviator user access token.
- Logged in to your OpenText Application Security session.

Required OpenText Application Security permissions for audit

Before you run the **audit** command, ensure that the OpenText Application Security user has these permissions required to download the source FPR, perform audit-related operations, and upload the audited FPR.

For a dedicated Fortify Remediation Aviator Service Role, assign the following permissions:

- Application and application version permissions
 - View application versions
 - Add applications and application versions
 - Edit application and application versions
 - Delete application versions
- FPR and analysis permissions
 - Download application version FPR with source code
 - Upload analysis results
 - Approve analysis results upload
- Issue permissions
 - Comment on issues
 - Suppress/unsuppress issues
- Custom tag and attachment permissions
 - Edit issue custom tag values and manage issue attachments
 - Edit restricted custom tag values
 - Manage custom tags
 - View custom tags
- Attribute permissions
 - Manage attribute definitions
 - View attribute definitions



Note

If Fortify Remediation Aviator needs to create application versions under existing applications, you can assign **Add application versions to existing applications only** instead of **Add applications and application versions**. This restricts the role from creating new parent applications.

Trigger audit



Caution

- For the Fortify Remediation Aviator to audit FPRs and provide remediation suggestions, ensure the following:
- Source code is available in the FPRs.
- Do not enable the option `-disable-source-bundling` or property `com.fortify.sca.FPRDisableSourceBundling` when you perform the scan through OpenText Fortify SAST.

To trigger an audit:

1. Create a user session to interact with Fortify Remediation Aviator:

```
fcli aviator session login --url <aviator_server_url> --token <access_token>
```



Note

The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`. Ensure to create a user session before auditing.

If you cannot locate your access token, contact your customer administrator.

Optional argument	Description	Default value
<code>--av-session, --aviator-session</code>	Name of the Fortify Remediation Aviator user session.	default

2. Audit the application:

This command downloads the FPR from the specified OpenText Application Security application version, Fortify Remediation Aviator audits the issues and uploads the result back to the OpenText Application Security.

```
fcli aviator ssc audit-sast --av <application_version_name:id>
```

Optional arguments	Description	Default value
<code>--app</code>	Name of the Fortify Remediation Aviator application. If the name is not specified, the build ID of the FPR is considered.	FPR build ID
<code>--tag-mapping</code>	Overrides the default tag mapping using the YAML file. See Audit tag mapping .	<code>tag mapping.yaml</code>
<code>--ssc-session</code>	Name of the OpenText Application Security session to use for auditing.	default
<code>--no-filterset</code>	Do not apply any filter sets, including the default enabled filter set from the FPR.	Not applicable
<code>--av-session</code> , <code>--aviator-session</code>	Name of the Fortify Remediation Aviator user session.	default
<code>--[no-]refresh</code>	By default, this option refreshes the source application version's metrics when copying from it. Note that for large applications, this can lead to an error if the timeout period expires.	Not applicable
<code>--refresh-timeout</code>	Set the timeout period for refreshing the metric. For example, 30s (30 seconds), 5m (5 minutes), 1h (1 hour).	60s
<code>--skip-if-exceeding-quota</code>	Skips audit if the number of open issues exceeds the available quota. When skipped, a summary with the top 10 unaudited categories is displayed.	Not applicable
<code>--source-encodings</code>	Comma-separated source encoding candidates to try in order when decoding source files. Use FPR to try the source encoding recorded in audit.fvdl. When writing remediated files, the accepted encoding is used.	○ FPR ○ UTF-8 ○ ISO-8859-1
<code>--test-exceeding-quota</code>	Checks whether the number of open issues exceeds the available quota and displays the result without performing an audit.	Not applicable

<code>--force-reaudit</code>	Re-audits issues previously processed by Fortify Remediation Aviator, while preserving suppressed and manually triaged issues. This is useful when you want to reprocess issues after making source changes and running a rescan, as issues previously audited by Fortify Remediation Aviator are excluded from new audits by default. Re-audits use the standard Fortify Remediation Aviator quota and processing time.	Not applicable
------------------------------	--	----------------

It might take a few minutes to process the FPR. The duration depends on the FPR size.



Important

If the system running the Fortify Remediation Aviator scan does not have sufficient physical memory, the audit may fail with an *OutOfMemoryError*. This error typically occurs when the available system memory or the configured JVM heap size is insufficient to process a huge number of vulnerabilities (FPR files containing thousands of issues) during the audit phase.

In such cases, ensure that the machine initiating the Fortify Remediation Aviator scan has adequate physical memory and increase the JVM heap allocation before re-running the audit.



Note

- You can use the same access token to audit multiple FPRs on different terminals at the same time.
- You can audit an FPR only once.

1. You can use the following optional methods to audit:

For details on how to prioritize the issues when the audit quota is limited, see [Issue prioritization when audit quota is limited](#).

- Select a filter set for auditing:

```
fcli aviator ssc audit-sast <app_version_name_or_id> --filterset <filterset_name_or_id>
```

Optional argument	Description	Default value
<code>--filterset</code>	Specifies the name or ID of the filter set to apply.	default

- Select folder or folders for auditing.

```
fcli aviator ssc audit-sast <app_version_name_or_id> --filterset <filterset_name_or_id> --folder <folder_name1>,<folder_name2>,...
```

Optional argument	Description	Default value
<code>--folder</code>	Filters the issues by a comma-separated list of specific folder names from the selected filter set (for example, Critical, High). This option requires an active filter set.	Not applicable
<code>--folder-priority-order</code>	Custom priority order for folder-based filtering when quota is exceeded (comma-separated, highest priority first). For example: Critical, High, Medium, Low.	default order

After the Fortify Remediation Aviator processes the FPR, the Action status and the number of audited issues are displayed. Fortify Remediation Aviator uploads the audited FPR back to the OpenText Application Security application version.

