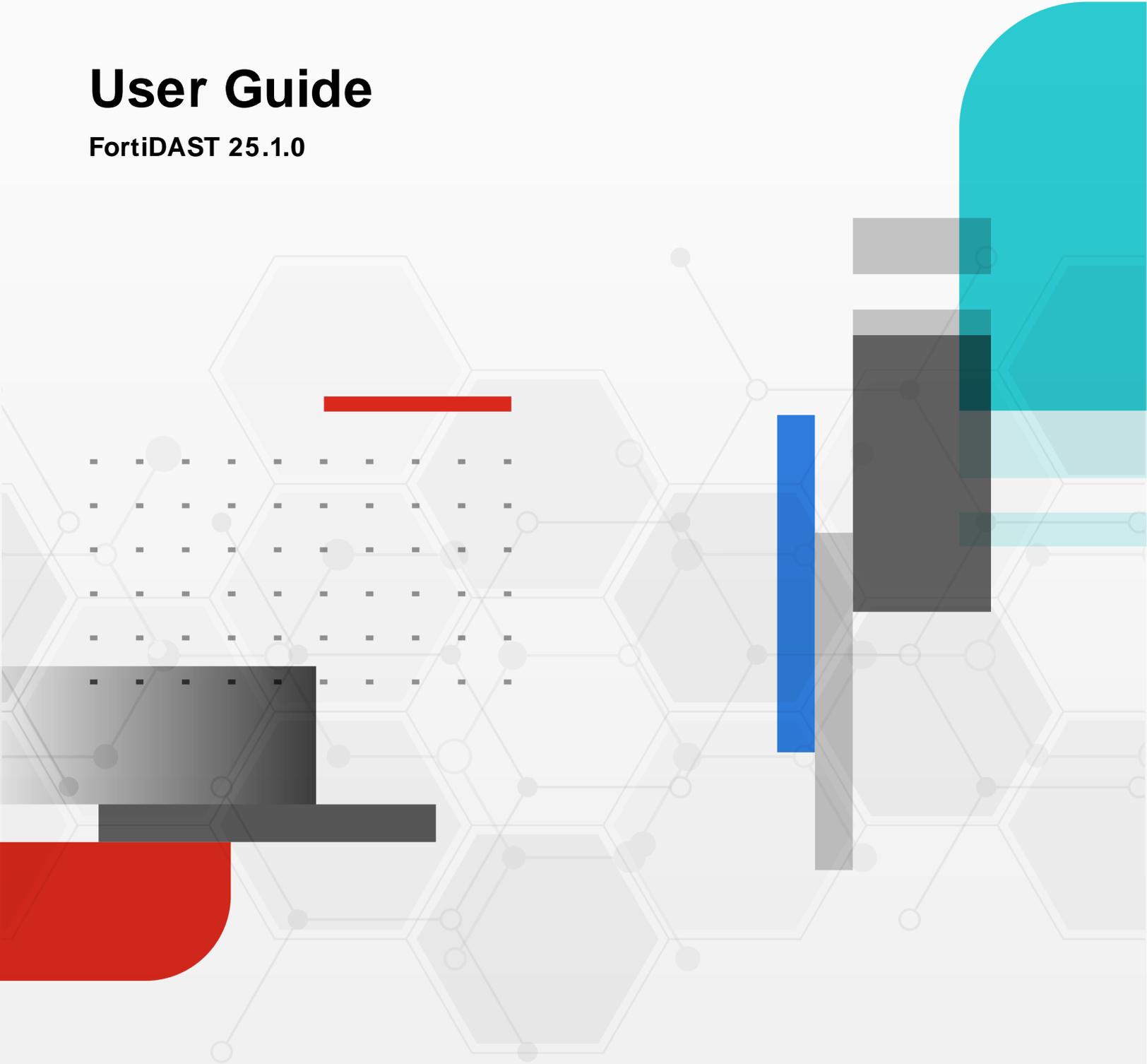


# User Guide

FortiDAST 25.1.0



**FORTINET DOCUMENT LIBRARY**

<https://docs.fortinet.com>

**FORTINET VIDEO LIBRARY**

<https://video.fortinet.com>

**FORTINET BLOG**

<https://blog.fortinet.com>

**CUSTOMER SERVICE & SUPPORT**

<https://support.fortinet.com>

**FORTINET TRAINING & CERTIFICATION PROGRAM**

<https://www.fortinet.com/training-certification>

**FORTINET TRAINING INSTITUTE**

<https://training.fortinet.com>

**FORTIGUARD LABS**

<https://www.fortiguard.com>

**END USER LICENSE AGREEMENT**

<https://www.fortinet.com/doc/legal/EULA.pdf>

**FEEDBACK**

Email: [techdoc@fortinet.com](mailto:techdoc@fortinet.com)

March 20, 2025

FortiDAST 25.1.0 User Guide

67-251-1129169-20250320

# TABLE OF CONTENTS

<b>Change log</b> .....	<b>5</b>
<b>Introduction</b> .....	<b>6</b>
What is FortiDAST .....	6
How FortiDAST Works .....	13
User Interface Overview .....	13
Notifications .....	14
Outbreak Alerts .....	14
<b>Licensing</b> .....	<b>15</b>
<b>Signing-on for FortiDAST</b> .....	<b>16</b>
Registering on FortiCloud .....	16
Accessing FortiDAST .....	16
External IDP Authentication .....	17
<b>Vulnerability Assessment (scanning) of an Asset</b> .....	<b>20</b>
Asset Authorization .....	20
Schedule Scan .....	22
Configuring the Scanner .....	22
HTTP (Authentication via Cookies) .....	23
Authentication (via GUI) .....	24
Coverage .....	26
Forced Browsing .....	29
Inclusion/Exemption List .....	30
WAF Configuration .....	31
Web Filter .....	33
API Crawling .....	33
Jira Integration .....	34
IDOR .....	35
FortiDAST Proxy .....	36
Custom Notifications .....	37
Automation .....	38
Business Logic Recorder .....	38
Login and Replay .....	40
Automation Scripts .....	46
Exploit Engine .....	49
Configuring Exploit Engine .....	63
Vulnerability Scanning .....	64
Progress Summary .....	64
FortiDAST Proxy Server .....	65
CI/CD Tools .....	67
Settings .....	69
Jira Integration .....	69
REST API .....	70
Email Notification .....	70
FortiAppSec Cloud WAF Virtual Patching .....	71

---

<b>Analyzing the Scan Data</b> .....	<b>73</b>
Scans Overview (Scan Result) .....	73
Compare Scans .....	73
Summary .....	74
Vulnerabilities .....	85
Virtual Patching .....	86
Dashboard .....	87
Outbreak Alerts .....	89
<b>Integrations</b> .....	<b>93</b>
WAF Integration .....	93
FortiAppSec Cloud WAF .....	93
FortiWeb Appliances .....	94
DevOps Integration .....	94
Jenkins Setup .....	95
GitLab Setup .....	97
Jira .....	99

## Change log

Date	Change Description
2025-03-20	Initial release

# Introduction

The vast growth of the World Wide Web or the internet led to an equally enormous spiral in network security vulnerabilities which could potentially be exploited due to the immense advancement in hacking techniques and cyber-attack methodologies. The almost global use of modern complex web applications makes them easily prone to cyber- attacks and violations. These web applications contain multiple unassessed security risks and vulnerabilities. In such a scenario, network security is of prime importance.

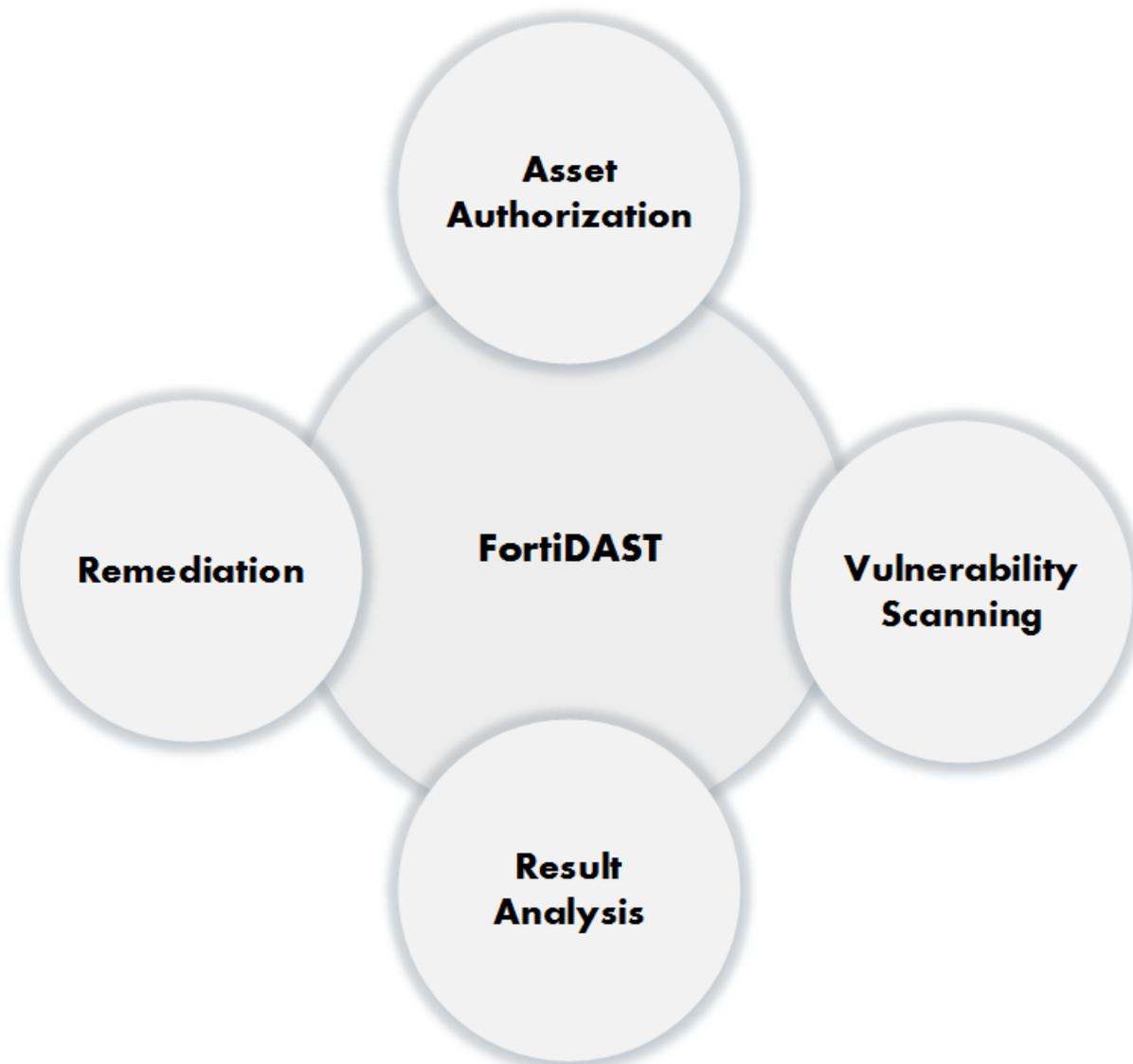
- [What is FortiDAST on page 6](#)
- [How FortiDAST Works on page 13](#)
- [User Interface Overview on page 13](#)

## What is FortiDAST

FortiDAST is a cloud enabled service that performs web application vulnerability testing through an intensive process of comprehensive and criteria based automated scanning and analysis. It adopts an organised technical approach of assessing your web applications running in an HTTP/HTTPS environment, to identify loopholes and vulnerabilities. Penetration testing (pen-testing) is the process to explore and exploit security vulnerabilities in an application using various malicious techniques to discover security gaps; securing your network and assisting in suitable remediation steps for the identified susceptibilities.

The goal of FortiDAST is to provide an easy-to-understand and non-intrusive evaluation of the security posture of your web applications. The outcome is an accurate and detailed vulnerability assessment report with a high vulnerability detection rate that facilitates appropriate measures for remediation and further network penetration testing.

This diagram lays down the building blocks of the FortiDAST vulnerability assessment and penetration testing service.



FortiDAST uses web Crawler and Fuzzer techniques to detect and scan your web applications for vulnerability assessment. The Common Vulnerability Scoring System (CVSS), Exploit Prediction Scoring System (EPSS), and Open Web Application Security Project (OWASP) Top 10 2021 are employed to assess the severity of vulnerabilities and identify security risks to web applications. The vulnerability assessment result is presented in a comprehensive dashboard and customized, downloadable reports with graphical representation and visualization of statistics.

**Note:** FortiDAST supports scanning both external (public) and internal (private) assets.

The following are some of the key features of FortiDAST.

- The web application scanning is comprehensive and provides accurate vulnerability assessment for a complete view of security risks.
- The automated scanning process allows you to simply and swiftly evaluate all of your web applications, reducing manual intervention.
- The scanning process is completely non-intrusive to prevent inactivity and disruptions; you can include additional headers to be included in the scan.

- A comprehensive dashboard as a combination of interactive chart and list based statistics. The dashboard provides detailed insight into the scanned web applications.
- Support for automatic upload of scan reports to WAF (FortiWeb) and rules generation.
- You can integrate FortiDAST plugin with Jenkins and with GitLab for Continuous Integration/Continuous Deployment to trigger automated vulnerability assessment scans.

**Note:** The term *asset* used henceforth in this document implies the web site that you are scanning.

FortiDAST implements/uses the following modules for vulnerability assessment.

- [Crawler Module](#)
- [Fuzzer Module](#)
- [Attack Chaining Module](#)

## Crawler Module

The web Crawler systematically crawls the web server asset to locate paths that are inputs to the fuzzer modules. It uses the quick and full scan modes. These modes are configurable, see [Configuring the Scanner on page 22](#). A quick scan is fast mode scanning that provides vulnerability assessment based on limited testing/scraping of the static pages of your asset. These pages are scraped by searching and extracting URLs from HTML tags and attributes. For example, the following tag which defines a hyperlink with `href` attribute.

```
<a href="http://example.com">
```

A Full scan provides vulnerability assessment based on complete testing/scraping of the static and dynamic pages of your asset. The Crawler also performs browsing simulation such as clicking of buttons, links, and images to test the interaction between the dynamic pages and the browser. This mode of vulnerability assessment takes longer.

The Crawler also uses websocket endpoints to collect relevant information for the fuzzer modules to assess vulnerabilities.

### Crawler Timeout

The Crawler times out after 5 hours, that is, it stops crawling your asset after 5 hours. If your asset is very large, you might obtain only partial scanning result.

### Inconsistent Crawler Result

The following are some reasons that might cause inconsistent crawling results.

- Dynamic contents: Forums and access logging.
- Redirections: HTTP redirects to HTTPS and redirection to WWW.
- Inconsistent response time: Presence of too much content affecting the response and loading time.
- Intermediate third party security product: Web Application Firewall (WAF) blocking some requests.

## Fuzzer Module

The FortiDAST also uses reconnaissance engine to provide server (host) and web service or application related information to the fuzzer modules to optimize and enhance vulnerability scanning. The Fuzzer modules scan the following to detect vulnerabilities.

- URLs in your web server asset

- Asynchronous requests for **Server Side Request Forgery**, **Remote Code Execution**, **XSS (Cross site scripting)**, **Path Traversal**, **File Inclusion**, **Server-Side Template Injection**, **XML external entity (XXE) injection**, **NoSQL**, **Open Redirect**, **SSTI**, and **LDAP**.
- Blind injection using the C2 server for **XSS (Cross site scripting)** and **File Inclusion**.
- REST APIs for **Server-Side Template Injection**, **Remote Code Execution**, **XSS (Cross site scripting)**, **Path Traversal**, **File Inclusion**,
- JSON data and HTTP methods.
  - GET, PUT, and PATCH for SSRF, RCE, XSS, and NoSQL
  - POST and GET for Path Traversal
  - POST for SQLi and XSS
  - File Inclusion for GET, POST, PUT, and PATCH
- XML data format
  - RCE, File Inclusion, NoSQLi, XSS, SSRF for GET, POST, PUT, and PATCH
  - Path Traversal for GET
  - FuzzerType2 for POST
- JSON Web Tokens (JWT)

This table describes the various Fuzzer modules used for vulnerability scanning.

Vulnerability Category	Vulnerability Description & Fuzzer Modules
<b>Injection</b>	<p><b>Remote Code Execution</b> - Scans if the provided URL together with other scan parameters are vulnerable to exploits due to command injection faults.</p> <p><b>Server-Side Template Injection</b> - Scans if the web application uses server-side template and if injecting malicious payload into the template can be executed.</p> <p><b>File inclusion</b> – Scans if the provided URL is vulnerable to dynamic file inclusion which occurs when the target contains procedures that use user-supplied file path input without proper validation.</p> <p><b>LDAP Injection</b> - Scans if the web application is vulnerable to LDAP injection attacks that occur when the LDAP statements based on user input are modified using a local proxy.</p> <p><b>NoSQL Injection</b> - Scans if the web application is vulnerable to malicious queries aimed to modify/alter the NoSQL database when the application communicates directly with the database.</p> <p><b>SQL Injection</b> - Scans if the web application is vulnerable to malicious SQL queries through unsanitized user input exposing sensitive information.</p> <p><b>XPATH Injection</b> - Scans if the web application is vulnerable to malicious Xpath queries through unsanitized user input exposing sensitive information.</p> <p><b>Code Injection</b> - Scans if the web application is vulnerable to HTML, PHP, and classic ASP injection attacks.</p> <p><b>Server Side Includes (SSI) Injection</b> - Scans if the provided URL is vulnerable to SSI injection attacks, where an attacker can execute arbitrary code on the server by injecting SSI commands into the web application.</p> <p><b>XSS (Cross site scripting)</b> - Scans for XSS vulnerabilities by sending executable scripts (payloads) in the form of specially crafted user inputs to a target URL. If the scripts end up being executed, the target is considered to be vulnerable.</p>

Vulnerability Category	Vulnerability Description & Fuzzer Modules
<b>Broken Access Control</b>	<p><b>Insecure file upload and manipulation via WebDAV</b> - Scans resources and properties of a particular directory to know if it is possible to obtain a recursive directory listing of all the files and folders from the provided URL using WebDAV. WebDAV is disabled when not in use or directory browsing permissions are restricted.</p>
	<p><b>Open Redirect</b> – Scans if the provided URL accepts a user controlled input that specifies a link to an external site, and uses that link in a redirect.</p>
	<p>ORM Injection - Object Relational Mapping injection is an attack using SQL injection against an ORM generated data access object model.</p>
	<p><b>Expression Language (EL) / Object Graph Navigation Library (OGNL) Injection</b> - Scans for blind injection and detects escalation of vulnerability to RCE.</p>
	<p><b>Forced Browsing</b> - Scans if the resources that are not referenced by the web application can be accessed leading to unauthorized information gathering.</p>
	<p><b>Server Side Request Forgery</b> - Scans if the HTTP requests coming from server-side applications can be controlled and redirected to a malicious web page. The C2 server is implemented to detect these vulnerabilities.</p>
	<p><b>Indirect object referencing (IDOR)</b> - Scans for any broken access control between two logged-in credentials.</p>
<b>Cryptographic Failures</b>	<p><b>Cross-Site Request Forgery (CSRF)</b> - Scans if the web application is vulnerable to CSRF attacks, which occur when an attacker tricks a user into unknowingly executing actions on a web application that they did not intend to perform.</p>
	<p><b>Path Traversal</b> - Scans if the files and directories can be accessed outside the web root folder on the target web server via a controlled web application variable.</p>
	<p><b>JSON Web Tokens (JWT)</b> - Scans web applications with JWT for authentication bypass via flawed signature verification and algorithm confusion attacks.</p>
	<p><b>SSL tests</b> - Scans if the provided URL together with other scan parameters has a valid SSL/TLS-enabled version and if so, whether there is an automatic HTTP to HTTPS redirection when a user visits the HTTP version of the website.</p>
	<p><b>Weak Ciphers</b> - Scans for vulnerable cipher suites that do not provide sufficient security to web applications.</p>