To view the Fortify Remediation Aviator's audit results:

1. Open the OpenText Application Security application and go to **Applications > Audit**.
2. Click each row to view the Audit details, such as analysis tag, remediation comment, and the highlighted vulnerable code segment.

Issue prioritization when audit quota is limited

When you run an individual audit or a bulk audit, and the number of open issues in the selected OpenText Application Security application version is higher than the available Fortify Remediation Aviator application quota, Fortify Remediation Aviator audits only a subset of issues. Issue selection is based on folder priority. Issues are prioritized by the folder order stored in the FPR file.

The default order follows the severity hierarchy: Critical → High → Medium → Low. The severe issues are audited first until the available quota is used.

You can customize the default folder prioritization and provide the required folder order by using the Fortify Remediation Aviator command-line option `fcli aviator ssc audit-sast <app_version_name_or_id> --filterset <filterset_name_or_id> --folder <folder_name1>,<folder_name2>,...` or configure the order in the OpenText Application Security application.

If there are more issues in a folder than the remaining quota, Fortify Remediation Aviator selects issues in a deterministic, repeatable way within that folder so that audit results are consistent across runs with the same inputs. Folder-based prioritization is applied consistently for each audit operation and applies to both individual and bulk audits. When the sufficient quota is not available to audit all open issues, the audit summary is displayed. This includes the subset of issues that are audited and shows which folders were fully audited, which were partially audited, and which were not audited.

For example,

- The number of open issues present is 1232 Critical, 242 High, 2324 Medium, and 2323 Low.
- The available Fortify Remediation Aviator application quota is 2000 issues.
- The default prioritization is Critical → High → Medium → Low.
- The following is the result:
 - All 1232 Critical issues are audited.
 - All 242 High issues are audited.
 - 526 Medium issues are audited ($2000 - 1232 - 242 = 526$).
 - 1798 Medium issues and all 2323 Low issues remain unaudited.
 - Remaining quota is 0.

4.8. Trigger DAST audit

Fortify Remediation Aviator performs LLM-based audit of OpenText Fortify DAST findings. For each unaudited, unsuppressed finding in a DAST FPR, Fortify Remediation Aviator constructs a prompt from the relevant issue context, sends it to the LLM for assessment, and writes the audit result back to the DAST FPR in a format compatible with OpenText Application Security upload flows. The result is an audited DAST FPR with audit tags already applied.

The DAST audit command downloads the DAST FPR from the specified OpenText Application Security application version, audits each qualifying finding, and uploads the audited FPR back to OpenText Application Security. Findings that were already audited (manually or by a previous Fortify Remediation Aviator audit run) or suppressed are excluded from processing.

The instructions provided in this topic are for auditing a single application. If you want to audit all applications, or a subset of applications having a certain attribute, see the [Perform bulk DAST audit](#) section.

Prerequisites

- You must be a customer user
- Ensure the following:
 - Have a processed DAST FPR in the OpenText Application Security application version
 - The application exists in Fortify Remediation Aviator and is linked to the OpenText Application Security application version
 - Have a valid user access token
 - Have an active OpenText Application Security session
 - Fortify Remediation Aviator user session is created

Audit DAST findings



Important

- Entitlement and quota availability are validated before any LLM processing begins.
- If the tenant lacks a valid entitlement or available quota, the audit run is rejected before processing starts.

1. Create a user session to interact with Fortify Remediation Aviator.

```
fcli aviator session login --url <aviator_server_url> --token <access_token>
```



Note

- The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.
- Ensure to create a user session before auditing.

2. Audit the DAST findings.

This command downloads the DAST FPR from the specified OpenText Application Security application version, Fortify Remediation Aviator audits the issues, and uploads the result back to the OpenText Application Security.

```
fcli aviator ssc audit-dast --av <application_version_name:id>
```

Optional argument	Description	Default value
<code>--app</code>	Fortify Remediation Aviator application name to associate with the DAST audit. If the name is not specified, the OpenText Application Security application name is used.	OpenText Application Security application name
<code>--tag-mapping</code>	Customize tag mapping for DAST audit results.	<code>tag mapping.yaml</code>
<code>--ssc-session</code>	Name of the OpenText Application Security session to use for auditing.	default
<code>--av-session</code> , <code>--aviator-session</code>	Name of the Fortify Remediation Aviator session.	default
<code>--[no-]refresh</code>	Refreshes out-of-date application version metrics before downloading the FPR. Note that for large applications this can lead to an error if the timeout expires.	Not applicable
<code>--refresh-timeout</code>	Set the timeout period for refreshing the application version metric. For example, 30s (30 seconds), 5m (5 minutes), 1h (1 hour).	60s

After processing completes:

- FCLI displays the audit status and the number of audited findings.
- The audited FPR is uploaded back to OpenText Application Security with audit tags applied.



Note

- Findings that are already audited (manually or by a previous Fortify Remediation Aviator run) and suppressed findings are skipped. Findings that cannot be processed because of quota exhaustion or LLM errors remain unaudited.
- Individual finding failures do not cause the entire audit run to fail.

To view the audit results:

1. Open the OpenText Application Security application and go to **Applications > Audit**.
2. Click each issue row to view audit details, such as the analysis tag, remediation comment, and the highlighted vulnerable code segment.

Quota and Entitlement

DAST auditing uses the same entitlement and quota model as SAST auditing. Before any LLM calls are made, the Fortify Remediation Aviator verifies valid entitlement and available quota. If either check fails, the run is rejected before processing begins. Quota is consumed per audited DAST finding. If quota is exhausted during the audit:

- Remaining findings are left unaudited.
- Completed audit results are preserved.
- The partially audited FPR is still uploaded to OpenText Application Security.
- The overall audit run does not fail.

For additional quota information, see [Limitations](#).

4.9. Perform bulk SAST audit



Important

If the system running the Fortify Remediation Aviator scan does not have sufficient physical memory, the audit may fail with an *OutOfMemoryError*. This error typically occurs when the available system memory or the configured JVM heap size is insufficient to process a huge number of vulnerabilities (FPR files containing thousands of issues) during the audit phase.

In such cases, ensure that the machine initiating the Fortify Remediation Aviator scan has adequate physical memory and increase the JVM heap allocation before re-running the audit.

The bulk audit option is available for the fcli v3.13.1 and later. It helps to audit multiple projects in OpenText Application Security with unaudited issues using a single command.

The bulk audit task identifies OpenText Application Security application versions with pending audit issues and automatically submits them to Fortify Remediation Aviator for auditing. The action filters to only process versions with actual pending audit issues, automatically ignoring versions for which audit information needs to be refreshed.

For application versions that are not available in Fortify Remediation Aviator, the action will automatically create them before submitting for audit.

You must specify either `--tag-mapping` or `--add-aviator-tags` option. The default tag mapping does not audit unsure issues, causing indefinite reselection. When `--tag-mapping` is specified, tags are required for all the values in the .yaml file, for example, for Tier_1 and Tier_2, all values FP, TP, and Unsure should have the tags correctly specified, else FPRs will be reaudited.

When `--add-aviator-tags` is specified, fcli runs `fcli aviator prepare` command for each application before auditing. This option automatically ensures that the Fortify Remediation Aviator specific custom tags exist in OpenText Application Security and is associated with the specified issue templates and/or application versions.

Prerequisites

Ensure the following:

- Fortify Remediation Aviator administrator configuration is created
- Active OpenText Application Security session
- User session is created

Trigger bulk audit

To perform bulk auditing for OpenText Application Security, run the following command.

```
fcli ssc action run bulkaudit-sast --add-aviator-tags
```

Optional argument	Description	Default value
<code>--max-audits</code> , <code>-m</code>	Maximum number of project versions to audit.	-1 (unlimited)
<code>--filter</code> , <code>-f</code>	An optional filter to apply when querying OpenText Application Security for projects. For example, 'Languages:java'. For information on how to use this option, see the Usage of --filter section.	no filtering
<code>--exclude-filter</code> , <code>-e</code>	An optional inverse filter to exclude projects from audit. Projects matching this filter will be skipped. For example, 'Languages:c#' to exclude .NET projects. This can be combined with <code>--filter</code> .	no exclusion
<code>--dry-run</code> , <code>-n</code>	Shows what Fortify Remediation Aviator commands will be run without actually running them.	false
<code>--tag-mapping</code> , <code>-t</code>	The path to the tag mapping YAML file. This custom mapping ensures that Fortify Remediation Aviator's 'Unsure' audit results are properly handled.  Note • Use this option if you do not use <code>--add-aviator-tags</code> . • Ensure to use either <code>--tag-mapping</code> or <code>--add-aviator-tags</code> when you run the <code>bulkaudit</code> .	Not applicable

<code>--add-aviator-tags</code>	If specified, fcli runs <code>fcli aviator prepare</code> for the application before auditing.  Note - Use this option if you do not use <code>--tag-mapping</code>. - Ensure to use either <code>--tag-mapping</code> or <code>--add-aviator-tags</code> when you run the <code>bulkaudit</code>.	Not applicable
<code>--aviator-app-mapping</code>	Controls how OpenText Application Security application versions map to Fortify Remediation Aviator applications. The supported values are: <code>app</code>: Maps one Fortify Remediation Aviator application to one OpenText Application Security application. <code>version</code>: Maps one Fortify Remediation Aviator application to one OpenText Application Security application version. Each application version consumes a separate application entitlement. When you use <code>version</code>, application names reflect the mapping by including both the OpenText Application Security application name and the OpenText Application Security application version name. For more information, see Aviator application mapping.	<code>app</code>
<code>--refresh</code>	Refreshes application version metrics before audit. For large applications, this can lead to timeout errors.	true

<code>--refresh-timeout</code>	Timeout period for metric refresh. For example, 30s (30 seconds), 5m (5 minutes), 1h (1 hour).	60s
<code>--log-file</code>	Saves the log output to a file. It is a good practice to use this option. You can refer to the log file for the complete information corresponding to the triggered <code>bulkaudit</code> .	./fcli.log
<code>--skip-if-exceeding-quota</code>	Skips audit if the number of open issues exceeds the available quota. When skipped, a summary with the top 10 unaudited categories is displayed.	Not applicable
<code>--test-exceeding-quota</code>	Checks whether the number of open issues exceeds the available quota and displays the result without performing an audit.	Not applicable



Note

Use either the `--tag-mapping` or `--add-aviator-tags` option, else an error is returned.