Vulnerability Category	Vulnerability Description & Fuzzer Modules
<b>Security Misconfiguration</b>	<p><b>XML external entity (XXE) injection</b> - Scans if the web application is vulnerable to XXE injections by validating and filtering the XML documents before processing.</p> <p><b>Information Disclosure</b> - Scans and identifies sensitive information such as passwords, phone numbers, email addresses, secret finders using regular expressions, and banner grabbing vulnerabilities. It extracts information on static and rendered HTML pages</p> <p><b>CORS misconfiguration</b> – Scans if the provided URL allows Cross-Origin Resource Sharing. CORS is a browser mechanism which enables controlled access to resources located outside of a given domain. Misconfiguration may allow attackers to perform cross-domain based attacks.</p> <p><b>Security HTTP Headers</b> - Scans if the HTTP response has specific headers to increase the security of your application.</p> <p><b>Weak Password</b> – Scans if the provided URL is subjected to authentication bypass using a dictionary bruteforce attack.</p> <p><b>Suspicious Domains</b> – Scans If the provided URL is referencing to domains which are either expired or not registered.</p> <p><b>Excessive authentication attempts</b> - Scans for improper restriction of excessive authentication attempts by brute forcing the login page by continuously sending random usernames and passwords.</p> <p><b>Git Directory Exploitation</b> - Scans for potential Git directory exposures that could reveal sensitive source code and configuration information.</p>
<b>Vulnerable and Outdated Components</b>	<p><b>Known vulnerability</b> - Scans if the asset (provided URL together with other scan parameters) is using such components that are known to have vulnerabilities. For components with Common Platform Enumeration (CPE) values, this module also queries the National Vulnerability Database (NVD) to find all reported vulnerabilities for each component. Each vulnerability in NVD is associated with a unique Common Vulnerabilities and Exposure (CVE) ID.</p>
<b>Identification and authentication Failures</b>	<p><b>URL Session Token</b> - Scans if the session tokens in the provided URL are vulnerable to leaks and uses secure methods to store session tokens.</p> <p><b>Session Fixation</b> - Scans if the value of the session cookie can be overwritten with an existent session ID. It ensures that a new session cookie is generated upon authentication.</p> <p><b>Mitigation against bruteforce attacks</b> - Scans for any protection mechanism against brute force attacks.</p> <p><b>Lack of session invalidation upon logout and session timeout</b> - Scans for insufficient inactivity session expiration (idle timeout of 15 minutes) and insufficient session invalidation on user logout (user logout function invalidates user session).</p>

Vulnerability Category	Vulnerability Description & Fuzzer Modules
<b>Software and Data Integrity Failures</b>	<p><b>Untrusted Data Deserialization</b> - Scans for vulnerabilities related to deserialization of Untrusted PHP/Java data. Serialized objects are not accepted from untrusted sources.</p> <p><b>Malware Detection</b> - Scans the provided URL to identify and prevent potential malware attacks on the web application. Malicious actors may target web applications with the objective of hosting malware, compromising sensitive user information, or inflicting various forms of damage, such as ransomware attacks.</p>
<b>Insecure Design</b>	<p><b>Unrestricted file upload</b> - Scans for insufficient validation of the name, type, contents, or size and headers of uploaded files.</p> <p><b>HTTP request smuggling</b> - Scans to detect HTTP request smuggling attack based on the known generic payloads.</p> <p><b>Authentication Bypass</b> - Scans for protection mechanism against malicious characters from user input.</p> <p><b>Web Cache Poisoning</b> - Scans for behavioral exploits of a web server and cache, to avoid serving a harmful HTTP response.</p> <p><b>Clickjacking</b> - Scans if the provided URL is vulnerable to clickjacking attacks. Clickjacking is a malicious technique where an attacker tricks a user into clicking on a hidden or disguised element on a webpage, which can lead to unintended actions or disclosure of sensitive information.</p>
<b>Improper Input Validation</b>	Scans if the web application properly validates user input, such as form data and query parameters, to prevent common input validation vulnerabilities.

## Attack Chaining Module

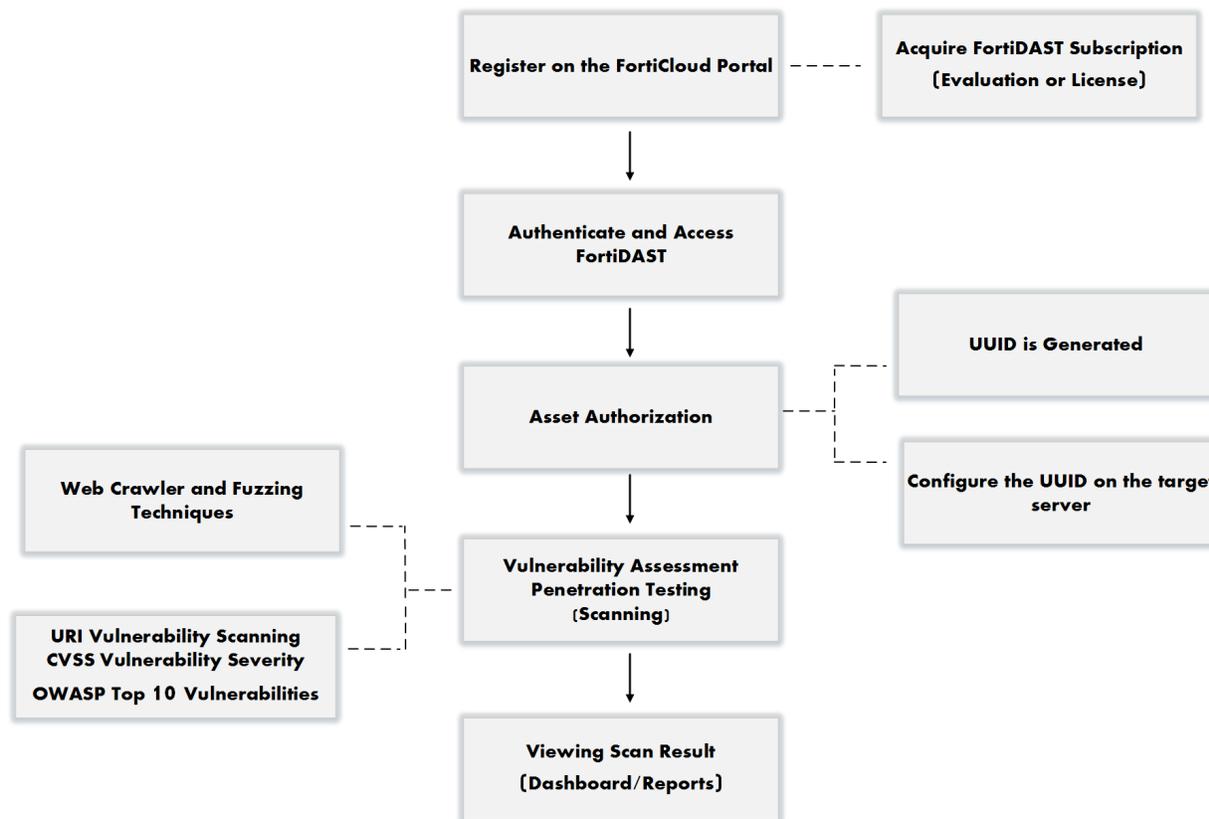
FortiDAST also uses the **Attack Chaining Module (ACM)** to perform deep scans combining a series of exploits from the initially discovered vulnerabilities, in case there is a possibility that more critical vulnerabilities are discovered, leading to a fully compromised asset. This release validates if the SSTI vulnerabilities identified in an asset can lead to RCE attacks.

The following categories of chaining attacks are supported.

- RCE with PHP Wrapper injections
- RCE with Template Injection for Node.JS EJS
- RCE with Template Injection for Node.JS Mustache
- RCE with Template Injection for Node.JS Underscore
- RCE with Template Injection for Node.JS Nunjucks
- RCE with Template Injection for Node.JS PUG
- RCE with Template Injection for Node.JS Velocityjs

## How FortiDAST Works

The FortiDAST user interface with its distinctive organization provides ease of accessibility and navigation. The interactive features allow you to scan your web applications effortlessly.



1. Register on the *FortiCloud* portal and access the FortiDAST user interface. See [Signing-on for FortiDAST on page 16](#).
2. Authorize your asset to perform vulnerability scanning. See [Asset Authorization on page 20](#).
3. Scan the asset for vulnerability assessment. See [Vulnerability Scanning on page 64](#)
4. View the vulnerability scanning results. See [Scans Overview \(Scan Result\)](#).

## User Interface Overview

The FortiDAST solution provides an interactive and easy to use GUI which enables easy vulnerability assessment. The GUI home page contains 3 sections accessible from the left navigation menu.

- 1  Dashboard
- 2  Scans Policy
- 3  Scans Overview

Section	Description
	The dashboard displays the overall statistics and details for a scanned asset. See <a href="#">Dashboard</a>
	The scans policy allows you add assets, authorize, and scan them for vulnerability scanning. The IP address/FQDN of web applications and the port are inputs on this page. The authorization and scan status of the assets are also displayed on this page. You can also configure the scanner for vulnerability assessment. See <a href="#">Vulnerability Assessment (scanning) of an Asset</a>
	The scans overview displays the detailed scan result for vulnerability assessment. <a href="#">Scans Overview (Scan Result)</a>

## Notifications

Click on the  icon to view the following notifications; notifications are retained for 7 days.

- License about to expire
- License expired
- Alerts configured in custom settings of email notifications
- WAF rules generated for the vulnerabilities selected
- The selected vulnerability is not detected for report generation
- Scheduled portal upgrade
- Portal upgrade completed

## Outbreak Alerts

The FortiGuard labs will constantly monitor for potential security vulnerabilities and will designate certain CVEs as outbreak alerts if they are being widely exploited. All the outbreak alerts for the top 5 scanned assets can be viewed in the [Dashboard > Outbreak Alerts](#) page and asset specific outbreak alerts can be viewed in the [Summary on page 74](#) and [Vulnerabilites](#) pages.

# Licensing

FortiDAST offers free (evaluation), paid (licensed) subscriptions, and the FortiCloud Premium trials. For more information see the [FortiCloud Data Sheet](#).

For advanced FortiDAST usage, you must purchase a license. Contact the Fortinet *Customer Support* team to acquire a license.

Evaluation	FortiCloud Premium Trial	Licensed
Valid for 60 days from the date of registration.	Valid for 364 days from the date of registration.	Valid for 364 days from the date of registration.

**Notes:**

- After the subscription expires, FortiDAST vulnerability assessment operations (asset authorization and scanning) are NOT available. You can ONLY view the scan result page and the reports.
- You will be able to delete your older assets and add new assets for scanning if your license has expired and you are renewing it. This applies only to subscribers who are renewing their licenses.

The following table describes FortiDAST services that are selectively available based on your subscription.

FortiDAST Service	Evaluation	FortiCloud Premium Trial	Licensed
Asset count	Allows you to perform vulnerability scanning for 1 asset.	Allows you to perform vulnerability scanning for 1 asset.	Allows you to perform vulnerability scanning for 10 assets per license.

**Note:** All features of FortiDAST are supported for the subscription types aforementioned.

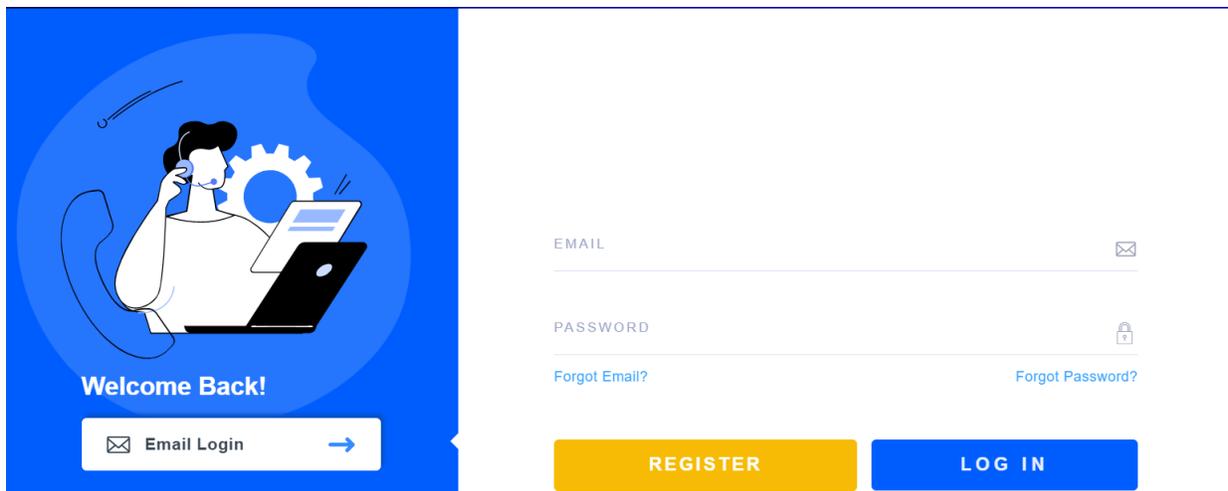
## Signing-on for FortiDAST

This release provides single sign-on support for FortiDAST along with FortiCloud suite of products. FortiDAST is accessible via the *FortiCloud* GUI - <https://support.fortinet.com> and <https://fortidast.com>. However, if you access <https://fortidast.com>, you are redirected to the FortiCloud login page.

- [Registering on FortiCloud on page 16](#)
- [Accessing FortiDAST on page 16](#)

## Registering on FortiCloud

Prior to using FortiDAST, you are required to register on the *FortiCloud* portal. Use the <https://support.fortinet.com> access link to register on the *FortiCloud* portal. A security code is emailed to the address specified during registration; use the code to complete registration and activate your account.



### Adding IAM Users

The Identity and Access Management (IAM) is a service to help you control access to FortiCloud portals and assets. You can use the portal to manage users, authentication credentials, and asset permissions. For more information, see [FortiCloud](#) documentation. Access the IAM service from the FortiCloud portal using the master FortiDAST account. To add an IAM user, see [Adding IAM Users](#).

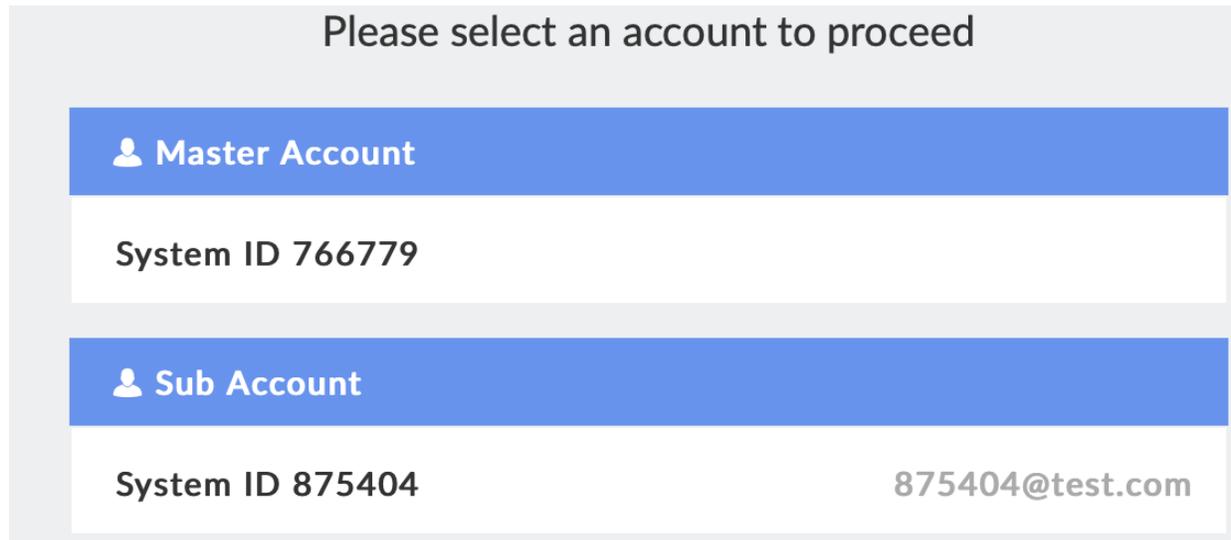
## Accessing FortiDAST

Any user registered on <https://support.fortinet.com> can access FortiDAST. Once you login into *FortiCloud*, click on **Services**, a banner with Fortinet products is displayed. Select **FortiDAST**. You are redirected to the

FortiDAST GUI, <https://fortidast.forticloud.com/>.

The associated sub-users of a *FortiCare* master account can also login into FortiDAST and select their master account for access. A sub-user can be part of multiple master accounts, the license available to use is based on the selected master account. (based on the availability limit).

When you login into FortiDAST, your master account and accounts on which you are configured as a sub-user are displayed; you are prompted to select one.



You can login into FortiCloud using your registered FortiCloud account details, **Email** and **Password** OR click **Sign in as IAM user**. Enter your registered IAM user credentials to login, the **Account ID** is that of the master account.

## External IDP Authentication

FortiDAST supports integration of third-party Identity Provider (IDP) services to log-in and manage networks. This feature is useful for enterprises that need to secure their user credentials and hence provision FortiDAST access through their own Identity Provider. The external IDP initiated Security Assertion Markup Language (SAML) assertion consisting of specific IDP attributes is used by FortiCloud/FortiDAST to verify the user account details and grant required access.

External IDP authentication is offered in conjunction with FortiCare and FortiAuthenticator. Contact the Fortinet *Customer Support* team to enable external IDP support and raise an enrollment request with the appropriate FortiCare accounts. After the enrollment is complete follow these setup procedures.

**Note:** Support for SAML 2.0 and IDP initiated assertion response is required.

- Create an IDP with SAML Service Provider Metadata. The following is an example where *company* is the unique name of your organization.
 

```
SP Entity ID http://customerss01.fortinet.com/saml-idp/proxy/
{company}/metadata/
SP Login URL https://customerss01.fortinet.com/saml-idp/proxy/
{company}/saml/?acs
Relay State https://customerss01.fortinet.com/saml-idp/proxy/{company}/login/
```
- Configure the SAML assertions with the *username* and *role* attributes for permission control in FortiCloud.

- Provide specific information to Fortinet, such as, the SAML Metadata file, company name, contact information, and the Fortinet master account that the IDP requires to connect to.
- Configure external IDP roles in FortiCloud to allow the required access to FortiDAST. See [Adding External IDP Roles on page 18](#).

After successful authentication on your Identity Provider, you are re-directed to the FortiCloud portal from where you access FortiDAST based on the configured roles.

## Adding External IDP Roles

Access the **Identity & Access Management (IAM)** service from the FortiCloud portal

1. Navigate to **Users > Add New > ADD IDP User** and click **Add IDP Role**.
2. Enter a unique **Role Name** and **Description** (optional).  
**Note:** The role name must exactly match the role attribute in the SAML assertion. For FortiDAST the role attributes defined are: **Admin, IDP\_ReadOnlyAll** and **IDP\_ReadWriteAll**.
3. Select an asset group from the **Asset Permissions** list.
4. Select the Permission profile. See, [Adding Permission Profiles](#).

**External IdP Role** ⓘ

Role Name	Status
Admin	Active
Description	

---

**PERMISSION SCOPE**

Asset Folder

My Assets

---

**PERMISSION PROFILE**

Profile Name

DAST\_AssetOnly\_Admin

---

**PERMISSION DETAILS**

Asset Management	FortiDAST
Access	Access
Access Type	Access Type
Additional Permission	Additional Permission
<p>✓</p> <p>Admin</p>	<p>✓</p> <p>Admin</p>

5. Click **Add Role**.

After the role is created, it is listed on the on the **Manage External IdP Roles** page. You can enable/disable or delete a created role. Select the role and click on the required option.

## Adding Permission Profiles

Access the **Identity & Access Management (IAM)** service from the FortiCloud portal. Configure the Cloud Management & Services permissions to enable access to FortiDAST Cloud.