Fortify Remediation Aviator application mapping

Bulk audit maps OpenText Application Security content to Fortify Remediation Aviator applications in two ways. Use the `--aviator-app-mapping` option to select the mapping mode.

- Project-based mapping (default): One Fortify Remediation Aviator application for one OpenText Application Security application. All project versions in the project are audited under the same Fortify Remediation Aviator application.
- Version-based mapping: One Fortify Remediation Aviator application for one OpenText Application Security application version. Use this mode when OpenText Application Security project versions represent separate components that you want to track and audit independently.



Note

In version-based mapping, each OpenText Application Security application version consumes a separate Fortify Remediation Aviator application entitlement. To avoid unexpected entitlement usage, use a consistent mapping mode across bulk audit runs.

Usage of `--filter`

The `--filter` option in fcli enables you to perform a bulk audit by targeting specific attributes in your OpenText Application Security application.

For default attributes

1. Refer to your OpenText Application Security application to obtain the corresponding value for the default attribute that you want to audit.
2. Run bulk audit:

```
fcli ssc action run bulkaudit-sast --filter <default_tag_attribute:value> --add-aviator-tags
```

For example:

```
fcli ssc action run bulkaudit-sast --filter Languages:ABAP --add-aviator-tags
```

For custom attributes

1. List the OpenText Application Security attributes:

This displays details, such as ID, Category, Guid, Name, Type, and Required.

```
fcli ssc attribute lsd
```

2. Get the Guid information of the required custom tag attribute:

This command displays the related options, which contain ID, guid, name, description, hidden, inUse, index, projectMetaDataDefId, publishVersion, and objectVersion.

```
fcli ssc attribute get-definition <custom_tag_attribute_guid_value>
```

3. Run the bulk audit:

```
fcli ssc action run bulkaudit-sast --filter <custom_tag_attribute_guid_value:required_guid_value> --add-aviator-tags
```

4.10. Perform bulk DAST audit

The bulk DAST audit action identifies OpenText Application Security application versions with a processed DAST scan result and submits them to Fortify Remediation Aviator for audit using a single command. You can process all eligible application versions or limit the scope by applying filters.

The action uses a two-stage selection process:

1. For each application version, it queries and checks for a processed DAST FPR:

```
/projectVersions/{parentId}/artifacts
```

Versions without a DAST artifact are skipped without querying attributes.

2. For versions with a DAST FPR:
 - The action reads the most recent DAST scan date.
 - It queries for the `last_dast_audit` attribute:

```
/projectVersions/{parentId}/attributes
```

A version is selected when:

- The attribute is absent, or
- The DAST scan date is later than the stored audit date.

A version is skipped when:

- The stored audit date is equal to or later than the most recent DAST scan date.

Before auditing an application version, the action checks for a Fortify Remediation Aviator application corresponding to the OpenText Application Security project.

- If no application exists, the action creates one using the OpenText Application Security project name.
- If application creation fails:
 - A warning is logged.
 - The action stops attempting to create applications for the remainder of the run.

- Processing continues for versions that already have a Fortify Remediation Aviator.

Prerequisites

Ensure the following:

- Fortify Remediation Aviator administrator configuration is created.
- An active OpenText Application Security session.
- An active Fortify Remediation Aviator session.
- User session is created.

Trigger bulk DAST audit

Run the following command:

```
fcli ssc action run bulkaudit-dast
```

Optional argument	Description	Default value
<code>--max-audits</code> , <code>-m</code>	Maximum number of project versions to audit.	-1 (unlimited)
<code>--filter</code> , <code>-f</code>	An optional filter that restricts the application versions considered for audit. For information on how to use this option, see the <i>Usage of <code>--filter</code></i> section in Perform bulk SAST audit.	no filtering
<code>--exclude-filter</code> , <code>-e</code>	An optional inverse filter that removes matching application versions from the candidate set. For example, <code>'Languages:c#'</code> to exclude .NET projects. This can be combined with <code>--filter</code> .	no exclusion
<code>--dry-run</code> , <code>-n</code>	Logs the application versions that would be processed without invoking the DAST audit command. - Output includes counts for: - Versions skipped because they do not have a DAST artifact. - Versions skipped because their <code>last_dast_audit</code> attribute is up to date.	false
<code>--tag-mapping</code> , <code>-t</code>	The path to tag mapping YAML file. This custom mapping ensures that Fortify Remediation Aviator's 'Unsure' audit results are properly handled.	Not applicable

<code>--aviator-app-mapping</code>	Controls how OpenText Application Security application versions map to Fortify Remediation Aviator applications. The supported values are: <code>app</code> : Maps one Fortify Remediation Aviator application to one OpenText Application Security application. <code>version</code> : Maps one Fortify Remediation Aviator application to one OpenText Application Security application version. Each application version consumes a separate application entitlement. When you use <code>version</code>, application names reflect the mapping by including both the OpenText Application Security application name and the application version name. For more information, see the <i>Fortify Remediation Aviator application mapping</i> section in Perform bulk SAST audit.	<code>app</code>
<code>--[no-]refresh</code>	By default, this option refreshes the source application version's metrics when copying from it. Note that for large applications, this can lead to an error if the timeout period expires.	Not applicable
<code>--refresh</code>	Refreshes application version metrics before audit. For large applications, this can lead to timeout errors.	true
<code>--refresh-timeout</code>	Timeout period for metric refresh. For example, 30s (30 seconds), 5m (5 minutes), 1h (1 hour).	60s

For each selected application version, the action runs `fcli aviator ssc audit-dast` in sequence. If auditing one application version fails, the action logs a warning and continues processing the remaining versions. When the action is complete, the action reports:

- Number of Fortify Remediation Aviator applications created.
- Number of audit runs attempted.
- Number of versions skipped because they have no processed DAST artifact.
- Number of versions skipped because `last_dast_audit` is equal to or later than the latest DAST scan date.
- Number of failures.
- Whether entitlement was exhausted during the run.