1. Navigate to **Permission Profiles**.
2. Click **Add New**.
3. Click **Add Portal** and select FortiDAST from the list and click Add.
4. Configure the required permissions for FortiDAST:
  - a. Toggle **Access** to allow access to FortiDAST.
  - b. Select the required Access Type: Admin, Read Write, or Read-Write.

The screenshot shows the configuration page for a Permission Profile. It is divided into two main sections: 'BASIC INFO' and 'PERMISSION PROFILE'.

**BASIC INFO**

- Permission Profile Name:** DAST\_AdminProfile
- Status:** Active
- Description:** Enter Permission Profile Description

**PERMISSION PROFILE**

There is an 'Add Portal' button in the top right corner of this section.

The 'FortiDAST' profile is shown with the following settings:

Access	Access Type	Additional Permission
<input checked="" type="checkbox"/>	<input checked="" type="radio"/> Admin <input type="radio"/> Read/Write <input type="radio"/> Read Only	

# Vulnerability Assessment (scanning) of an Asset

Perform these procedures to perform vulnerability assessment and penetration testing on your web applications.

**Note:** Ensure that the firewall allows the communication between the target asset and FortiDAST (IP: 35.222.40.56).

- [Asset Authorization](#)
- [Schedule Scan](#)
- [Configuring the Scanner](#)
- [Automation](#)
- [Exploit Engine](#)
- [Vulnerability Scanning](#)
- [FortiDAST Proxy Server](#)
- [CI/CD Tools](#)
- [Settings](#)

## Asset Authorization

An asset must be successfully authorized to perform vulnerability scanning. The authorization process verifies asset ownership.

1. Navigate to **Scans Policy** and click **New Scan**. Enter the **IP address/FQDN** and the **Port** of the asset.

The screenshot shows a 'New Scan' configuration form. It has the following fields and controls:

- Scan Name:** A text input field containing 'Scan1'.
- URI:** A text input field containing 'https://192.168.1.1'.
- Port:** A text input field containing '443'.
- Uuid:** A text input field containing a long alphanumeric string.
- Schedule Scan:** A button with a radio button icon.
- Configuration:** A button with a plus sign icon.

The maximum number of assets you can scan is displayed on the GUI as per your subscription. See [Licensing](#).

2. A unique asset token, UUID, is generated for each asset and is displayed on the page. Copy the UUID and configure it in any of the following methods.

- Create a `<UUID>.html` file in the webroot of the asset's web server with no content. For example, a UUID `ded8024f-54c1-4bd2-8d82-9ad30bf3e35e` is generated for your asset, create an empty file named `ded8024f-54c1-4bd2-8d82-9ad30bf3e35e.html`.
- Create a `forti-uuid.html` file in the webroot of the asset's web server with `<forti-uuid hidden><UUID></forti-uuid>` as the content. For example, a UUID `ded8024f-54c1-4bd2-8d82-9ad30bf3e35e` is generated for your asset, create a file named `forti-uuid.html` with `<forti-uuid hidden>ded8024f-54c1-4bd2-8d82-9ad30bf3e35e</forti-uuid>` as content.
 

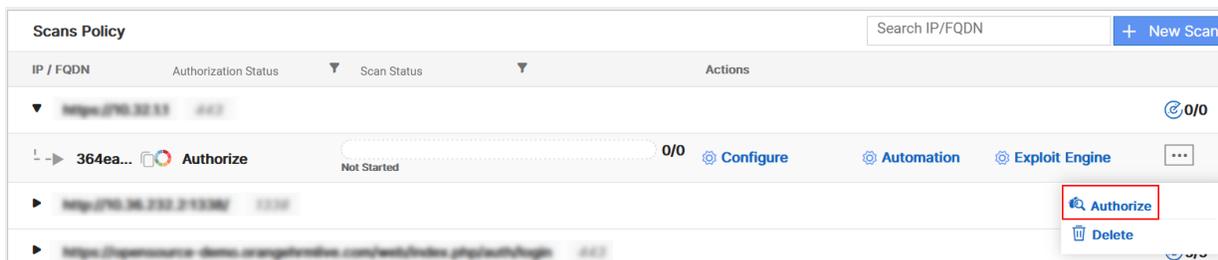
```
#cat forti-uuid.html
<forti-uuid hidden>ded8024f-54c1-4bd2-8d82-9ad30bf3e35e</forti-uuid>
```
- Store the UUID as a custom attribute/create a DNS Text record with the data, `forti-uuid=<UUID>` in the domain management page.

@	TXT	1h	"forti-uuid=a038f133-561e-4c29-8cb1-752d2b49f39d"	Delete	Edit
---	-----	----	---	--------	------

Add the DNS text record as per the configured asset URL. Consider the following examples.

- If the configured asset is `https://example.com` then add the DNS text record in the root domain, `example.com`.
- If the configured asset is `https://web.example.com` then add the DNS text record in the sub-domain, `web.example.com`. Authorization fails if the DNS text record is added in the root domain, `example.com`.

3. Click on the **Actions** icon -  and select **Authorize**. The status of the authorization process is displayed.



**Note:** The licensing mechanism does not allow you to modify or delete an asset after it is authorized.

Any variation in the FQDN, IP address, or port is considered as a separate asset. The following are some examples of such variations that are treated as separate assets.

- `http://example.com`
- `http://fortinet.example.com`
- `http://example.com:9020`
- `http://10.34.222.202:8080`

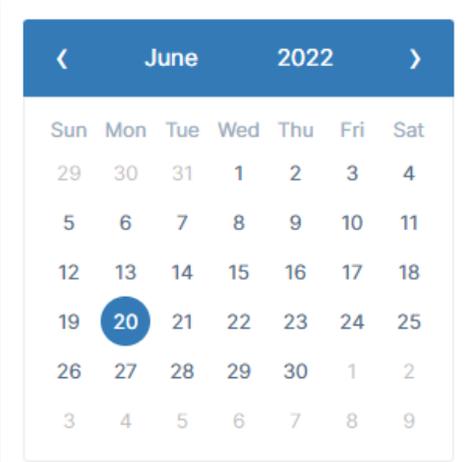
If you have already authorized one of the root domains in your current license, you do not need to configure DNS TXT records or UUID for any new FQDN that is a subdomain for an existing authorized asset. You will not be required to authorize the subdomain again and the authorization will be bypassed.

For example, if you have authorized the root domain `fortinet.com`, you do not need to configure DNS TXT records or UUID for the subdomains `fortinet.com/subdomain1` and `fortinet.com/subdomain2`.

## Schedule Scan

You can schedule recurring vulnerability scans for your asset that run automatically at a configured time. Select the date of the scan in the displayed calendar and set the time.

Schedule Scan



Time Selection ( UTC )

10 AM

Recurrence

Repeat Every Weekly

Repeat On S M **T** W T F S

You can enable **Recurrence** to configure the scan frequency with the following combinations.

Frquency	Description
Daily	Scan is scheduled to run once every day at the set time.
Weekly	Scan is scheduled to run once every week at the set time.
Monthly	Scan is scheduled to run once every month on the selected day as per the available options at the set time.



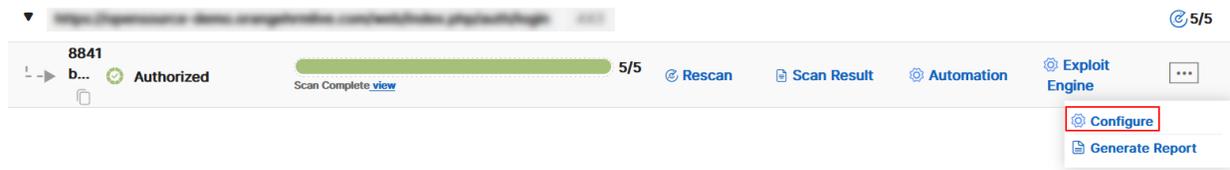
- If the time is not set, the vulnerability scan occurs at the default time of 12 AM (midnight).
- For targets requiring multi-factor authentication, scheduled scans cannot capture the required tokens or responses. Use a standard scan to ensure proper scanning of authenticated URLs.

## Configuring the Scanner

You can configure the scanner in the **New Scan** page ([Asset Authorization](#)) or click **Configure** in the **Scans Policy** page.



To update configuration for a scanned asset, click the actions icon and select **Configure**.



The configuration page allows you to configure and manage the following features.

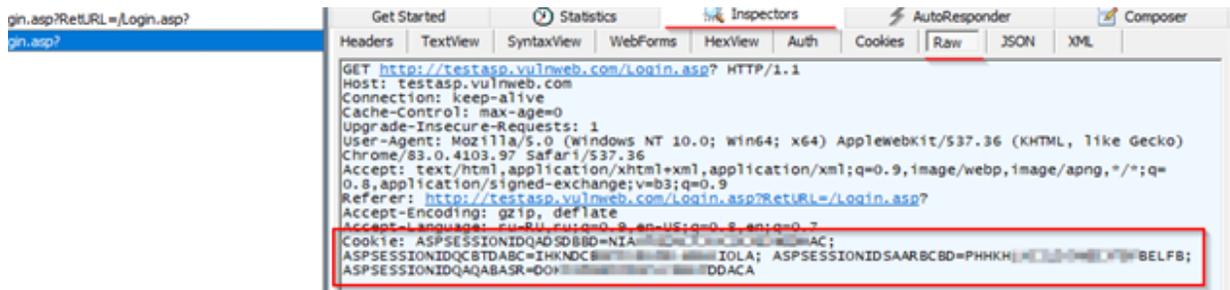
- HTTP (Authentication via Cookies)
- Authentication (via GUI)
- Coverage
- Forced Browsing
- Inclusion/Exemption List
- WAF Configuration
- Web Filter
- API Crawling
- Jira Integration
- IDOR
- FortiDAST Proxy
- Custom Notifications

## HTTP (Authentication via Cookies)

You are required to extract cookies from the website.

**Request Headers** - Use any web debugging application to extract cookies from the asset. Copy and paste the cookie in this field. The required format for the header is *Content-type :Value; Cookie :Value*. Consider the following example.

The cookie is extracted using a web debugger application.



Copy the cookie and paste it in the **Request Headers** field.

```
Cookie: ASPSESSIONIDQADSDBBD=NIA...AC;
ASPSESSIONIDQCBTDABC=IHKND...IOLA;
ASPSESSIONIDSAARBCBD=PHHKH...BELFB;
ASPSESSIONIDQAQABASR=DO...DDACA]
```

You can additionally configure the following **HTTP proxy Server** fields.

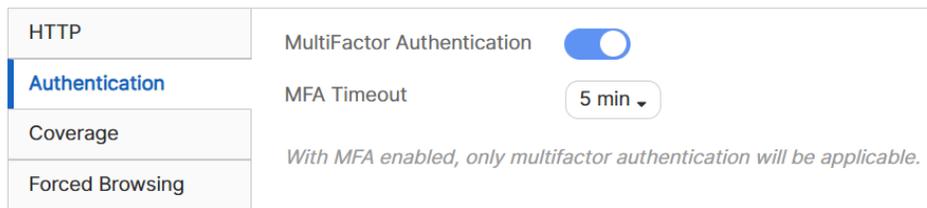
- **Address** - The IP address of the HTTP proxy server in use.
- **Allow Insecure SSL** - Allow insecure SSL content for vulnerability assessment.

## Authentication (via GUI)

This is required for assets that have authentication enabled.

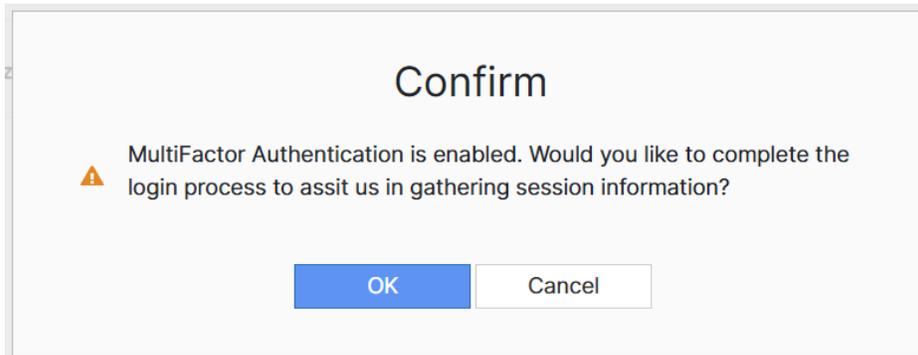
- **Multi-Factor Authentication:** You can configure a timeout for multi-factor authentication (MFA) for target website logins. This feature is useful when the login process requires additional steps, such as CAPTCHA or one-time password entry.

To enable MFA, toggle Multi-Factor Authentication and configure the timeout setting. You can set the timeout between 3 and 10 minutes. This timeout is the duration allowed to complete the MFA login process.



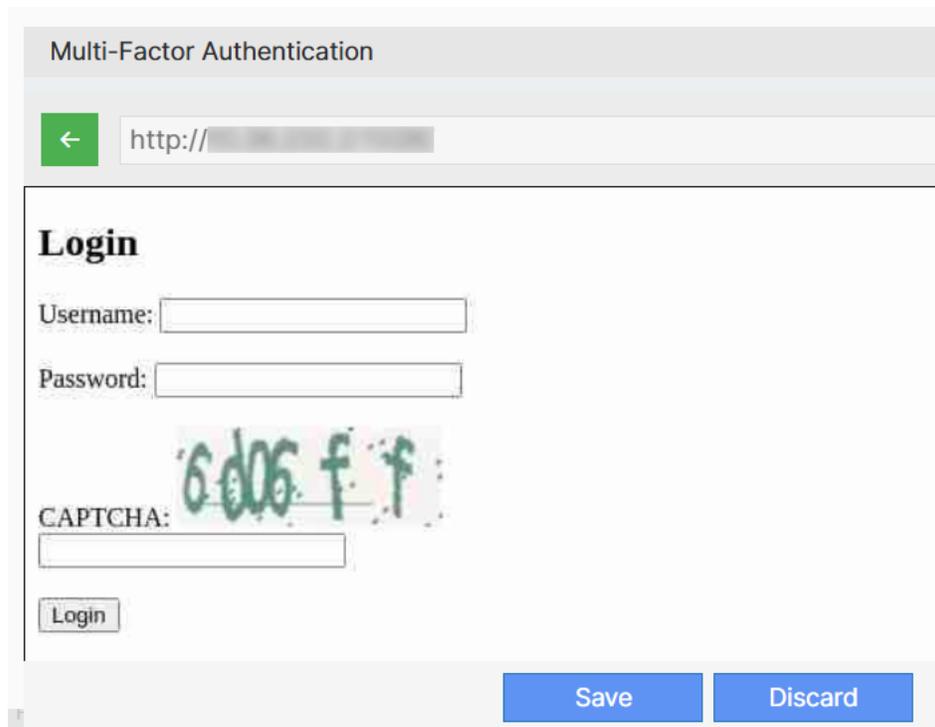
The screenshot shows a configuration panel for HTTP Authentication. On the left is a sidebar with tabs: HTTP, Authentication (selected), Coverage, and Forced Browsing. The main area shows 'MultiFactor Authentication' with a blue toggle switch turned on. Below it, 'MFA Timeout' is set to '5 min' with a dropdown arrow. A note at the bottom reads: 'With MFA enabled, only multifactor authentication will be applicable.'

When you initiate a scan with MFA enabled, click **OK** when prompted to proceed with the MFA login.



The target URL opens in an embedded browser. Complete the login process, including any required MFA steps, within the configured timeout period. Click **Save**.

If you do not click **Save** or click **Discard** after completing the MFA login, the scan will continue without session information.



Multi-Factor Authentication

← http://

**Login**

Username:

Password:

CAPTCHA: 6d06 f f

Login

Save Discard



When MFA is enabled, only multi-factor authentication is applicable.

- 
- **Web Authentication** or **HTTP Authentication** - This is required for websites that have a login form and require credentials to be validated. For a website that has authentication enabled, you are required to enter the username and password for the entire web application to be scanned. HTTP Basic, Digest, and NTLM authentication frameworks are supported.

- **Login URL** - The login URL of your asset.
- **Logout URL** - The logout URL of your asset.

HTTP Authentication

User Name:	<input type="text" value="bee1@gmail.com"/>
Password:	<input type="password" value="....."/>

Web Authentication

User Name:	<input type="text" value="bee1@gmail.com"/>
Password:	<input type="password" value="....."/>
Login URL:	<input type="text" value="http://testasp.vulnweb.com/login"/>
Logout URL:	<input type="text" value="http://testasp.vulnweb.com/logout"/>

## Coverage

You can select/deselect OWASP Top 10 - 2021 categories of your choice to use for vulnerability assessment during scanning. For each of the selected OWASP 10 category, you can enable specific Fuzzer modules/sub-categories to fine tune the scan as per your network requirements.

Asset Crawling Scope:	<input type="text" value="Same Host"/>																								
Scan Flag:	<input type="text" value="Full Scan"/>																								
Category Selection:	<table><tr><td><input checked="" type="checkbox"/></td><td>Injection</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Broken Access Control</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Identification and Authentication Failures</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Cryptographic Failures</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Security Misconfiguration</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Software and Data Integrity Failures</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Vulnerable and Outdated Components</td><td>&gt;</td></tr><tr><td><input checked="" type="checkbox"/></td><td>Insecure Design</td><td>&gt;</td></tr></table>	<input checked="" type="checkbox"/>	Injection	>	<input checked="" type="checkbox"/>	Broken Access Control	>	<input checked="" type="checkbox"/>	Identification and Authentication Failures	>	<input checked="" type="checkbox"/>	Cryptographic Failures	>	<input checked="" type="checkbox"/>	Security Misconfiguration	>	<input checked="" type="checkbox"/>	Software and Data Integrity Failures	>	<input checked="" type="checkbox"/>	Vulnerable and Outdated Components	>	<input checked="" type="checkbox"/>	Insecure Design	>
<input checked="" type="checkbox"/>	Injection	>																							
<input checked="" type="checkbox"/>	Broken Access Control	>																							
<input checked="" type="checkbox"/>	Identification and Authentication Failures	>																							
<input checked="" type="checkbox"/>	Cryptographic Failures	>																							
<input checked="" type="checkbox"/>	Security Misconfiguration	>																							
<input checked="" type="checkbox"/>	Software and Data Integrity Failures	>																							
<input checked="" type="checkbox"/>	Vulnerable and Outdated Components	>																							
<input checked="" type="checkbox"/>	Insecure Design	>																							

**Asset Crawling Scope-** This feature crawls and scans only those URLs that are on the same domain/host as the target asset. Specify the scope of crawling URLs for the target asset whether on the **Same Host** or **Same Domain**.

**Note:**

- The following sub-categories are enabled by default. This setting cannot be modified.
  - A3 Sensitive Data Exposure - Information Disclosure and SSL Tests
  - A5 Broken Access Control - Forced Browsing
  - A6 Security Misconfiguration - CORS Misconfiguration, Security HTTP Headers, and Suspicious Domains
- The Exploit Engine and Forced Browsing configurations override the scan coverage configurations in the scan result data.

**Scan Flag** - Configures the type of scan, **Quick Scan** or **Full Scan** (default).

Fuzzer Modules	Quick Scan	Full Scan
Cross-Site Scripting	Uses a limited set of payloads.	Uses the full set of payloads.
Server-Side Template Injection		
Local/Remote File Inclusion		
Open Redirection		
Weak Form Password	Uses limited dictionary for brute force vulnerabilities.	Uses full dictionary for brute force vulnerabilities.
Suspicious Domains	<= 30 web domains are scanned for vulnerabilities.	All web domains found are scanned for vulnerabilities.
Information Disclosure	Extracts information on static HTML and scans for banner grabbing vulnerabilities.	Extracts information on static and rendered HTML, scans for banner grabbing vulnerabilities and secret finders using regular expressions.
Security Headers	Employs same scanning techniques for both quick and full scan.	
Cross-Origin Resource Sharing Misconfiguration		
Known Vulnerabilities	Detects components based on HTTP headers, HTML meta tags, HTML content, and script URLs.	Additional detection of JavaScript components via their version functions.

Fuzzer Modules	Quick Scan	Full Scan
Session Fixation	Uses HTTP library to set cookies in the request and analyze if there is a <i>set-sookie</i> in the response.	Uses Chromedp to set cookies in the browser and analyze its values after the request is received. Performs <i>HTTPOnly</i> flag check for the session cookie.
SSL/TLS Tests	Employs same scanning techniques for both quick and full scan.	
URL Session Token	Full scan uses better thresholds than quick scan.	
NoSQL Injection	Uses basic form scan, delay checks, and database error checks.	Uses full payload scan, delay checks, and database error checks.
XML external entity (XXE) injection	Full scan detects blind vulnerabilities.	
LDAP Injection	Checks error messages.	Checks error messages and performs boolean based checking.
Weak Ciphers	Uses a few checks for bad bulk ciphers only.	Uses all checks for weak algorithms (ciphers-key exchanges-hashes)
Path Traversal	Uses simple dot-slash pair checks.	Uses encoded dot-slash pairs checks.
Remote Command Execution	Uses echo commands.	Uses echo, cat, type, wget, and curl commands.
XPATH Injection	Employs same scanning techniques for both quick and full scan.	
SQL Injection	Processes a maximum of 100 requests. Boolean based blind SQL Injection	Processes unlimited requests. Boolean and time based blind SQL Injection
ORM Injection -	Processes a maximum of 100 requests. Boolean based blind SQL Injection	Processes unlimited requests. Boolean and time based blind SQL Injection
Expression Language (EL) / Object Graph Navigation Library (OGNL) Injection	Detects by computing the product of two random numbers.	Detects by computing the product of two random numbers. Detects blind injection. Detects escalation of vulnerability to RCE.
IDOR	NA	Verifies broken access control between two logged-in credentials. Asynchronous (fetch, XHR) POST requests with parameters and API calls must be accessible only by the session authorizing the original request.

Fuzzer Modules	Quick Scan	Full Scan
Mitigation against brute force attacks	Detects if the target has a protection for brute-force attacks.	
Lack of session invalidation upon logout and session timeout	Detects insufficient inactivity session expiration (idle timeout of 15 minutes) and insufficient session invalidation on user logout (user logout function invalidates user session).	
ACM	NA	Validates if the SSTI vulnerabilities identified in an asset can lead to RCE attacks.
Unrestricted file upload	Uploads different file extensions to the target web server with less payloads.	Uploads different file extensions to the target web server with additional payloads.
HTTP request smuggling	<i>Content-Length</i> and <i>Transfer-Encoding</i> variant payloads are used for scanning.	Additional variant payloads are used for scanning.
Excessive authentication attempts	Uses brute force by continuously sending random usernames and passwords to scans for improper restriction of excessive authentication attempts.	
Authentication bypass	Detects malicious attacks using simple HTTP request.	Detects malicious attacks using the Google Chrome browser.
Web cache poisoning	Detects malicious attacks using simple HTTP request.	Detects malicious attacks using the Google Chrome browser.
Code injection	Scans form and query via the golang HTTP client.	Scans header, cookie, form and query via Google Chrome browser..
Malware detection	Employs same scanning techniques for both quick and full scan.	
Clickjacking	Scans for x-frame-options.	Target is loaded in an iframe.

## Forced Browsing

You can test if restricted resources are accessible leading to sensitive information exposure using a built-in word list with SAP technology. You can also upload a custom word list.

Use Forced Browsing to check if Restricted resources are accessible

UPLOAD

Current Wordlist: No wordlist uploaded

SAP x

**Note:** This feature is available only in the full scan mode.

## Inclusion/Exemption List

You can exclude one/multiple specific URIs from scanning, for example, you can exclude common user actions such as *Logout* and *Login* from the vulnerability scanning. Add the paths or upload a list in the *.txt* format.

Type : **Exemption** ▼

Add exemption paths for http://10.36. [redacted] :8301

http://10.36. [redacted] :8301/logout.aspx **ADD**

Upload exemption List : **CHOOSE FILE ..**

Exemption List ▼

**DELETE ALL** **DOWNLOAD ALL**

http://10.36. [redacted] :8301/login.aspx 

Inclusion List >

You can include one/multiple specific URIs during scanning, for example, you can include paths that are not found automatically by the Crawler such as those without visible links on the web pages. Add the paths or upload a list in the *.txt* format.

Type : **Inclusion** ▼

Add inclusion paths for http://10.36. [redacted] :8301

http://10.36. [redacted] :8301/includepath2 **ADD**

Upload inclusion List : **CHOOSE FILE ..**

Exemption List >

Inclusion List ▼

**DELETE ALL** **DOWNLOAD ALL**

http://10.36. [redacted] :8301/includepath1 

## WAF Configuration

Configure the FortiWeb details in this tab to generate WAF rules. Provide the **WAF URL** (FortiWeb IP address/URL), FortiWeb **Username** and **Password**, and the **Administrative Domain** (VDOM name). Click **Validate** to authenticate the FortiWeb credentials. After the scan is complete, the XML is generated and rules are created in FortiWeb dynamically based on the **Actions** configured and the WAF supported **Vulnerability Selection**.

**Note:** WAF is supported for FortiWeb on-prem only.

WAF type :

---

WAF URL:

Username:

Password:

Administrative Domain:

VALIDATE

Action: High

Medium

Low

Vulnerability Selection:

- Injection >
  - Broken Access Control
  - Identification and Authentication Failures
  - Cryptographic Failures
  - Security Misconfiguration
  - Software and Data Integrity Failures
  - Vulnerable and Outdated Components
  - Insecure Design
- [Unselect All](#)
- File Inclusion
  - Remote Code Execution
  - Server Side Template
- Injection
- SQL/XPath Injection
  - noSQL injection
  - LDAP injection
  - Expression Language
- Injection
- XSS
  - Open Redirection
  - webdav

Notifications for the newly created rules are sent through email and through the notification icon in FortiDAST.

You can generate WAF reports manually from **Scans Overview > Summary > Overview** section. See [Exporting Scan Result to FortiWeb WAF](#).

## Web Filter

FortiDAST integrates the FortiGuard web filtering feature to report malicious or questionable out-of-scope domain links identified on the asset. This feature monitors and reports the assets under the **Suspicious Domains** category of **Security Misconfiguration** vulnerabilities.

Web Filter Lookup :

**Adult/Mature Content** >

**Potentially Liable**

**Security Risk**

Unselect All

**Alternative Beliefs**

**Abortion**

**Other Adult Materials**

**Advocacy Organizations**

**Gambling**

**Nudity and Risque**

**Pornography**

**Dating**

**Weapons (Sales)**

**Marijuana**

**Sex Education**

**Alcohol**

**Tobacco**

**Lingerie and Swimsuit**

**Sports Hunting and War Games**

## API Crawling

FortiDAST allows crawling and scanning REST API endpoints for vulnerabilities considering their potential attack vectors. This feature crawls and scans REST APIs that are on the same domain/host as the target asset, however, sometimes the APIs are hosted on a different domain. In this case, FortiDAST enables crawling on one other domain outside the scope of the target asset.

Enable API crawling and scanning to configure the following parameters.

- **API Definition URL** - Enter the API definition URL **OR** upload an API definition file. This file contains details such as API host servers, parameters required for API requests, expected responses, security scheme of the API endpoints and so on. This file is in JSON, YAML, or WADL format. FortiDAST crawler parses the API definitions to identify the paths for each API endpoint and the fuzzer modules scan them for vulnerabilities.

If neither the API definition URL nor the API definition file is provided then FortiDAST auto-discovers API endpoints from common locations and Fetch/XHR data and then crawls the discovered endpoints.

- **API Crawling Scope** - Specify the scope of crawling the REST APIs whether on the **Same Host, Same Domain**, or **Other** domain. In case the APIs are in a different domain, provide the following domain related information.
  - **Domain Name**
  - **HTTP Authentication** credentials
- **API Token URL** - Specify API token to access some API endpoints in the following ways.
  - **API Token Manually** - Specify the API token header including the authentication key/token/cookie and other required information.
  - **API Token URL** - Select this option to obtain the authentication key from FortiDAST. Enter the **API Token URL**, the **API Token Header**, optionally enter the **API Token Prefix** and **Request Body** including the parameters and their values.

API crawling and scanning :

API Definition URL ⓘ

API Crawling Scope ⓘ

API TOKEN URL ⓘ

API Token Header ⓘ

API Token Prefix ⓘ

Request Body ⓘ

## Jira Integration

Toggle the **Enable Jira Plugin** option to enable Jira integration. Select the Jira project to map to the target asset and assign the severity for the different categories. Once Jira plugin enabled, Jira bugs are created and updated with subsequent run of scans for the same target and is updated (bug removed) if the vulnerability no longer exists. For more information, see [Jira](#).

Enable Jira Plugin:

Project:

**PB x** ▼

Severity:

**Medium x High x Critical x Low x** ▼

Vulnerability:

- Injection** >
- Broken Access Control
- Identification and Authentication Failures
- Cryptographic Failures
- Security Misconfiguration
- Software and Data Integrity Failures
- Vulnerable and Outdated Components
- Insecure Design

**Unselect All**

- File Inclusion
- Remote Code Execution
- Server Side Template Injection
- SQL/XPath Injection
- noSQL injection
- LDAP injection
- Expression Language Injection
- XSS
- Open Redirection
- webdav

## IDOR

IDOR (Insecure Direct Object Reference) is a vulnerability that allows unauthorized access to resources in the system directly, for example database records or files. This occurs when an application provides direct access to objects based on user supplied input, enabling attackers to bypass authorization and access. You can configure access credentials for HTTP, API, and web authentication.

IDOR - HTTP Authentication

User Name:

elastic

Password:

••••••••

IDOR - API Authentication

User Name:

Password:

IDOR - Web Authentication

User Name:

Password:

IDOR - Headers

IDOR - Request Headers

Content-type:Value  
Cookie:Value  
...

IDOR - API Request Headers

x-auth-token:AUTHTOKEN1002

## FortiDAST Proxy

FortiDAST enables scanning of internal assets in your network (non-public IP addresses) using a proxy server. For more information, see [FortiDAST Proxy Server](#).

Enable the FortiDAST Proxy server feature and click **Copy** to copy the Docker compose file (*docker-compose.yml*).

Enable DAST Proxy:

Autonomous DevOps

```
version: '3.7'

# Proxy modes:
# Exit After Scan: this is the default mode in which the proxy
# container will exit once the scan is completed. The user has
# to start the container again to run the next scan.
#
# Daemon mode: In this mode of proxy, the container will be running
# in the background and always connected to the cloud through secure
# tunnels.
#
# How to enable this mode?
# Follow the instructions mentioned below referenced with 'Daemon mode proxy'.

services:
```

OK

## Custom Notifications

Email notifications are sent per asset indicating the scan progress such as the start, severity findings, and scan completion. Ensure that Email Notifications are enabled in the global settings. See [Email Notification](#).

### Custom Notifications

\* Please make sure Custom Email Notifications is enabled under Settings



**Scan Started**

Receive an email when a scan starts.



**Scan Finished**

Receive a summary email when a scan is finished.



**Critical Severity Findings**

Receive an email every time a critical severity finding is detected.



**High Severity Findings**

Receive an email every time a high severity finding is detected.



**Medium Severity Findings**

Receive an email every time a medium severity finding is detected.

## Automation

The *Replay with Automation* section in FortiDAST's *Configuration* page provides options to enhance scan accuracy and coverage. You can automate complex workflows, capture login sequences, and leverage existing automation scripts to improve data collection, discover hidden URLs, and replay core business logic. This automation allows FortiDAST to more effectively identify vulnerabilities within your web applications.

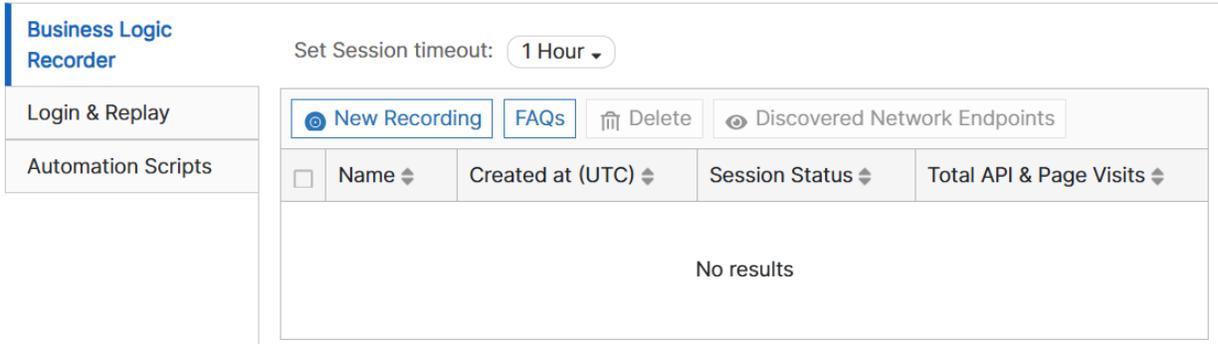
You can configure automation settings by clicking **Automation** on the **Scans Policy** page, or by clicking **Configure** and scrolling down to the **Replay with Automation** section.



- [Business Logic Recorder](#)
- [Login and Replay](#)
- [Automation Scripts](#)

## Business Logic Recorder

The Business Logic Recorder allows you to browse and record navigation within your web application or target, enabling FortiDAST to discover additional navigation paths, URIs, cookies, and APIs for enhanced scan coverage.



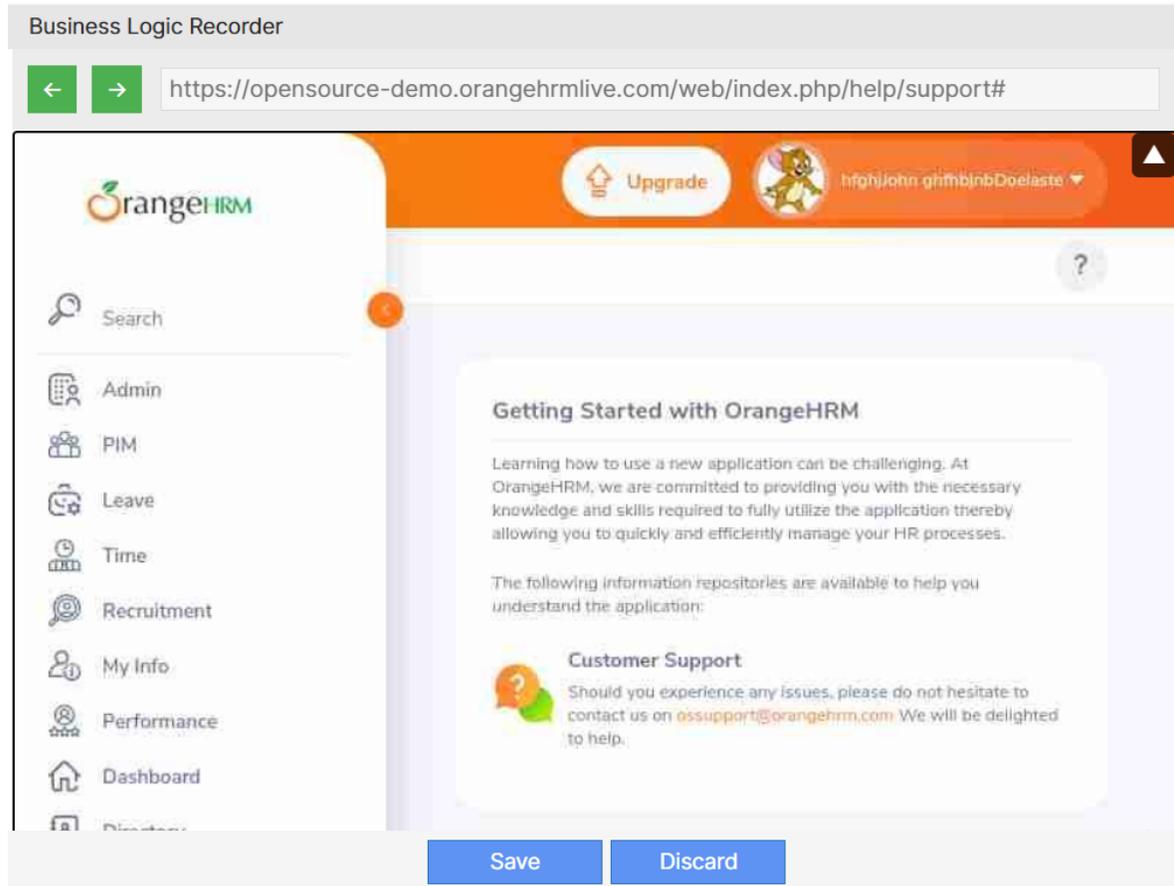
- The Business Logic Recorder does not support DevSecOps or proxy-based scans.
- For targets requiring multi-factor authentication, scheduled scans cannot capture the required tokens or responses. Use a standard scan to ensure proper scanning of authenticated URLs.

- [Creating a new recording](#)
- [Downloading Discovered Network Endpoints](#)
- [Deleting a recording](#)

### Creating a new recording

You can create up to 10 recordings. To record a business logic sequence:

1. Navigate to **Scans Policy > Scan Configuration** page. See [Configuring the Scanner](#).
2. In the **Replay with Automation** section, select **Business Logic Recorder** tab.
3. Configure a session timeout from one to 24 hours.
4. Click **New Recording**. The target URL launches in **Business Logic Recorder** window.
5. Perform the necessary navigation and actions within the web application and click **Save**. You cannot use the tab key within the embedded browser. If the back button is clicked multiple times, a blank page may display. Click **Discard** and start a new recording.



6. Provide a name for the recording (maximum 64 characters) and click **Save**. The discovered URIs, cookies, and APIs from the recorded business logic sequence are included in subsequent scans, enhancing scan coverage.  
Alternatively, click **Discard** at any time during the recording to cancel.
7. Click **OK**.



The **Session Status** shows if the session is valid. If expired, you are prompted to log in again when you perform scan.

You can modify the session timeout at any time, and the latest timeout value updates all recordings based on the cookies or session details of the most recent recording.

## Downloading Discovered Network Endpoints

Select a recording and click **Download Discovered Network Endpoints** to download a *swagger-specification.json* file containing detected endpoints.

## Deleting a recording

Select a recording you want to delete and click **Delete**.

The screenshot shows the 'Business Logic Recorder' interface. On the left, there is a sidebar with 'Login & Replay' and 'Automation Scripts' tabs. The main area has a 'Set Session timeout: 1 Hour' dropdown. Below this are buttons for 'New Recording', 'FAQs', 'Delete', and 'Discovered Network Endpoints'. A table lists recordings with columns for Name, Created at (UTC), Session Status, and Total API & Page Visits.

<input type="checkbox"/>	Name ↕	Created at (UTC) ↕	Session Status ↕	Total API & Page Visits ↕
<input checked="" type="checkbox"/>	Login_Workflow	2025/03/19 17:15:46	<input checked="" type="checkbox"/>	64

## Login and Replay

The FortiDAST Login and Replay feature is a powerful tool that enables you to capture complex login sequences, including multi-form and multi-factor authentication (One-Time Passwords or Tokens) for vulnerability scans.

Modern web applications, such as e-commerce sites, cloud providers, and bank websites, often involve complex login sequences.

- Multi-form/multi-page authentication: The user must first provide their email address or phone number, and then enter their credentials on the next page.
- Multi-factor authentication: The user must log in with their username and password, and then enter a one-time password (OTP) or token sent to their phone.
- SSO (single sign-on): SSO allows users to log in to multiple applications using a single set of credentials.

The FortiDAST Login and Replay feature can be used to capture these login sequence and replay it during the vulnerability scan, ensuring that the scan is performed as if a real user is logged in. This can help to identify vulnerabilities that would not be found by a traditional vulnerability scan.