4.11. Perform bulk correlation

The bulk correlation action automatically identifies application versions that have both a OpenText Fortify SAST scan result and a OpenText Fortify DAST scan result, then runs SAST-DAST correlation across all of them or a filtered subset. This allows you to run correlation from a single command, without triggering it individually for each application version.

The action uses a two-stage selection process before submitting any work. For each application version, it first checks whether both a processed SAST FPR and a processed DAST FPR are present. Versions missing either scan type are skipped. For versions that have both, the action compares the most recent scan date against the `last_correlation` custom attribute on the application version. If the attribute records a date equal to or later than the most recent scan, the version is skipped because the scan data has not changed since the last correlation run. Versions with a newer scan date, or no recorded correlation date, are selected for correlation.

If an application version does not exist, the action creates it before running correlation. If entitlement is exhausted, a warning is logged and remaining versions that would require a new application are skipped; versions with an existing application continue to be processed. A failure on one application version does not abort the overall run.

For instructions on correlating a single application version, see [Correlation of OpenText SAST and OpenText DAST results](#).

Prerequisites

Ensure the following:

- Fortify Remediation Aviator administrator configuration is created
- Active OpenText Application Security session
- User session is created

Trigger bulk correlation

To perform bulk SAST-DAST correlation for an OpenText Application Security application, run the following command:

```
fcli ssc action run bulkcorrelate
```

Optional argument	Description	Default value
<code>--max-correlations</code> , <code>-m</code>	Maximum number of application versions to correlate in a single run.	-1 (unlimited)
<code>--filter</code> , <code>-f</code>	An optional filter to apply when querying for projects. For example, <code>'Languages:java'</code> .	no filtering
<code>--exclude-filter</code> , <code>-e</code>	An optional inverse filter to exclude projects from correlation. Projects matching this filter are skipped. For example, <code>'Languages:c#'</code> to exclude .NET projects. This can be combined with <code>--filter</code> .	no exclusion
<code>--dry-run</code> , <code>-n</code>	Shows which application versions would be processed without actually running correlation. Dry-run output includes the number of versions skipped due to missing SAST or DAST results.	false

At completion, FCLI displays the following:

- Number of applications created during the run.
- Number of correlations attempted.
- Number of failures.
- Whether entitlement was exhausted during the run.

After a successful run, the `last_correlation` custom attribute is written to each processed application version in OpenText Application Security. This timestamp is used on subsequent bulk correlation runs to skip versions whose scan data has not changed.

4.12. Perform auto-remediation

Fortify Remediation Aviator allows users to download the audited FPR file from the OpenText Application Security artifact ID and automatically apply the remediation proposed by Fortify Remediation Aviator to the source code with a single command. This enables users to fix their source code without any manual intervention. You can apply remediation from:

- The most recent Fortify Remediation Aviator audit
- All Fortify Remediation Aviator audits in an application version
- A specific Fortify Remediation Aviator artifact

Fortify on Demand users can apply the remediation from a specific artifact. For more information, see the *OpenText Fortify on Demand User Guide*.

Prerequisites

Ensure the following:

- Have an active user session.
- Already performed the Fortify Remediation Aviator audit.
- Logged in to OpenText™ Application Security /OpenText™ Fortify on Demand.

Apply the auto-remediation

For OpenText™ Application Security (Software Security Center)

Downloads FPR artifact(s) from OpenText Application Security, or reads a local remediations cache zip, and applies Fortify Remediation Aviator-generated remediations to the specified source directory. Based on how you want to apply remediations, ensure to specify one of the options `--from-cache`, `--artifact-id`, `--latest`, or `--all`. Online selection requires an active OpenText Application Security session for all the options except `--from-cache`, which is an offline selection.

1. Download audited FPR file(s) from OpenText Application Security into a single remediations cache zip for use with `apply-remediations --from-cache`. Exactly one of `--artifact-id`, `--latest`, or `--all` must be specified. Requires `-f / --file`, which is the destination remediations cache zip path. Overwriting an existing file requires confirmation (`-y / --confirm`).

```
fcli aviator ssc download-remediations-cache -f <cachezip_path>
```

Optional arguments	Description	Default value
<code>--ssc-session</code>	Name of the OpenText Application Security session.	default
<code>-y</code> , <code>--confirm</code>	Confirm overwriting the existing remediations cache file.	Not applicable
<code>--rel</code> , <code>--release</code>	Specifies release ID or the application's release name.	Not applicable

2. Select one of the following options based on how you want to choose Fortify Remediation Aviator audit data:

- Apply remediations from a remediations cache zip:

```
fcli aviator ssc apply-remediations --from-cache ./remediations.zip
```

- Apply selected issue IDs from a cache zip:

```
fcli aviator ssc apply-remediations --from-cache ./remediations.zip --issue-ids <id1>,<id2>,...
```

- Apply remediations from a known artifact using artifact ID:

```
fcli aviator ssc apply-remediations --artifact-id <artifactID>
```



Tip

Run `fcli ssc artifact list --av <application_versionnameorID>` to get the application version artifact information.