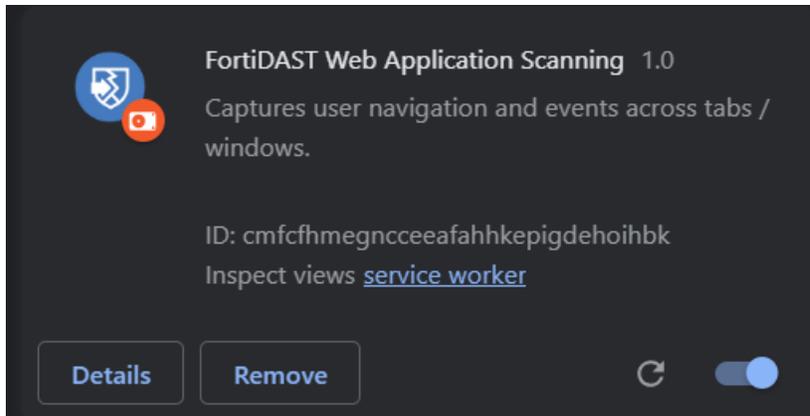
The login sequences are captured via *FortiDAST Web Application Scanning* Chrome extension and saved as JSON file. You must upload the saved JSON file in the *Scan Configuration > Login & Replay* tab. FortiDAST will replay the recording to simulate the login process and automatically download the session cookies to continue authenticated scanning of the web pages.

**Note:** The Login and Replay feature is not supported in Proxy mode. Therefore, FortiDevsec users who use FortiDAST for scanning will not be able to use this feature, as their scans are performed through Proxy.

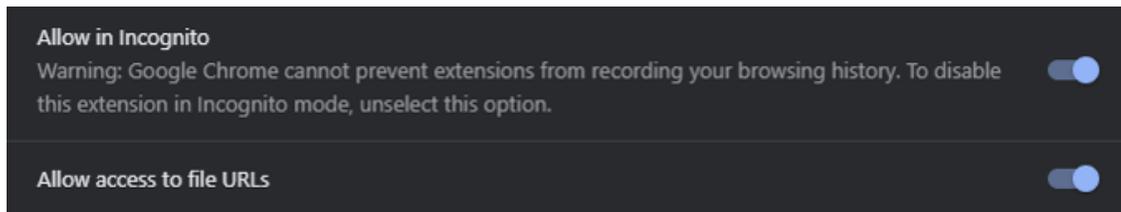
Following is an overview of login and replay process:

1. Download the *FortiDAST Web Application Scanning* Chrome extension. See [Downloading FortiDAST Web Application Scanning extension](#).
2. Capture the web application login sequence. See [Capturing login sequence](#).





8. Provide necessary permissions to the extension.
  - a. Click **Details**.
  - b. In the FortiDAST Web Application Scanning page, scroll down and toggle **Allow in incognito** option.
  - c. Toggle **Allow access to file URLs** option.

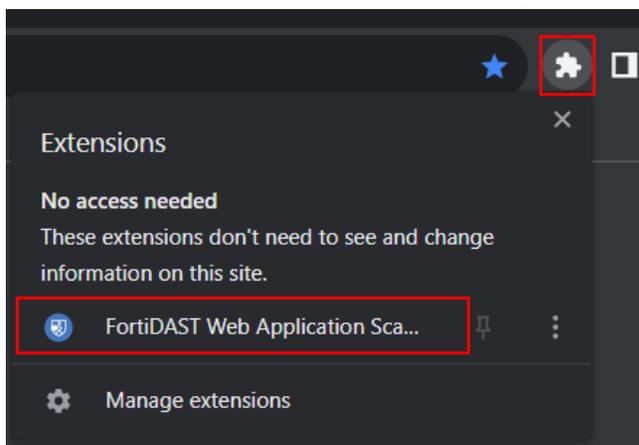


**Note:** Since FortiDAST Web Application Scanning extension operates only in incognito mode and requires access to file URLs, this step is mandatory and is important for the proper functioning of the extension.

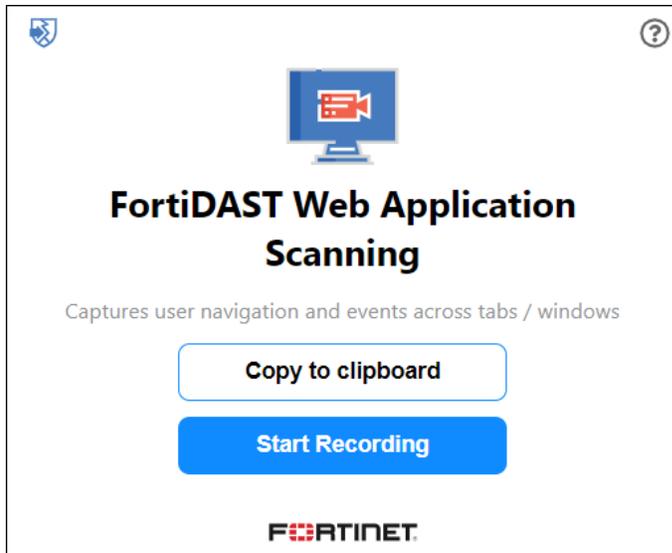
### Capturing login sequence

To capture the web application login sequence, perform the following steps:

1. Launch Google Chrome browser.
2. Click **Extensions** icon on the Chrome toolbar.
3. Select **FortiDAST Web Application Scanning**.



4. Click **Start Recording** to start a new recording. A new incognito tab opens.



**Note:** **Copy to clipboard** option is used to download the previously recorded sequence.

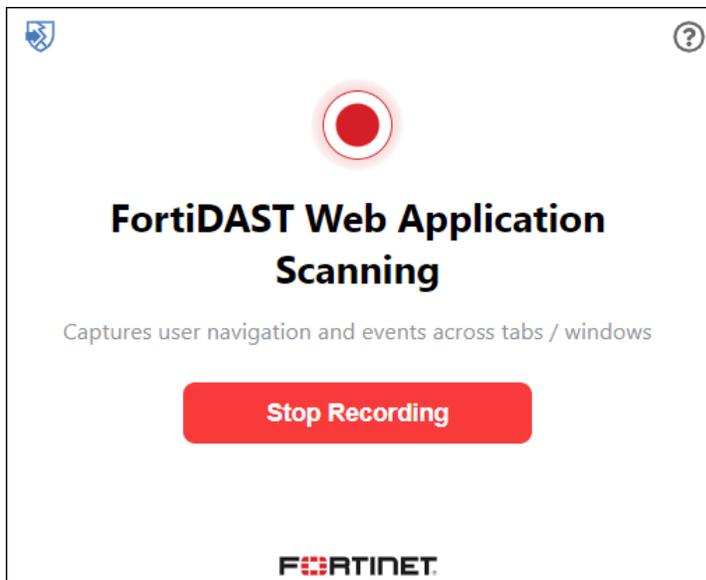
5. Navigate to the desired web application and perform the login process.

**Notes:**

- You must manually enter credentials or any input. Copy pasting in the input field will not capture the required data.
- Do not switch to a different tab when recording is in progress. It is recommended to complete the login sequence before switching to a different tab.

6. Once you complete all the activities, click **Extensions** icon on the Chrome toolbar, select **FortiDAST Web Application Scanning**, and click **Stop Recording**. The incognito tab closes and the recording(.json file) is downloaded to the local machine automatically.

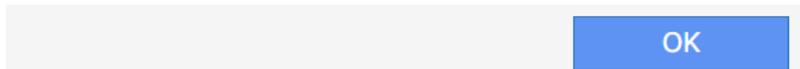
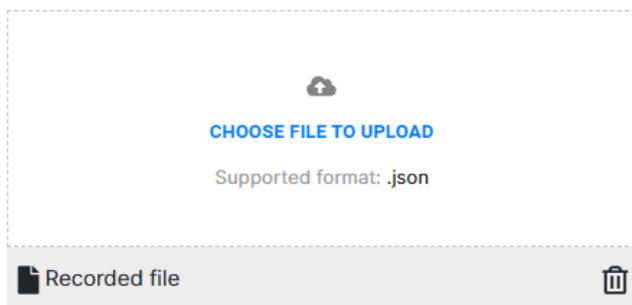
**Note:** Stop recording on the logged-in page where the login cookies or storage need to be collected. Before stopping the recording, wait for the web page to load completely after logging in. If you stop the recording before the page loads completely, the replay may fail.



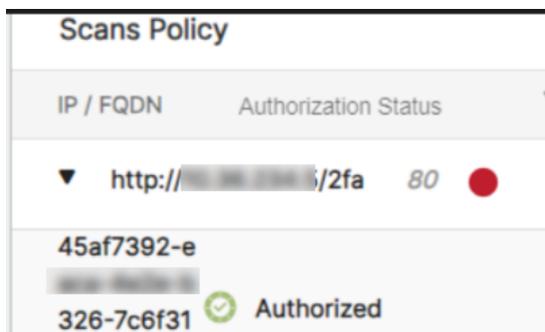
7. Click **Downloads** icon in the Chrome toolbar to access the downloaded file. Click on the **Open in folder** icon to open the download folder.

### Uploading recording to FortiDAST

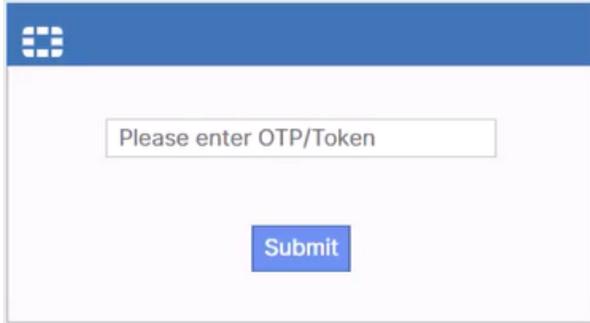
1. Login to FortiDAST portal.
2. Navigate to **Scans Policy > Scan Configuration** page. See [Configuring the Scanner](#).
3. In the **Replay with Automation** section, select **Login & Replay** tab.
4. Click **Choose File to Upload**.
5. Browse and select the previously downloaded recording (.json file).
6. Click **OK**.



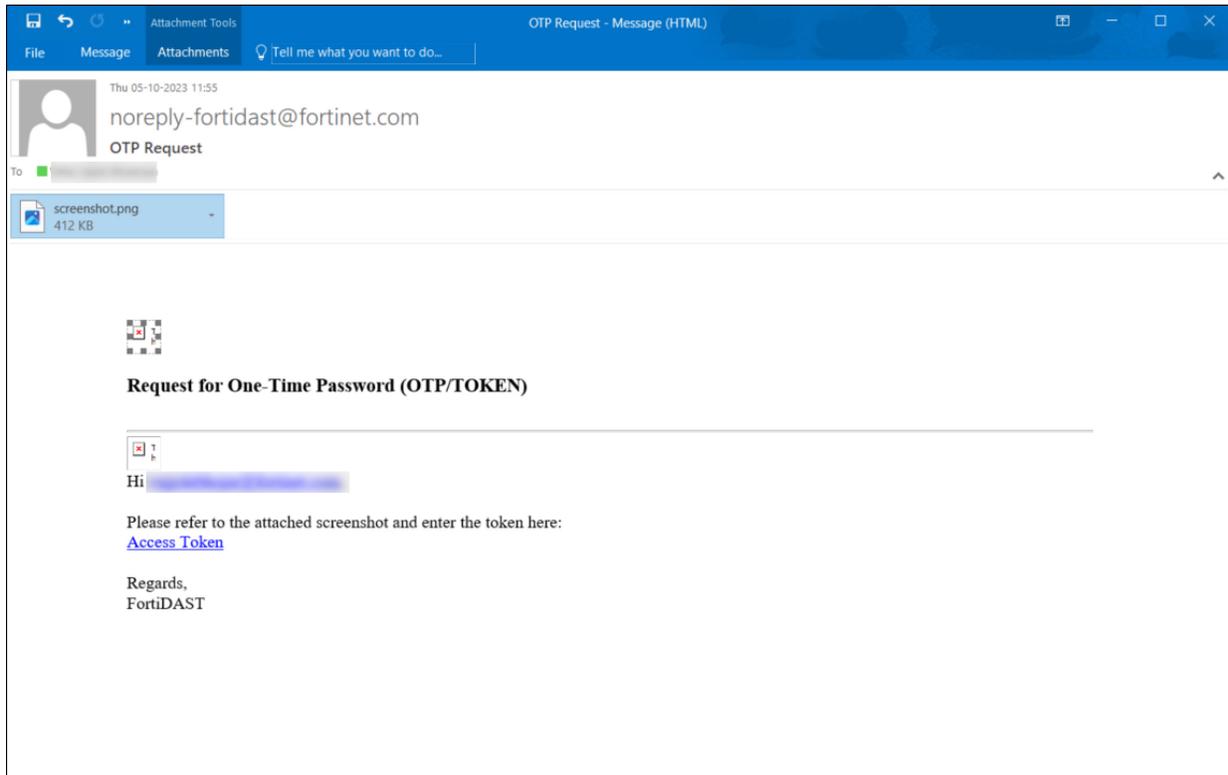
Once the scan is initiated, a red icon is displayed next to the scan IP in the **Scan Policy** page. This indicates that the scan is awaiting user input to proceed.



Click the red icon, a pop up is displayed with input field to provide OTP/Token. Enter the OTP/Token and click **Submit**. If the correct data is entered, scan continues.



Also, an email will be sent registered email address with a link to provide OTP/Token.



## Automation Scripts

You can upload your existing automation scripts to FortiDAST. Once uploaded, these scripts are replayed in a controlled environment, providing several benefits:

- Improve data collection for authentication, crawling, and fuzzing of web applications.
- Discover hard-to-find URLs through efficient web crawling.
- Replay core business logic to identify vulnerabilities.

<input type="button" value="+ Create"/> <input type="button" value="Delete"/> <input type="button" value="Show Details"/> <input type="button" value="Select for session logging"/> <input type="button" value="FAQs"/>				
<input type="checkbox"/>	Script Name	Validation Status	Session Logging	Validation Errors
<input type="checkbox"/>	Sample Script	In Progress	Disabled	

1

- [Adding an automation script](#)
- [Script validation](#)
- [Session logging](#)
- [Viewing script details](#)
- [Deleting a script](#)

## Adding an automation script



Before uploading to FortiDAST ensure the script requirements are met. Also, the automation script must be configured with target *URL*, *UUID*, and *FortiDAST API key*. See [Automation Script Prerequisites](#)

You can also click **FAQs** in the GUI to learn more about script configuration.

1. Navigate to **Configure > Replay with Automation**.
2. Select the **Automation Scripts** tab.
3. Click **+ Create**.
4. Enter a unique name for the script. This must match with **script\_name** parameter configured in the script.
5. Browse and upload the script file.

Add Script

Script Details

Script Name

SampleScript

Upload Files



SampleScript.py

5.44 KiB

Run script on every scan

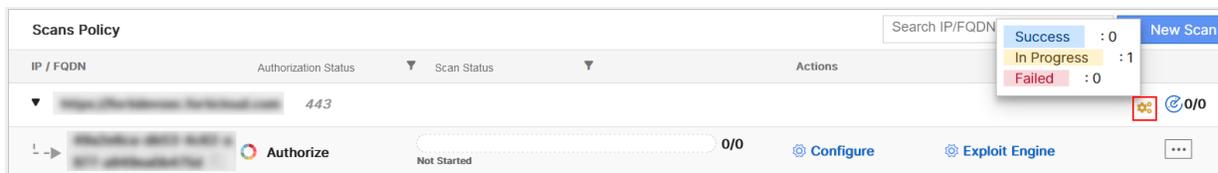
6. Enable **Run script on every scan** toggle to execute the uploaded script with each scan.
7. Click **OK** to save the script.

### Script validation

The **Validation Status** indicates the script's current validation state.

- **New:** Script uploaded but not yet validated.
- **In Progress:** Script validation is ongoing.
- **Successful:** Script is validated and ready for use.
- **Failed:** Script validation failed. Check **Validation Errors** for details.

The status of scripts validation can also be viewed on the Scans Policy page. A gear icon is displayed when scripts are added. Hover on the icon to view the count of scripts *In Progress*, *Success* or *Failed*.



## Session logging

Session logging captures cookies and other session details during script execution. To enable session logging, select a script and click **Select for session logging**. Confirm your selection in the popup window.

## Viewing script details

Select a script and click **Show Details**. This displays the script details and validation results. You can also download the script execution output, containing traversed URLs and APIs.

## Deleting a script

Select a script you want to delete and click **Delete**.



During rescan, the output from initial script validation and execution is used.

## Automation Script Prerequisites

To ensure successful integration with FortiDAST, review the script requirements and configuration steps.

### Requirements

Following are the script requirements.

- Script size limit: **200 MiB**
- Supported formats: **.py**
- Scripting languages: **Python** and **Selenium IDE**
- Supported browser driver: **Chrome**

### Configuration

Perform the following steps to configure your automation script before uploading it to FortiDAST.

#### 1. Adding logging preferences

Include the following Selenium logging preferences to capture visited URLs and network APIs during script execution.

```
from selenium import webdriver

chrome_options = webdriver.ChromeOptions()
```

```
chrome_options.set_capability("goog:loggingPrefs", {"performance": "ALL", "browser": "ALL"})
driver = webdriver.Chrome(options=chrome_options) # Assuming chromedriver is in PATH
```

## 2. Exporting automation output

Integrate the `export_output` function to send the script's execution data to FortiDAST for analysis.

```
# Call the function before exiting the webdriver

def export_output(self, method):
    requestBody = {}
    requestBody['url'] = "<Target URL>" # Replace with your target URL
    requestBody['uuid'] = "<UUID>" # Replace with your FortiDAST assigned UUID
    requestBody['script_name'] = "<Script name>" # Replace with your script's unique
name
    requestBody['json_content'] = self.driver.get_log('performance')
    jsonData = json.dumps(requestBody)
    # Print for debugging purposes (optional)
    # print(jsonData)
    headers = {"X-API-Key": "{0}".format("<FortiDAST Privileged API Key>"), "Content-
Type": "application/json; charset=utf-8"} # Replace with FortiDAST privileged API key
    resp = requests.post("https://fortidast.forticloud.com/api/v1.0/asset/business_
trace", headers=headers, data=jsonData, verify=False)
    self.driver.quit()
```

Note that information like *target URL*, *UUID*, and *FortiDAST API key* must be replaced.

- A UUID is generated when a new asset is added, see [Asset Authorization](#). You can also copy the UUID from the [configuration](#) page.
- See [REST API](#) to generate a privileged API key.

## Exploit Engine

The FortiDAST Scripting Engine (FSE) is a proprietary exploit engine that allows you to detect specific CVE vulnerabilities using built-in signatures covering **ZeroShell, WordPress, Joomla, SAP, Java Primefaces, Apache Struts, Phpunit, Thinkphp, Sharepoint, MExchange, Apache HTTP Server, Nginx, Allegro, SMB, VMware, GitLab, Zoho, Spring-framework, Atlassian, GLPI, CentOS, Cacti, Microsoft, OpenSSL, Apache Log4J, dotCMS, IIS, DVR, Telerik, SolarView, NetScaler, ColdFusion, JetBrains, Palo Alto, C-DATA, Check Point, ConnectWise, D-Link, PHP, Rejetto, ServiceNow, SolarWinds, GeoServer, CrushFTP, Apache Kafka, Apache OFBiz, Liner eMerge, Dahua, Jenkins, Craft CMS, SuiteCRM, Progress - Kemp, Mitel, and Redis**. For more information on exploit engine configuration, see [Configuring Exploit Engine on page 63](#).

The following table lists the vulnerabilities supported by FSE. For more information on the vulnerabilities listed in this table, see [CVE Details](#).

CVE	Description
SAP	

CVE	Description
CVE-2015-8840	The XML Data Archiving Service (XML DAS) in SAP NetWeaver AS Java.
CVE-2016-3973	The chat feature in the Real-Time Collaboration (RTC) services 7.3 and 7.4 in SAP NetWeaver Java AS 7.1 through 7.5.
CVE-2016-3975	Cross-site scripting (XSS) vulnerability in SAP NetWeaver AS Java 7.1 through 7.5.
CVE-2018-2366	SAP Business Process Automation (BPA) By Redwood, 9.0, 9.1.
CVE-2020-6287	SAP NetWeaver AS JAVA (LM Configuration Wizard), versions - 7.30, 7.31, 7.40, 7.50.
CVE-2022-22536	A memory pipes (MPI) de-synchronization vulnerability.
<b>WordPress</b>	
CVE-2018-7422	A Local File Inclusion vulnerability in the Site Editor plugin through 1.1.1 for WordPress.
CVE-2019-9978	The social-warfare plugin before 3.5.3 for WordPress.
CVE-2014-9119	Directory traversal vulnerability in download.php in the DB Backup plugin 4.5 and earlier for WordPress allows remote attackers to read arbitrary files via a .. (dot dot) in the file parameter.
CVE-2015-1579	Directory traversal vulnerability in the Elegant Themes Divi theme for WordPress allows remote attackers to read arbitrary files via a .. (dot dot) in the img parameter in a revslider_show_image action to wp-admin/admin-ajax.php.
CVE-2015-6522	SQL injection vulnerability in the WP Symposium plugin before 15.8 for WordPress allows remote attackers to execute arbitrary SQL commands via the size parameter to get_album_item.php.
CVE-2020-10257	The ThemeREX Addons plugin before 2020-03-09 for WordPress lacks access control on the /trx_addons/v2/get/sc_layout REST API endpoint.
CVE-2020-10564	A directory traversal in the File Upload plugin before 4.13.0 for WordPress can lead to remote code execution by uploading a crafted txt file into the lib directory, because of a wfu_include_lib call.
CVE-2023-28121	An authentication bypass vulnerability affecting the WooCommerce Payments plugin version 4.8.0 through 5.6.1. Successful exploitation of the vulnerability could allow an unauthorized attacker to gain admin privileges on the WordPress websites installed with the vulnerable version of the plugin enabled.
CVE-2023-6961	The WP Meta SEO plugin for WordPress is vulnerable to Stored Cross-Site Scripting via the 'Referer' header in all versions up to, and including, 4.5.12 due to insufficient input sanitization and output escaping.

CVE	Description
CVE-2024-2194	A Cross-Site Scripting Vulnerability in WordPress Project WP Statistics. The vulnerability is due to improper validation of user input. A remote, unauthenticated attacker could exploit the vulnerability by sending a crafted request to the target server. Successfully exploiting this vulnerability could result in arbitrary script execution.
CVE-2024-3552	A SQL injection vulnerability was found in the Web Directory plugin for WordPress. Successful exploitation could allow attackers to execute arbitrary SQL commands on the vulnerable database.
CVE-2024-3922	This vulnerability affects the Doken plugin for WordPress. It allows attackers to inject malicious SQL queries into the database through crafted input, potentially leading to data exposure, modification, or deletion.
CVE-2024-5522	This vulnerability affects the WP HTML5 Video Player plugin for WordPress. It allows attackers to inject malicious SQL queries into the database through crafted input, potentially leading to data exposure, modification, or deletion.
CVE-2024-3495	A critical SQL injection vulnerability affecting the <i>Country State City Dropdown CF7</i> WordPress plugin (versions $\leq 2.7.2$ ). Potential consequences include unauthorized access to sensitive database information like user credentials, personal data, and financial records.
CVE-2024-4443	The vulnerability is due to insufficient sanitization of user supplied inputs in the application. A remote attacker could exploit this vulnerability by sending a crafted request to the target server. Successful exploitation could lead to arbitrary SQL code execution in the security context of the database service connected to the application.
CVE-2024-6028	The vulnerability is due to insufficient sanitization of user supplied inputs in the application. A remote attacker could exploit this vulnerability by sending a crafted request to the target server. Successful exploitation could lead to arbitrary SQL code execution in the security context of the database service connected to the application.
<b>MS-Exchange</b>	
CVE-2021-26855	A Server-Side Request Forgery (SSRF) vulnerability.
CVE-2021-33766	An Information Disclosure vulnerability (ProxyToken).
CVE-2021-34473	A Remote Code Execution vulnerability (ProxyShell).
CVE-2021-42321	A high severity Remote Code Execution vulnerability that occurs due to improper validation of cmdlet arguments.
CVE-2022-41082	MS Exchange Proxynotshell Remote Code Execution vulnerability.
<b>Sharepoint</b>	
CVE-2019-0604	A remote code execution vulnerability exists in Microsoft SharePoint when the software fails to check the source markup of an application package.

CVE	Description
CVE-2020-1147	A remote code execution vulnerability exists in .NET Framework, Microsoft SharePoint, and Visual Studio when the software fails to check the source markup of XML file input.
CVE-2020-16952	A remote code execution vulnerability exists in Microsoft SharePoint when the software fails to check the source markup of an application package, aka 'Microsoft SharePoint Remote Code Execution Vulnerability'.
CVE-2021-31181	The <code>EditingPageParser.VerifyControlOnSafeList</code> method fails to properly validate user-supplied data. This can be leveraged by an attacker to leak sensitive information in rendered-preview content.
CVE-2020-0646	A remote code execution vulnerability exists when the Microsoft .NET Framework (versions 3.5 and 4.x) SharePoint servers using vulnerable .NET frameworks are affected.
CVE-2021-31950	A Server Spoofing (SSRF) vulnerability.
<b>Joomla!</b>	
CVE-2015-8562	Joomla! 1.5.x, 2.x, and 3.x before 3.4.6 allow remote attackers to conduct PHP object injection attacks and execute arbitrary PHP code via the HTTP User-Agent header.
CVE-2023-23752	An issue was discovered in Joomla! 4.0.0 through 4.2.7. An improper access check allows unauthorized access to webservice endpoints.
<b>Apache</b>	
CVE-2006-3747	Off-by-one error in the LDAP scheme handling in the <i>Rewrite</i> module in Apache 1.3 from 1.3.28, 2.0.46 and other versions before 2.0.59, and 2.2. When <i>RewriteEngine</i> is enabled, allows remote attackers to cause a denial of service (application crash) and possibly execute arbitrary code via crafted URLs that are not properly handled using certain rewrite rules.
CVE-2017-5638	The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
CVE-2021-41773	A path traversal vulnerability in Apache HTTP Server 2.4.49. An attacker could use a path traversal attack to map URLs to files outside the directories configured by Alias-like directives.
CVE-2021-42013	It was found that the fix for CVE-2021-41773 in Apache HTTP Server 2.4.50 was insufficient. An attacker could use a path traversal attack to map URLs to files outside the directories configured by Alias-like directives.

CVE	Description
CVE-2021-44228	Log4j versions prior to 2.15.0 are subject to a remote code execution vulnerability via the LDAP JNDI parser. The affected products are, Apache Struts (2.5.8), Elastic Search (5.0.0-5.6.10, 6.0.0-6.3.2), Apache Solr (7.4.0-7.7.3, 8.0.0-8.11.0), Apache JSPwiki (2.11.0), Apache Druid (0.22), and Apache OFBIZ(18.12.03).
CVE-2021-45046	The fix to address CVE-2021-44228 Log4Shell in Apache Log4j 2.15.0 was incomplete in certain non-default configurations. This could allow attackers to craft malicious input data using a JNDI Lookup pattern resulting in an information leak and remote code execution in some environments and local code execution in all environments.
<b>Zeroshell</b>	
CVE-2009-0545 (Zeroshell2.0rc2)	cgi-bin/kerbynet in ZeroShell 1.0beta11 and earlier allows remote attackers to execute arbitrary commands via shell metacharacters in the type parameter in a NoAuthREQ x509List action.
CVE-2019-12725 (zeroshell3.9.0)	Zeroshell 3.9.0 is prone to a remote command execution vulnerability. An unauthenticated attacker can exploit this issue by injecting OS commands inside the vulnerable parameters.
CVE-2020-29390 (zeroshell3.9.3)	Zeroshell 3.9.3 allows an unauthenticated attacker to execute a system command by using shell metacharacters and the %0a character.
<b>PHPUnit</b>	
CVE-2017-9841	Util/PHP/eval-stdin.php in PHPUnit before 4.8.28 and 5.x before 5.6.3 allows remote attackers to execute arbitrary PHP code via HTTP POST data beginning with a "<?php " substring, as demonstrated by an attack on a site with an exposed /vendor folder.
<b>ThinkPHP</b>	
CVE-2018-20062	NoneCms V1.3. thinkphp/library/think/App.php allows remote attackers to execute arbitrary PHP code
<b>SMB</b>	
CVE-2020-0796	A remote code execution vulnerability exists in the way that the Microsoft Server Message Block 3.1.1 (SMBv3) protocol handles certain requests, aka "Windows SMBv3 Client/Server Remote Code Execution Vulnerability."
<b>Java PrimeFaces</b>	
CVE-2017-1000486	A Remote Code Execution vulnerability.
<b>Nginx</b>	
CVE-2009-2629	Buffer underflow in <i>src/http/nginx_http_parse.c</i> in nginx 0.1.0 through 0.5.37, 0.6.x before 0.6.39, 0.7.x before 0.7.62, and 0.8.x before 0.8.15 allows remote attackers to execute arbitrary code via crafted HTTP requests.

CVE	Description
CVE-2014-0133	Heap-based buffer overflow in the SPDY implementation in nginx 1.3.15 before 1.4.7 and 1.5.x before 1.5.12 allows remote attackers to execute arbitrary code via a crafted request.
<b>OpenSSL</b>	
CVE-2014-0160	The TLS and DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to <code>d1_both.c</code> and <code>t1_lib.c</code> , that is, the <i>Heartbleed</i> bug.
<b>Allegro</b>	
CVE-2014-9222	Allows remote attackers to gain privileges via a crafted cookie that triggers memory corruption, aka the <i>Misfortune Cookie</i> vulnerability.
<b>IIS</b>	
CVE-2017-7269	Buffer overflow in the <code>ScStoragePathFromUrl</code> function in the WebDAV service in IIS 6.0 in Microsoft Windows Server 2003 R2 allows remote attackers to execute arbitrary code via a long header beginning with <code>"If: &lt;http://"</code> in a <code>PROPFIND</code> request.
<b>dotCMS</b>	
CVE-2022-35740	A XSS filter mechanism bypass was found in dotCMS version 22.05 and below using Matrix Parameters. The XSS filter is an input sanitizer designed by the vendor to minimize CORS attack, XSS and CSRF vulnerabilities in the administrator portal, by abusing this an attacker can cause critical compromise.
CVE-2022-37033	A Server-Side Request Forgery bypass was found in dotCMS version 22.05 and below due to the incomplete validate private address. By using redirection technique, an attacker can request to server internal resources.
CVE-2022-37034	A Denial-of-Service was found in dotCMS version 22.05 and below. The issue is located in <code>TempFileAPI</code> when it tries to access and download the contents of remote URL. Directing it to access a heavy file using multiple requests at once results in memory exhaustion or DoS.
CVE-2022-37431	Multiple endpoints were found to be vulnerable to XSS in the dotCMS admin portal. This occurs when the configuration has <code>XSS_PROTECTION_ENABLED=false</code> .
<b>Redis</b>	
CVE-2022-0543	Redis (Debian version lower than 5:5.0.14-1+deb10u2 (buster) and Debian version lower than 5:6.0.16-1+deb11u2 (bullseye)), a persistent key-value database, due to a packaging issue, is prone to a (Debian-specific) Lua sandbox escape, which could result in remote code execution.
<b>VMware</b>	

CVE	Description
CVE-2021-21974	VMware ESXi servers vulnerable to the OpenSLP heap-overflow vulnerability and are being exploited through the OpenSLP, port 427 to deliver a new ransomware “ESXiArgs”. The ransomware encrypts files in affected ESXi servers and demand a ransom for file decryption. Also, this vulnerability can result in remote code execution, allowing the attacker to get full control of the target.
CVE-2021-22005	The vCenter Server contains an arbitrary file upload vulnerability in the Analytics service. A malicious actor with network access to port 443 on vCenter Server may exploit this issue to execute code on vCenter Server by uploading a specially crafted file.
CVE-2023-20887	Aria Operations for Networks contains a command injection vulnerability. A malicious actor with network access to VMware Aria Operations for Networks may be able to perform a command injection attack resulting in remote code execution.
<b>Cacti</b>	
CVE-2022-46169	In affected versions of Cacti v1.2.22, a command injection vulnerability allows an unauthenticated user to execute arbitrary code on a server running Cacti. Gaining access to the Cacti instance of an organization could give attackers with the opportunity to learn about the types of devices on the network and their local IP addresses.
<b>Atlassian Confluence</b>	
CVE-2022-26134	A critical 0-day vulnerability on Atlassian Confluence Data Center and Server is actively being exploited in the wild. The vulnerability is established via the Object Graph Navigation Language (OGNL) injection that allows an unauthenticated user to execute arbitrary code.
CVE-2021-26084	An attack attempt to exploit a Remote Code Execution Vulnerability in Atlassian Confluence. The vulnerability is due to insufficient input validation while handling a malicious request. A remote attacker may be able to exploit this to execute arbitrary code via a crafted HTTP request.
CVE-2022-26138	This indicates an attack attempt to exploit an Information Disclosure vulnerability in Atlassian Questions For Confluence app. The vulnerability is due to insufficient validation of user supplied input. A remote attacker can exploit this to gain unauthorized access to sensitive information.
<b>CentOS</b>	

CVE	Description
CVE-2022-44877	A command injection vulnerability that allows remote attackers to easily exploit CWP (Control Web Panel) with a crafted HTTP request which can result in Remote Code Execution. This vulnerability can be leveraged to perform ransomware attacks or exfiltration of data.
<b>Zoho</b>	
CVE-2021-40539	APT Actors are actively exploiting Zoho ManageEngine ServiceDesk Plus which is an IT help desk software with asset management. The exploit is rated critical due to its capability for unauthenticated remote code execution (RCE).
<b>GitLab</b>	
CVE-2021-22205	An issue has been discovered in GitLab CE/EE affecting all versions starting from 11.9. GitLab was not properly validating image files that were passed to a file parser which resulted in a remote command execution.
<b>Spring-framework</b>	
CVE-2022-22963	In Spring Cloud Function versions 3.1.6, 3.2.2 and older unsupported versions, when using routing functionality it is possible for a user to provide a specially crafted SpEL as a routing-expression that may result in remote code execution and access to local resources.
CVE-2022-22965	A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.
CVE-2022-22980	A Spring Data MongoDB application is vulnerable to SpEL Injection when using @Query or @Aggregation-annotated query methods with SpEL expressions that contain query parameter placeholders for value binding if the input is not sanitized.
<b>GLPI-Project</b>	
CVE-2022-35914	/vendor/htmlawed/htmlawed/htmlawedTest.php in the htmlawed module for GLPI through 10.0.2 allows PHP code injection.
<b>Microsoft</b>	
CVE-2023-21554	Microsoft Message Queuing Remote Code Execution Vulnerability
CVE-2023-32057	It is an out-of-bounds write vulnerability in the Message Queuing service of Microsoft Windows. The vulnerability could potentially lead to unauthenticated remote code execution in the Message Queuing service due to the lack of bound checks when reading user-controlled section sizes.
<b>Realtek</b>	

CVE	Description
CVE-2021-35394	Realtek Jungle SDK Vulnerability is an arbitrary command injection vulnerability in Realtek Jungle SDK. Successful exploitation of this vulnerability allows a remote attacker to execute arbitrary code on vulnerable devices, leading to system compromise. Realtek Jungle SDK based IoT devices are available from multiple vendors.
<b>Tplink</b>	
CVE-2023-1389	TP-Link Archer AX-21 Command Injection Attack. TP-Link Archer AX21 (AX1800) firmware versions before 1.1.4 contains a command injection vulnerability in the web management interface specifically in the <i>Country</i> field. There is no sanitization of this field, so an attacker can exploit it for malicious activities and gain foothold. The vulnerability has been seen to be exploited in the wild to deploy Mirai botnet.
<b>RocketMQ</b>	
CVE-2023-33246	A command injection vulnerability that affects Apache RocketMQ versions 5.1 and lower. Successful exploitation of the vulnerability allows a remote attacker to execute commands as the system user under which RocketMQ is running by using the update configuration function.
<b>PaperCut</b>	
CVE-2023-27350	PaperCut MF/NG Improper Access Control Vulnerability. An unauthenticated attacker can perform a Remote Code Execution (RCE) on a vulnerable PaperCut Application Server. According to the vendor, the specific flaw exists within the SetupCompleted class and could be achieved remotely without authentication. PaperCut MF/NG Improper Access Control Vulnerability has been seen exploited in the wild.
<b>Ivanti</b>	
CVE-2023-35078	Ivanti Endpoint Manager Mobile (EPMM, formerly MobileIron Core) contains an authentication bypass vulnerability (CVE-2023-35078) that allows unauthenticated access to specific API paths. An attacker with access to these API paths can access personally identifiable information (PII) such as names, phone numbers, and other mobile device details for users on a vulnerable system. An attacker can also make other configuration changes including installing software and modifying security profiles on registered devices.
CVE-2024-21893	A server-side request forgery vulnerability in the SAML component of Ivanti Connect Secure.
CVE -2024-22024	A XML external entity injection (XXE) vulnerability in the SAML component of Ivanti Connect Secure, Ivanti Policy Secure and ZTA gateway.
CVE-2024-9379	SQL injection in the admin web console of Ivanti CSA before version 5.0.2 allows a remote authenticated attacker with admin privileges to run arbitrary SQL statements.

CVE	Description
CVE-2024-8963	The vulnerability is due to insufficient validation of user-supplied inputs. A remote unauthenticated attacker can exploit this vulnerability by sending maliciously crafted requests to a vulnerable application. Successful exploitation could result in information disclosure and access to restricted functionality of the application.
<b>dvr</b>	
CVE-2018-9995	Authentication bypass vulnerability in various TBK DVR4104 and DVR4216 devices, allowing attackers to gain administrative access without proper credentials.
<b>ColdFusion</b>	
CVE-2023-26360	Critical improper access control vulnerability in Adobe ColdFusion, enabling potential remote code execution by unauthenticated attackers.
CVE-2024-20767	ColdFusion versions 2023.6, 2021.12 and earlier are affected by an Improper Access Control vulnerability that could lead to arbitrary file system read. An attacker could leverage this vulnerability to bypass security measures and gain unauthorized access to sensitive files and perform arbitrary file system write.
CVE-2023-29298	Adobe ColdFusion is affected by an Improper Access Control vulnerability that could result in a Security feature bypass.
CVE-2023-38205	
<b>NetScaler</b>	
CVE-2023-4966	Citrix NetScaler ADC and Gateway vulnerability allowing sensitive information disclosure, potentially including user session tokens.
<b>Telerik</b>	
CVE-2017-11317	Vulnerability in the Telerik UI for ASP.NET AJAX component that allows attackers to upload arbitrary files or execute code due to weak encryption in the RadAsyncUpload feature.
CVE-2024-4358	The vulnerability is due to improper validation of crafted HTTP requests. A remote, unauthenticated attacker could exploit this vulnerability by sending maliciously crafted packets to a target system. Successfully exploiting the vulnerability could allow an attacker to create an administrator user and execute code under the security context of the target server.
<b>SolarView</b>	
CVE-2022-29303	A command injection vulnerability in SolarView Compact ver. 6.00 (conf_mail.php) allows attackers to execute arbitrary code on the affected system.
CVE-2022-40881	SolarView Compact 6.00 was discovered to contain a command injection vulnerability via network_test.php
<b>JetBrains</b>	

CVE	Description
CVE-2023-42793	Critical authentication bypass in JetBrains TeamCity on-premises servers, potentially allowing unauthenticated remote code execution.
CVE-2024-27198	In JetBrains TeamCity before 2023.11.4 authentication bypass allowing to perform admin actions was possible.
<b>Palo Alto</b>	
CVE-2024-3400	A command injection as a result of arbitrary file creation vulnerability in the GlobalProtect feature of Palo Alto Networks PAN-OS software for specific PAN-OS versions and distinct feature configurations may enable an unauthenticated attacker to execute arbitrary code with root privileges on the firewall.
CVE-2024-9465	An attack attempt to exploit an SQL Injection vulnerability in Palo Alto Networks Expedition. The vulnerability is due to insufficient sanitization of user supplied inputs in the application. A remote, unauthenticated attacker could exploit it by sending a specially crafted request to the target server. Successful exploitation could lead to arbitrary SQL code execution in the security context of the database service connected to the application.
CVE-2024-5910	A missing authentication vulnerability in Palo Alto Networks Expedition to its known exploited vulnerability (KEV) list. Expedition is a migration tool aiding in configuration migration, tuning, and enrichment from one of the supported vendors to Palo Alto Networks. Successful exploitation can lead to an admin account takeover.
CVE-2024-9463	An Authenticated Command Injection vulnerability in Palo Alto Networks Expedition. This vulnerability is due to insufficient validation of user-supplied inputs. A remote authenticated attacker can exploit this vulnerability by sending maliciously crafted requests to a vulnerable application. Successful exploitation could result in arbitrary command execution in the security context of the application.
CVE-2024-9466	A cleartext storage vulnerability in Palo Alto Networks Expedition that allows an authenticated attacker to reveal sensitive information like firewall usernames, passwords, and API keys. 1 This vulnerability poses a significant risk to the security of affected systems.
CVE-2024-9474	An Elevation of Privilege Vulnerability in Palo Alto Networks PAN-OS. The vulnerability is due to an error in the vulnerable application when handling a maliciously crafted input. A remote attacker may be able to exploit this to perform actions with root privileges.
<b>C-DATA</b>	
CVE-2022-4257	A vulnerability was found in C-DATA Web Management System affecting some unknown processing of the file cgi-bin/jumpto.php of the component GET Parameter Handler. The manipulation of the argument hostname leads to argument injection.