- Apply remediations from the latest Fortify Remediation Aviator-processed artifact for an application version:

```
fcli aviator ssc apply-remediations --av <appName:appID> --latest
```

- Apply remediations from the latest artifact, restricting to those uploaded in the last few days:

```
fcli aviator ssc apply-remediations --av <appName:appID> --latest --since <duration>
```

- Apply remediations across all Fortify Remediation Aviator-processed artifacts for an application version:

```
fcli aviator ssc apply-remediations --av <appName:appID> --all
```



Note

This option downloads and processes multiple FPR files. Execution time may increase for application versions with many Fortify Remediation Aviator runs.

- Apply remediations across all artifacts uploaded in the last few weeks:

```
fcli aviator ssc apply-remediations --av <appName:appID> --all --since <duration>
```

- Apply remediations from the latest artifact using a custom source directory.

```
fcli aviator ssc apply-remediations --av <appName:appID> --latest --source-dir <source_dir>
```

Optional arguments	Description	Default value
<code>--ssc-session</code>	Name of the OpenText Application Security session.	default
<code>--av</code> , <code>--appversion</code>	Specifies application version ID or <application>:<version> name.	Not applicable
<code>--source-dir</code>	Specifies the directory containing the source code where remediations are applied. If the directory is not specified, remediations are applied to the current working directory.	Current working directory
<code>--since</code>	Limits artifact selection by upload time. The supported formats are: - Relative durations: <code>7d</code> , <code>2w</code> , <code>1M</code> - Dates/timestamps: <code>2025-01-01</code> , <code>2025-01-01T10:30:00Z</code>	Not applicable
<code>--from-cache</code>	Local remediations cache zip produced by <code>download-remediations-cache</code> . Mutually exclusive with online selection options. Does not require an OpenText Application Security session.	Not applicable
<code>--issue-ids</code>	Comma-separated list of issue IDs to apply. Matches requested values against remediations.xml instancelid values. Requires <code>--from-cache</code> so integrations can download once using <code>download-remediations-cache</code> and apply selected remediations without repeated OpenText Application Security downloads.	Not applicable
<code>--preview</code>	Shows available remediations and their proposed changes without modifying source files. Does not validate whether the proposed changes apply cleanly to the current source.	Not applicable
<code>--latest</code>	Select the most recent Fortify Remediation Aviator-processed artifact. Requires <code>--av</code> / <code>--appversion</code> . Mutually exclusive with <code>--artifact-id</code> and <code>--all</code> .	Not applicable

After the remediations are applied, FCLI displays the following:

- Application version ID and artifact ID
- Number of artifacts processed and skipped
- Total number of remediations identified
- Number of remediations applied and skipped
- Overall action status

For OpenText™ Fortify on Demand

For information on how to configure a static scan using Fortify on Demand, see the *OpenText™ Fortify on Demand User Guide*.

`apply-remediations` command downloads the FPR from a Fortify on Demand release, or reads a local remediations cache zip, and applies Fortify Remediation Aviator remediations to the source directory. Exactly one of `--release / --rel` or `--from-cache` must be specified. Online selection requires a Fortify on Demand session; `--from-cache` does not because it is an offline selection.

1. Get the release details for the specified application.

```
fcli fod release list --app <appNameOrId>
```

2. Download the latest static audited FPR for a Fortify on Demand release into a remediations cache zip for use with `apply-remediations --from-cache`. Requires `-f / --file`, which is the destination remediations cache zip path. Overwriting an existing file requires confirmation (`-y / --confirm`).

```
fcli fod aviator download-remediations-cache -f <cachezip_path>
```

Optional arguments	Description	Default value
<code>--fod-session</code>	Name of the Fortify on Demand session.	default
<code>-y</code> , <code>--confirm</code>	Confirm overwriting the existing remediations cache file.	Not applicable
<code>--rel</code> , <code>--release</code>	Specifies release ID or the application's release name.	Not applicable

3. Apply remediation for the required release ID.

```
fcli fod aviator apply-remediations --rel <releaseID>
```

Optional arguments	Description	Default value
<code>--source-dir</code>	Specifies the directory containing the source code where remediations are applied. If the directory is not specified, remediations are applied to the current working directory.	Current working directory
<code>--from-cache</code>	Local remediations cache zip produced by <code>download-remediations-cache</code> . Mutually exclusive with <code>--release / --rel</code> . Does not require a Fortify on Demand session.	Not applicable
<code>--issue-ids</code>	Comma-separated list of issue IDs to apply. Matches requested values against <code>remediations.xml</code> <code>instanceId</code> entries. Requires <code>--from-cache</code> so integrations can download once using <code>download remediations-cache</code> and apply selected remediations without repeated Fortify on Demand downloads.	Not applicable
<code>--fod-session</code>	Name of the Fortify on Demand session.	default
<code>--preview</code>	Shows available remediations and their proposed changes without modifying source files. Does not validate whether the proposed changes apply cleanly to the current source.	Not applicable

After the remediations are applied, FCLI displays the following:

- Release ID
- Total number of remediations identified
- Number of remediations applied and skipped
- Overall action status



Note

- If the source code file is still the same from the time when it was scanned by OpenText Fortify SAST and then audited by Fortify Remediation Aviator, the remediations are implemented.
- If the source code file has been changed in the meantime, remediations are implemented only if the relevant context is located accurately else remediations are skipped.

Apply remediation from a local FPR

You can provide a local FPR file to `fcli aviator ssc apply-remediations` or `fcli fod aviator apply-remediations`. When you provide a local FPR, FCLI loads the remediations from that file and applies them without downloading an FPR from OpenText Application Security or Fortify on Demand.

Use the command's local FPR input option to specify the path to the FPR file. FCLI validates the path before processing starts. If the file does not exist, cannot be read, or is not a valid FPR, FCLI returns a clear error.

If you do not provide a local FPR path, FCLI keeps the existing behavior and downloads the audited FPR from the selected OpenText Application Security or Fortify on Demand artifact.

Missing source files

When a source file referenced by a remediation is missing, FCLI logs a warning and skips the remediation for the affected issue. FCLI continues processing the remaining remediations.

The final summary includes successfully remediated issues, skipped issues caused by missing source files, and failed issues. Missing source files do not terminate the overall execution.

Source file encoding behavior

When processing source files, `fcli aviator ssc apply-remediations` does not assume that all files use UTF-8 encoding. Instead, it attempts to read files using the following default encoding strategy:

- Encoding specified in the FPR
- UTF-8
- ISO-8859-1

If a source file cannot be processed using any of these encodings, the command reports the file processing failure explicitly.

If your source files use a different encoding, specify one or more additional encodings when running the command.



Note

OpenText recommends validating remediation results when processing source files that use multiple encodings, as mixed encodings can affect remediation outcomes.

Deterministic application behavior

`fcli aviator ssc apply-remediations` applies remediations predictably and consistently, ensuring the same result when run against the same source code and remediation data. To ensure reliable outcomes, the command:

- Normalizes generated fixes before applying them.
- Uses canonical hash validation that consistently handles file encodings, line endings, and trailing newlines.
- Supports both canonical and legacy hash formats for compatibility, and logs which format was matched.
- Recognizes identical or nearly identical fixes at the hunk level, allowing a single code change to satisfy multiple remediation instances.
- Verifies that the expected source code is present at the target location before applying a fix and reports any mismatches.

Unique anchor match validation

When the file hash does not match the state recorded during the scan, FCLI uses a three-line anchor (a small snippet of the original source code used to locate the remediation target) to find the correct location before applying the remediation.

- If exactly one anchor match is found, FCLI applies the remediation.
- If multiple anchor matches are found, FCLI reports the remediation as ambiguous, does not apply it, and displays the candidate line numbers for manual review.
- If no anchor match is found, FCLI reports the remediation as failed.
- FCLI never selects a match automatically. It does not use nearest-line matching or any other best-guess approach.



Note

Anchor matching is used only when the file hash does not match the scanned state and FCLI must locate the target using source-code context. When the file hash matches, FCLI uses the verified location directly and does not perform anchor matching.

Remediation outcomes

Each remediation result is in one of the following outcomes.

Remediation outcomes

Outcome	Field	Description
Identical	<code>identicalRemediations</code>	The remediation proposes the same change as a remediation that was already applied earlier in the same run. Because the proposed change is already present, no additional change is written.
Superseded	<code>supersededRemediations</code>	The remediation applies to code that is fully contained within a change that was already applied, and the earlier change already contains the proposed remediation. No additional change is written.
Possible remediated	<code>possiblyRemediatedRemediations</code>	The remediation applies to code that is fully contained within an earlier change, but the resulting code does not exactly match the proposed remediation. Another remediation might have already addressed the issue, so no additional change is written.
Conflicts	<code>skippedRemediations</code> under the reason "Conflicts with another fix"	The remediation partially overlaps a change that was already applied, and neither remediation fully contains the other. Fortify Remediation Aviator does not apply the overlapping remediation to avoid making an ambiguous change.
Applied	<code>appliedRemediations</code>	The remediation does not meet any of the preceding conditions. Fortify Remediation Aviator locates the applicable source code and writes the remediation to the source file.
Skipped	<code>skippedRemediations</code>	Fortify Remediation Aviator cannot apply the remediation for another reason, such as when the expected source context cannot be found or the target file is unavailable.



Note

Fortify Remediation Aviator evaluates these outcomes in the order shown above. Because broader remediations are processed first, a narrower remediation can be classified as **Superseded**, **Possibly remediated**, or **Conflicts** based on a remediation that was applied earlier in the same run.

5. Manage tenant information

Prerequisite

You must be a registered customer administrator.

List of customer administrator tasks

You can perform the following administrator tasks using the Fortify Remediation Aviator fcli:

- [Manage administrator configurations](#)
 - Create an administrator configuration.
 - List the administrator configurations.
 - Delete an administrator configuration.
- [Manage user tokens](#)
 - Create a user access token.
 - List the access tokens.
 - Validate, revoke, or delete the access tokens.
- [Manage applications](#)
 - Create an application.
 - Get the application details.
 - List the available applications.
 - Update or delete the applications.
- [Manage entitlements](#)
 - Retrieve the entitlement details for a specified tenant.