CVE	Description
<b>Check Point</b>	
CVE-2024-24919	Potentially allowing an attacker to read certain information on Check Point Security Gateways once connected to the internet and enabled with remote Access VPN or Mobile Access Software Blades.
<b>ConnectWise</b>	
CVE-2024-1709	ConnectWise ScreenConnect 23.9.7 and prior are affected by an Authentication Bypass Using an Alternate Path or Channel vulnerability, which may allow an attacker direct access to confidential information or critical systems.
<b>D-Link</b>	
CVE-2024-3273	A vulnerability, which was classified as critical, was found in D-Link DNS-320L, DNS-325, DNS-327L and DNS-340L up to 20240403. Affected is an unknown function of the file /cgi-bin/nas_sharing.cgi of the component HTTP GET Request Handler. The manipulation of the argument system leads to command injection.
CVE-2021-40655	This vulnerability affects legacy D-Link products. All associated hardware revisions have reached their end-of-life (EOL) or end-of-service (EOS) life cycle and should be retired and replaced per vendor instructions.
CVE-2014-100005	Multiple cross-site request forgery (CSRF) vulnerabilities in D-Link DIR-600 router (rev. Bx) with firmware before 2.17b02 allow remote attackers to hijack the authentication of administrators.
CVE-2024-10914	The vulnerability is due to insufficient validation of user supplied inputs in the device. A remote attacker may be able to exploit this to execute arbitrary OS commands within the context of the device, via a crafted HTTP request.
<b>PHP</b>	
CVE-2024-4577	An argument injection vulnerability in PHP, specifically Windows-based PHP used in CGI mode, that can be exploited to achieve remote code execution (RCE).
<b>Rejetto</b>	
CVE-2024-23692	Rejetto HTTP File Server, up to and including version 2.3m, is vulnerable to a template injection vulnerability.
<b>ServiceNow</b>	
CVE-2024-5217	Incomplete Input Validation in GlideExpression Script. This vulnerability could enable an unauthenticated user to remotely execute code within the context of the Now Platform.
CVE-2024-4879	Jelly Template Injection Vulnerability in UI macros that could enable an unauthenticated user to remotely execute code within the context of the Now Platform.

CVE	Description
<b>SolarWinds</b>	
CVE-2024-28995	A Directory Traversal Vulnerability in SolarWinds Serv-U software is being actively exploited in the wild. Tracked as CVE-2024-28995, the vulnerability is due to improper validation of the user-supplied inputs.
CVE-2024-28987	An attack attempt to exploit an Authentication Bypass Vulnerability in SolarWinds Web Help Desk (WHD). The vulnerability is due to the existence of hardcoded credentials. A remote, unauthenticated attacker can exploit this vulnerability by sending a maliciously crafted request to the vulnerable server. Successful exploitation could result in an attacker gaining full access to the application's data and functionality.
<b>CrushFTP</b>	
CVE-2024-4040	A zero-day security vulnerability has been uncovered in an enterprise file-transfer software CrushFTP. The vulnerability allows unauthenticated remote attackers to read files from the file system outside of the VFS Sandbox, gain administrative access, and perform remote code execution on the server.
<b>GeoServer</b>	
CVE-2024-36401	The vulnerability is due to lack of input validation when handling requests. A remote, unauthenticated attacker can exploit this vulnerability by sending maliciously crafted requests to the vulnerable server. Successful exploitation could result in arbitrary code execution in the security context of the application.
CVE-2022-24816	Code injection in the jt-jiffle extension of GeoServer.
<b>Apache Kafka</b>	
CVE-2023-52251	KAFKA UI Arbitrary Code Injection. The vulnerability is due to insufficient sanitizing of user supplied inputs in the application. A remote attacker may be able to exploit this to execute arbitrary code within the context of the application.
CVE -2024-32030	KAFKA UI Remote Code Execution.
<b>Apache OFBiz</b>	
CVE-2024-36104	A Path Traversal vulnerability in Apache OFBiz that exposes endpoints to unauthenticated users, who could leverage it to achieve remote code execution via specially crafted requests.
CVE-2024-38856	An Incorrect Authorization vulnerability, meaning that an unauthenticated user can access restricted functionalities.
<b>Linear eMerge</b>	

CVE	Description
CVE-2019-7256	An access control system called Linear eMerge E3-Series is affected by an OS command injection flaw that could allow an attacker to cause remote code execution and full access to the system.
<b>Dahua</b>	
CVE-2021-33044	This indicates an attack attempt to exploit an Authentication Bypass Vulnerability in Dahua Products.
CVE-2021-33045	The vulnerability is due to a lack of proper validation of user-supplied data. A remote, unauthenticated attacker could exploit this vulnerability by sending a crafted request to the target device. Successful exploitation could gain control of the affected device.
<b>Jenkins</b>	
CVE-2024-23897	This vulnerability may enable remote code execution (RCE) potentially leading to unauthorized access and data compromise. Exploiting this vulnerability allows attackers to read any files on the Jenkins controller file system.
CVE-2024-43044	Jenkins 2.470 and earlier, LTS 2.452.3 and earlier allows agent processes to read arbitrary files from the Jenkins controller file system by using the `ClassLoaderProxy#fetchJar` method in the Remoting library.
<b>Craft CMS</b>	
CVE-2024-37843	This vulnerability affects Craft CMS. It allows attackers to inject malicious SQL queries into the database through crafted input, potentially leading to data exposure, modification, or deletion.
<b>SuiteCRM</b>	
CVE-2024-36412	This vulnerability affects SuiteCRM. It allows unauthorized attackers to inject malicious SQL queries into the database through crafted input, potentially leading to data exposure, modification, or deletion.
<b>Progress - Kemp</b>	
CVE-2024-1212	A Code Injection Vulnerability in Progress Kemp LoadMaster. The vulnerability is due to insufficient sanitizing of user-supplied input. An attacker can exploit this issue to inject arbitrary code, which will be executed in the target user's system.
<b>Mitel</b>	
CVE-2024-41713	The vulnerability is due to improper validation of user-supplied inputs. A remote, unauthenticated attacker could exploit this vulnerability by sending a crafted request to the target server. Successful exploitation could result in unauthorized access to sensitive files, administrative endpoints, and restricted functionality within the application.

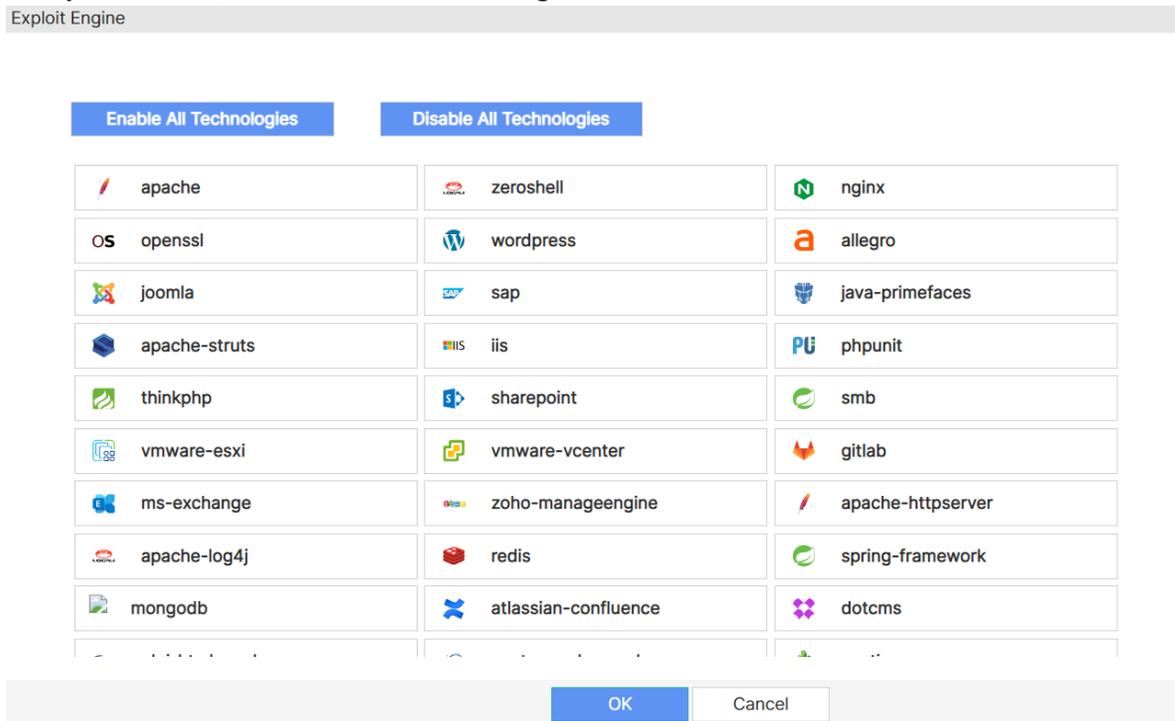
## Configuring Exploit Engine

To configure exploit engine, perform the following steps:

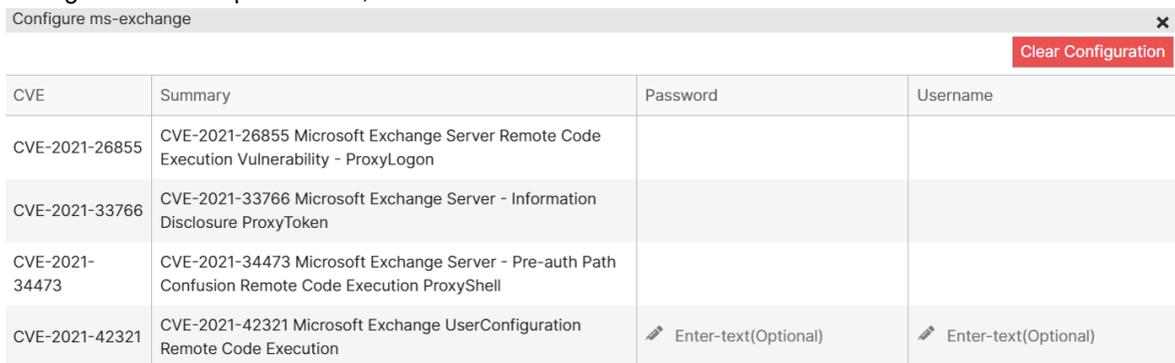
1. Navigate to **Scans Policy** page.
2. Click **Exploit Engine**.



3. Click the desired technology or to select all the available technologies click **Enable All Technologies**. To clear your selection, click **Disable All Technologies**.



4. Configure additional parameters, if available.



5. Click **Save**. A green tick is displayed next to the selected technology.
6. Click **OK**.

## Vulnerability Scanning

After successful authorization, the asset requires to be scanned for vulnerability assessment. To configure the vulnerability scanner, see [Configuring the Scanner on page 22](#).

Click **Scan**. A progress bar is displayed with the status of the scanning process. You can terminate the scanning process by clicking on **Stop**. After the current scanning process is complete, you can scan the asset again, click **Rescan**.



You can rescan URIs that fail the vulnerability assessment, see [Dashboard on page 87](#) and rescan vulnerable URIs see [Dashboard on page 87](#)

## Progress Summary

The progress summary pane displays the detailed information of the scan in progress.

Click view under the progress bar to open *Progress Summary* pane. The following information is displayed.

- **Recon** - Displays the count of *Distinct Ports Detected* and *Distinct Technologies Detected*.
- **Crawling** - Displays the *Total URIs detected*.
- **Scanning** - Displays the *Currently Scanning URIs* list along with *Time Estimated* in seconds to complete the scan.

The data in progress summary pane will be auto refreshed as the scan progresses.

**Progress Summary**

**Recon**

Distinct Ports Detected: **8** ✔ Network Enumeration Complete

Distinct Technologies Detected: **12** ⋮ Application Enumeration In Progress

**Crawling**

Total URIs detected: **1751** ✔ Crawling Complete

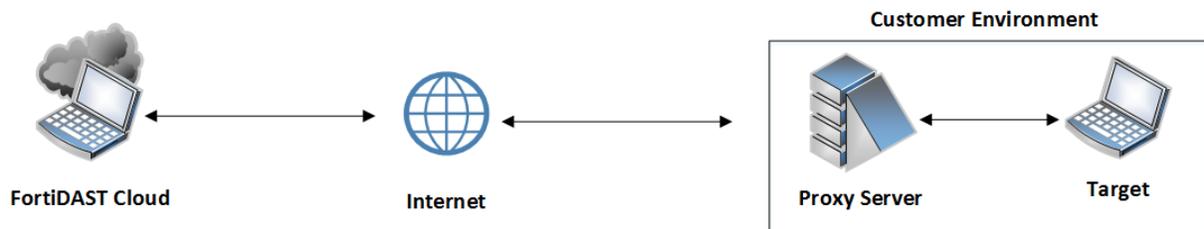
**Scanning**

**57/62** ⋮ Scan In Progress

Currently Scanning URIs	Time Estimated(Seconds)
<a href="#">http://[redacted]utillidae/index.php?page=documentation/usage-instructions.php</a>	775
<a href="#">http://[redacted]utillidae/index.php?page=view-someones-blog.php</a>	1335
<a href="#">http://[redacted]utillidae/hints-page-wrapper.php?level1HintIncludeFile=64</a>	1526
<a href="#">http://[redacted]utillidae/index.php?page=pen-test-tool-lookup-ajax.php</a>	1846
<a href="#">http://[redacted]utillidae/hints-page-wrapper.php?level1HintIncludeFile=57</a>	1848

## FortiDAST Proxy Server

You can use proxy servers with FortiDAST to scan internal web applications that are not exposed to the internet. This is particularly useful in scanning applications-in-development, internal corporate applications, integrating FortiDAST as a DAST scanner in DevOps pipelines, and so on. Deploy the proxy server in your environment as a Docker container, and it runs with minimum manual intervention.



The FortiDAST Proxy server supports two modes of deployment.

- **Autonomous** - The proxy server (Docker container) creates a reverse secure tunnel to the FortiDAST cloud and waits for commands. This is useful if you want to scan the application manually from the FortiDAST GUI or through the FortiDAST cloud APIs. This mode supports the following configurations.
  - Exit after scan - The container exits after completing the scan; re-install the proxy server to rescan the asset.
  - Daemon - This is a one-time installation mode wherein, the container always runs in the background until you stop it. This configuration minimizes manual intervention.

- **DevOps** - This automates the scanning process along with a secure tunnel framework, and enables DAST scanning of web applications by integrating into CI/CD pipelines. When the container is invoked with the required configurations, it interacts with the FortiDAST cloud through REST APIs, and begins the scanning process automatically. This involves authorizing the asset, initiating the scan, waiting for the scan to complete, and stopping the container after the scan is complete.

## Configuring the Proxy Server

This section describes the pre-requisites, recommendations, and set-up procedure for configuring a proxy server in your environment.

### Pre-requisites

- Minimum system requirements are 4 GB RAM and 4 core CPU for running 10 proxy server Docker containers.
- Ensure that the VM hosting the proxy server container has internet connectivity through a NIC, this is required to fetch commands from the FortiDAST cloud and scan the application. Also, the VM must be able to reach the target web server through internal network.
- Ensure that the target machine is not reachable from FortiDAST through a public IP address and is accessible only from the proxy server.
- You can meet the aforementioned requirements either through a dedicated NIC or by configuring routing entries.
- Ensure that the SSH service is enabled and the firewall allows the communication between the proxy server and FortiDAST cloud (*IP: 34.72.80.123* and *Port: 22*).

### Recommendations

- Using Linux OS for configuring the proxy server, Windows OS is NOT supported.
- Do not run the proxy server Docker container and the target web server on the same machine.
- Whenever you generate new API keys, copy the updated Docker-compose file from the FortiDAST GUI, and bring up the proxy server again using the new compose file.

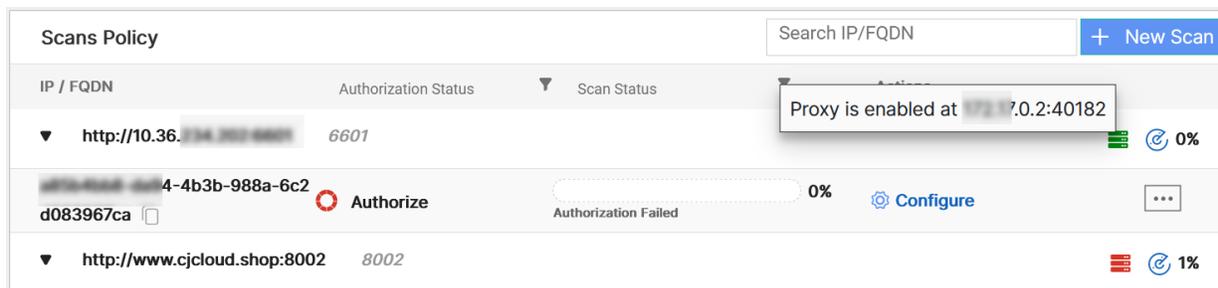
**Note:** Only IPv4 networking is supported for proxy servers. IPv6 is NOT supported.

1. Install a Linux version (Ubuntu or Centos) in a virtual machine or physical server.
2. Install the latest version of Docker engine and the Docker compose.

```
sudo apt install docker.io
sudo apt install docker-compose
```
3. Copy the Docker compose file (*docker-compose.yml*) from the **FortiDAST Proxy** tab of the scan configuration page in the GUI. See [FortiDAST Proxy](#).
4. Bring up the proxy server container and run the command, `sudo docker-compose -f <docker-compose.yml> up -d`.

You can view the status of the FortiDAST - proxy server feature in the **Scans Policy** page. The green icon indicates that communication is established between the proxy server container and the FortiDAST cloud service. Initially, the status is indicated by a red icon, that implies a lack of communication. Hover over the icon to view the port over which the communication is established.

Ensure that the status of the FortiDAST proxy status is green before initiating the scan for the target.