Note

You must create an **admin-config** to interact with Fortify Remediation Aviator before performing an administrator task.

5.1. Manage administrator configurations

You can perform the following tasks using the **fcli aviator admin-config** command.

- Create an administrator configuration for interacting with Fortify Remediation Aviator:

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-key  
<path_to_private_key.pem>
```



Note

The `--private-key` can be a file containing the key or the key itself. Ensure that the key is in the PEM format.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

- List the available Fortify Remediation Aviator administrator configurations:

```
fcli aviator admin-config list
```

- Delete the Fortify Remediation Aviator administrator configuration:

```
fcli aviator admin-config delete --admin-config <administrator_configuration_name>
```

5.2. Manage user tokens

You can perform the following tasks using the **fcli aviator token** command.

- Create an access token for a user:

```
fcli aviator token create --email <user_email_id> --save-token <output_file>
```

Optional arguments	Description	Default value
<code>--name</code>	Specifies the token name.	Derived from <code>--email</code>
<code>--save-token</code>	Saves the generated raw token string to the specified file. By default, the string is in json format.	NA
<code>-o</code> , <code>--output</code>	Specifies the token format. The available formats are csv, table, expr, json, xml, and yaml.	NA
<code>--to-file</code>	Specifies a file to save the output.	NA
<code>--end-date</code>	Specifies the token expiration date in YYYY-MM-DD format.	30 days from the date of creation



Note

OpenText recommends using `--save-token` optional argument to ease the user experience when using the access token.

- Retrieve the list of access tokens within Fortify Remediation Aviator tenant:

```
fcli aviator token list
```

- Retrieve a list of access tokens for a specified user within Fortify Remediation Aviator tenant:

```
fcli aviator token list --email <user_email_id>
```

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

- Validate an existing access token for a user within Fortify Remediation Aviator tenant, checking its authenticity and status:

```
fcli aviator token validate --token <access_token>
```



Note

The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

- Revoke an access token for a user within an Fortify Remediation Aviator tenant. This task invalidates the access token without permanently deleting it:

```
fcli aviator token revoke --token <access_token>
```



Note

The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

- Delete an existing access token for a user:

```
fcli aviator token delete --token <access_token>
```



Note

The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

5.3. Manage applications

You can perform the following tasks using the **fcli aviator app** command.

- Create new Fortify Remediation Aviator applications:

```
fcli aviator app create <aviator_application_name>
```

Optional argument	Description	Default value
--admin-config	Name of the Fortify Remediation Aviator administrator configuration.	default

- Add one entitlement to an existing Fortify Remediation Aviator application:

```
fcli aviator app add-entitlement <application_id>
```

This operation increments the entitlement multiplier of an existing Fortify Remediation Aviator application by one by one and displays the updated application details.

Optional argument	Description	Default value
--admin-config	Name of the Fortify Remediation Aviator administrator configuration.	default

- Get the details of an existing Fortify Remediation Aviator application for a particular tenant:

```
fcli aviator app get <application_id>
```

Optional argument	Description	Default value
--admin-config	Name of the Fortify Remediation Aviator administrator configuration.	default

- List all Fortify Remediation Aviator applications for a particular tenant in the admin configuration:

```
fcli aviator app list
```

Optional argument	Description	Default value
--admin-config	Name of the Fortify Remediation Aviator administrator configuration.	default

- Rename an existing Fortify Remediation Aviator application identified by its ID:

```
fcli aviator app update <application_id> -n <new_application_name>
```

Optional argument	Description	Default value
--admin-config	Name of the Fortify Remediation Aviator administrator configuration.	default

- Delete a Fortify Remediation Aviator application by its ID:

```
fcli aviator app delete <application_id>
```

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration.	default

5.4. Manage entitlements

You can manage OpenText Fortify SAST and OpenText Fortify DAST entitlements using the **fcli aviator entitlement** command



Note

The `fcli aviator entitlement list` command has been deprecated.

- Retrieve the OpenText Fortify SAST entitlement details for a specified tenant:

```
fcli aviator entitlement list-sast
```

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration to use.	default

- Retrieve the OpenText Fortify DAST entitlement details for a specified tenant:

```
fcli aviator entitlement list-dast
```

Optional argument	Description	Default value
<code>--admin-config</code>	Name of the Fortify Remediation Aviator administrator configuration to use.	default

The following entitlement details are retrieved:

- Total number of entitlements available.
- Number of active entitlements.
- Total number of applications linked to the entitlements that are already consumed.
- Number of remaining applications under the available entitlements.
- Expiration date for each entitlement.



Note

Deactivated applications and removed entitlements are not included in active customer-facing entitlement responses. They remain unavailable for use, while their quota history is preserved for quota accounting.

6. FAQs

What should you do if you disagree with the Fortify Remediation Aviator audit result?

OpenText recommends creating a support request and sharing the FPR.

What should you do if you cannot locate your access token?

Contact your customer administrator.

What should you do if you run out of entitlements?

Contact your account representative.

What happens if your audit exceeds the quota?

The audit will stop once the quota is reached. The FPR will still be generated and uploaded, but further issues will not be audited.

Will quota changes affect past audits after configuring new Initial and Monthly quotas?

No, changes apply only to future audits and quota top-ups.

How to check your remaining quota?

Quota usage is visible in the audit logs for each application or contact your customer administrator.



© Copyright 2026 Open Text

For more info, visit <https://docs.microfocus.com>