**Note:** When newer version of the FortiDAST becomes available, you must remove the existing FortiDAST proxy docker image from your virtual machine and pull the latest version.

- To remove existing FortiDAST proxy docker image run `docker rmi registry.fortidast.com/fptproxyserver` command.
- To pull the latest docker image run `docker pull registry.fortidast.forticloud.com/dastproxy` command.
- Also, ensure that you copy the latest Docker compose file (*docker-compose.yml*) from the **FortiDAST Proxy** tab of the scan configuration page in the GUI. See [FortiDAST Proxy](#).

## CI/CD Tools

FortiDAST Proxy server supports scanning in the following CI/CD tools.

- [Jenkins](#)
- [GitLab](#)
- [GitHub Actions](#)
- [Azure DevOps](#)

### Jenkins

Following is a sample code segment that can be configured in **Jenkins > (Your App) > Configure > Add build step > Execute Shell**.

**Note:** Make sure to update the parameters in the sample code according to your environment before using it.

```
export EMAIL=account_email LICENSE_SERIAL=your_serial_number ASSET_TOKEN=your_asset_token SCANURL=target_asset_url SCANTYPE=1 ASSET=asset_UUID
env | grep -E "EMAIL|LICENSE_SERIAL|ASSET_TOKEN|SCANURL|SCANTYPE|ASSET" > /tmp/env
docker pull registry.fortidast.forticloud.com/dastdevopsproxy:latest
docker run --rm --env-file /tmp/env --network=host
registry.fortidast.forticloud.com/dastdevopsproxy:latest
```

### GitLab

Following is a sample code segment that can be configured in *gitlab-ci.yml* file.

**Note:** Make sure to update the parameters in the sample code according to your environment before using it.

```
tags:
  - fptproxyserver
variables:
  EMAIL: "account_email"
  LICENSE_SERIAL: "your_serial_number"
  ASSET_TOKEN: "your_asset_token"
  SCANURL: "target_asset_url"
  SCANTYPE: "1"
  ASSET: "asset_UUID"
image: registry.fortidast.forticloud.com/dastdevopsproxy:latest
script:
  - cd /home/fortinet/
  - python3 proxyScan.py
```

### GitHub Actions

Following is a sample code segment that can be configured in *main.yml* file.

**Note:** Make sure to update the parameters in the sample code according to your environment before using it.

```
name: FortiDAST Proxy Server
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
jobs:
  build:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v2
      - name: FPT_PROXY_SERVER
        run: |
          export EMAIL=account_email LICENSE_SERIAL=your_serial_number ASSET_
          TOKEN=your_asset_token SCANURL=target_asset_url SCANTYPE=1 ASSET=asset_UUID
          env | grep -E "EMAIL|LICENSE_SERIAL|ASSET_TOKEN|SCANURL|SCANTYPE|ASSET" >
          /tmp/env
          docker pull registry.fortidast.forticloud.com/dastdevopsproxy:latest
          docker run --rm --env-file /tmp/env --network=host
          registry.fortidast.forticloud.com/dastdevopsproxy:latest
```

### Azure DevOps

Following is a sample code segment that can be configured in *azure-pipelines.yml* file.

**Note:** Make sure to update the parameters in the sample code according to your environment before using it.

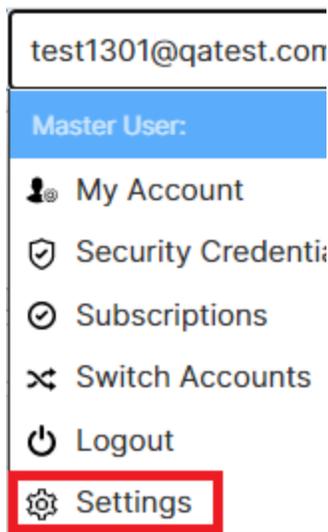
```
trigger:
  - main
pool:
  vmImage: ubuntu-latest

steps:
```

```
-task: Bash@3
  displayName: Install_Run_FortiDAST
  inputs:
    targetType: 'inline'
    script: |
      export EMAIL=account_email LICENSE_SERIAL=your_serial_number ASSET_
TOKEN=your_asset_token SCANURL=target_asset_url SCANTYPE=1 ASSET=asset_UUID
      env | grep -E "EMAIL|LICENSE_SERIAL|ASSET_TOKEN|SCANURL|SCANTYPE|ASSET" >
/tmp/env
      docker pull registry.fortidast.forticloud.com/dastdevopsproxy:latest
      docker run --rm --env-file /tmp/env --network=host
registry.fortidast.forticloud.com/dastdevopsproxy:latest
```

## Settings

You can configure the following FortiDAST settings from the home page. Click on the username and select **Settings**.



## Jira Integration

To gain access to the Jira server, configure the Jira server URL, email ID, and the API key for authentication. You can enable integration in [Jira Integration](#). For more information, see [Jira](#).

JIRA Integration

URL

Email ID

API Key

[VALIDATE](#)

## REST API

REST APIs are available to start and stop a scan, to query scan status, scan results, scan summary results, and all configured assets.

API Key Generation OAS3 ↓

---

Basic Privileged

---

Serial Number FFPENT0000222704 ▼

Validity (UTC) i Choose a date REGENERATE

GmzNFkmkPVKzVC3HRZZw4kv( 📄

i This is the only place where this key will be provided. Keep this information secure. The bearer of this API key will be granted all access privileges assigned to this serial number.

You can generate the key (per serial number) for REST API authentication. Copy and save the key as it is not retained in FortiDAST. Click on OAS3 ↓ to download API documentation in the swagger format, Swagger hub is required to use APIs from the swagger interface. You can skip SSL certificate validation.

- API keys generated for basic access are only for read operation.
- API keys generated for privileged access are for both read/write operations.
- API keys are allowed to be copied only at the time of generation.

## Email Notification

You can configure automated email notification settings to monitor product updates and ongoing scans with specific types of alerts.

## Email Notifications

Select the type of email notifications you would like to receive from FortiPenTest. Read our [email notifications guide](#) for more information.

## Choose Settings

- No Notifications**  
You will not receive any email notifications from FortiPenTest.
- Scan Notifications**  
Receive a scan summary every time we finish scanning one of your scan profiles.
- Custom Notifications**  
Customise your notifications by defining email triggers for each scan profile.
  - Scan Started**  
Receive an email when a scan starts.
  - Scan Finished**  
Receive a summary email when a scan is finished.
  - Critical Severity Findings**  
Receive an email every time a critical severity finding is detected.
  - High Severity Findings**  
Receive an email every time a high severity finding is detected.
  - Medium Severity Findings**  
Receive an email every time a medium severity finding is detected.

- **Subscribe for Email Updates** is enabled by default and sends email notifications for available product updates such as new features, upgrades and so on.
- Enable **Scan Notifications** to receive an email with the scan summary every time a vulnerability scan is completed.
- Enable **Custom Notifications** to receive email notifications indicating the scan progress such as the start, severity findings, and scan completion.  
**Note:** You receive custom email notifications only when these are enabled for specific actions globally and per asset. For example, to receive notifications for critical severity findings, you must enable it in the global settings as described in this section and also per asset as described in [Configuring the Scanner on page 22](#).

**Save** the configuration.

## FortiAppSec Cloud WAF Virtual Patching

FortiAppSec Cloud WAF virtual patching simplifies the process of addressing vulnerabilities detected during DAST scans. It combines FortiDAST's detection capabilities with FortiAppSec Cloud WAF's custom rule creation, allowing you to address security issues. The following fuzzers are supported for virtual patching.

- Server-Side Template Injection (SSTI)
- Expressive Language Injection (ELI)
- LDAP Injection (LDAPi)
- ASP PHP Code Injection
- Local File Inclusion (LFI)

Perform the following steps to integrate FortiAppSec Cloud WAF with FortiDAST.

1. Navigate to **DAST Settings > FortiAppSec Cloud WAF** section.
2. Add **API Key Secret** generated from FortiAppSec Cloud. See [FortiAppSec Cloud User Guide > Settings](#).

**3. Click **Validate**.**

FortiAppSec Cloud WAF

API Key Secret

Validate



Virtual patching is supported for FortiAppSec Cloud WAF only.

---

# Analyzing the Scan Data

You can view, analyze, and manage the scan result in two interactive and distinct GUI pages.

- [Scans Overview \(Scan Result\)](#)
- [Dashboard](#)

## Scans Overview (Scan Result)

The scan result for vulnerability assessment is populated in a comprehensive dashboard with graphical representation and visualization of statistics as a combination of summary charts and detailed data. You can access the **Scans Overview** page on the left navigation menu of the GUI or click on **Scan Result** in the **Scans Policy** page.

IP/FQDN	Scan Name	UUID	Auth/Scan Status	Scan Time	Threat level	Severity
<a href="#">192.168.1.100</a>		8ceddffe-7980-49cd-8305-8e2c14a95544	Success	2021-07-22 12:11:56 ,11 months ago	5.9	3 9 21 0
<a href="#">192.168.1.100</a>		46057dd9-e2a8-43ec-be61-d9588109ea15	Stopped	2021-07-19 06:53:34 ,11 months ago	0.0	0 0 0 0
<a href="#">192.168.1.100</a>		34305140-c6a4-4f53-bc58-25ac42e62d83	Success	2021-07-20 11:54:57 ,11 months ago	5.6	3 9 32 1
<a href="#">192.168.1.100</a>		ffa91b12-90cb-4595-89f8-88f03ed119ff	Success	2021-04-19 05:07:37 ,1 month ago	6.6	6 40 56 1
<a href="#">192.168.1.100</a>		492a90fb-e631-48ad-a157-d21d3a89e7b1	Success	2021-04-01 07:34:06 ,1 month ago	6.3	0 7 18 1
<a href="#">192.168.1.100</a>		0fe142bc-5cce-4bd4-b26a-9deb654d17be	Success	2021-07-12 15:01:29 ,11 months ago	5.7	72 112 138 5

- [Compare Scans](#)
- [Summary](#)
- [Virtual Patching](#)
- [Vulnerabilities](#)

## Compare Scans

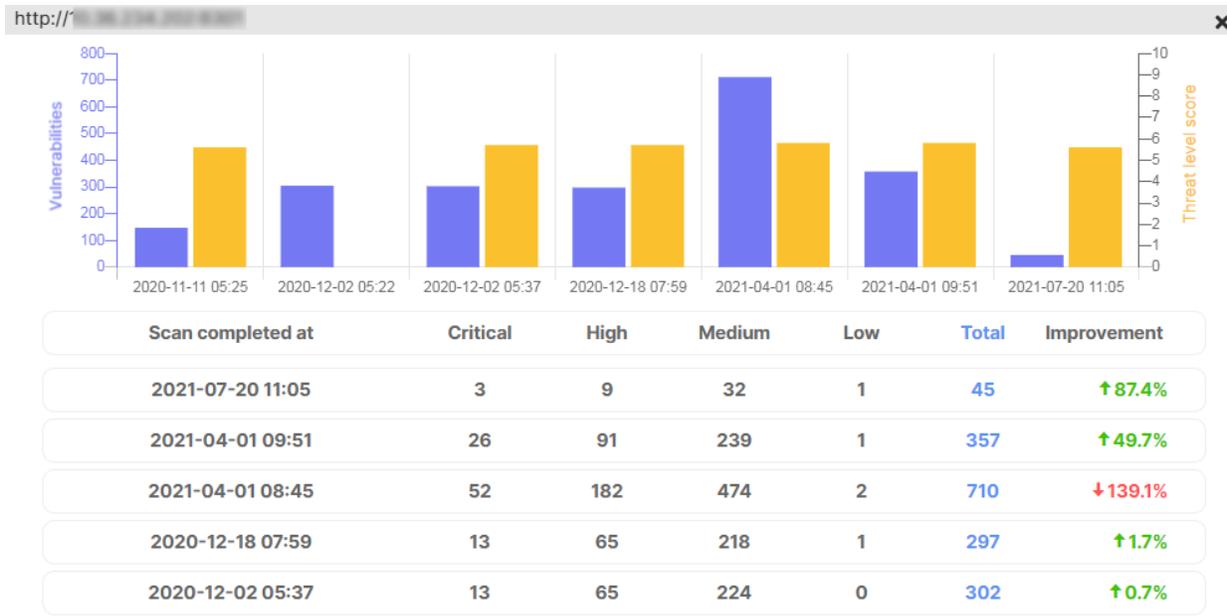
You can view and analyze the evolution of the asset threat level scores over a period of time that indicate an increase or decrease in the associated risk of the asset. This feature compares asset security assessment of past scans graphically and depicts a general threat mitigation trend.

To compare scans, perform the following steps.

1. Go to **Scans Overview**.
2. Select the asset that you want and click **Compare Scans**.



The x-axis displays the date and time of each scan and the y-axis displays the vulnerabilities' count and the threat level score. The data comparison is done at the asset level and the graphs are displayed for the last 10 scans.



## Summary

The dashboard is divided into donut/pie charts with each color coded wedge of the chart representing a particular count/percentage. Hover over different parts of the chart to view details.

Click **Outbreak Alerts** to view asset specific outbreak alerts detected after a successful scan.

- [Overview](#)
- [Activity](#)
- [Scan Status](#)
- [CVSS Score](#)
- [OWASP Category](#)
- [Top SANS Risks](#)
- [Technologies, Ports, and Services](#)
- [Exploit Engine](#)

## Overview

This section provides a summary of the scanned asset such as its URI, the associated port, the assigned UUID, scan status, and the date of the last scan (default). The **Scan Date** field displays all the scans conducted on the asset. You can filter and select scan details that you wish to view. The summary page is populated based on the selected scan.

The screenshot shows the 'Overview' section of the FortiDAST interface. It displays the following information:

- URI:** http://10.36.2.../dvwa
- PORT:** 9505
- UUID:** 7d1f3b2b-...ec7819281ff
- Scan Status:** Scan Complete
- Scan Date:** 2022-06-29 10:15

To the right, there is an 'Activity' section with a green progress bar and a circular gauge. A dropdown menu is open over the 'Scan Date' field, showing two entries:

Scan Date	Scan Status	Progress	Score
2022-06-29	Scan Complete	100%	5.9
2022-06-29	Stopped	0%	..

At the bottom left, there are buttons for 'Configure' and 'Rescan'.

You can also generate reports, rescan, and configure an asset from this section.

This screenshot shows the 'Overview' section of the FortiDAST interface, similar to the one above. The 'Scan Date' is 2022-06-29 10:15. At the bottom of the overview card, three buttons are highlighted with a red border:

- Configure**
- Rescan**
- Generate Report**

## Generate Report

You can generate reports to view the FortiDAST scan result in a *.pdf* format or the FortiWeb compatible scan report.

- [Exporting Scan Result to PDF](#)
- [Exporting Scan Result to FortiWeb WAF](#)

## Rescan

This option triggers a rescan for vulnerability assessment only after completing an ongoing scan (if any). Select the asset and then click **Rescan**.

## Configure

You can modify the scanner configuration. See [Configuring the Scanner](#).

## Exporting Scan Result to PDF

Perform the following steps to download FortiDAST scan result in *.pdf* format.

1. Navigate to **Scans Overview > Summary > Overview** section.
2. Click **Generate Report**.
3. In the **Download Reports > PDF Report** tab, select **Summary Report** or **Detailed Report** based on your requirement. By selecting detailed report, you will be able to:
  - Include proof of exploit and HTTP headers.
  - Filter vulnerabilities by type .
4. Select the desired categories from the **Category** drop down.
5. Optionally, you can password protect the report by selecting **Password Protection** and configuring the password.
6. Click **Download**.

**Download Reports**

PDF Report
WAF Report

---

Report :  Summary Report  Detailed Report

Category : New x +3

Password :  Password Protection

••••••••

## Exporting Scan Result to FortiWeb WAF

Perform the following steps to download FortiWeb compatible scan report. The report is downloaded in the *.xml* format.

1. Navigate to **Scans Overview > Summary > Overview** section.
2. Click **Generate Report**.
3. In the **Download Reports > WAF Report** tab, include/exclude detected vulnerabilities based on your requirement.
4. Click **Download**.

**Note:** These reports are available based on the scan timing selected in the summary.

Download Reports

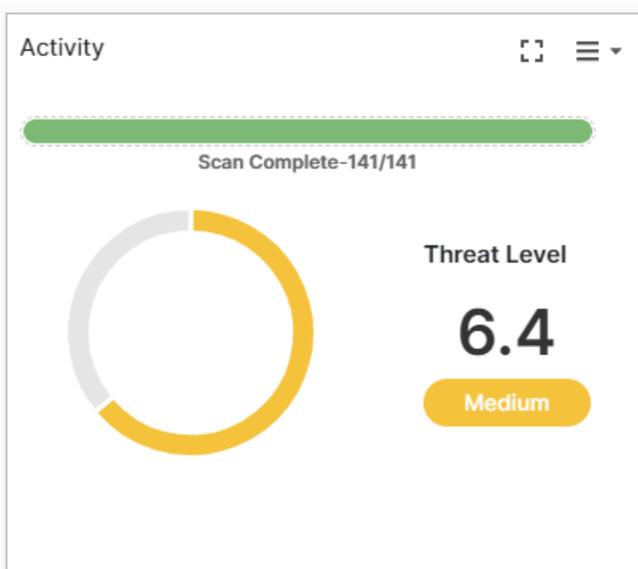
PDF Report **WAF Report**

WAF type : FortiWeb

Vulnerability Selection:  Identification and Authentication Failures  Session fixation  Security Misconfiguration

Activity

This section displays the current scan status and the progress of an ongoing scan along with the overall threat score.



The threat level score for a scanned asset to prioritize asset remediation. The threat level score is derived from the CVSS score categories and is represented by values between 0 and 10. The threat level scores are categorized based on the following severities.

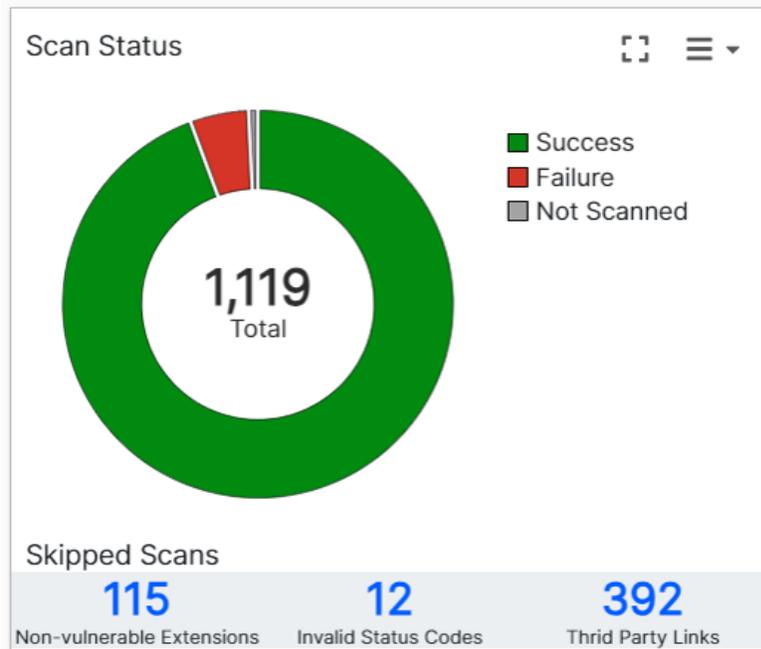
- **Low** - 0.1 - 3.9
- **Medium** - 4.0 - 6.9
- **High** - 7.0 - 8.9
- **Critical** - 9.0 - 10.0

## Scan Status

The vulnerability scan detects and assess URIs in the asset. The URI statistics displayed on the chart represent the total number of URIs detected (center of the chart) with each wedge of the chart representing the count/percentage of the following:

- The URIs with scan success
- The URIs with scan failure
- The URIs not scanned

The Scan Status widget also displays the details of skipped scans. See [Skipped Scans](#).



Clicking on the chart or on the displayed statistics brings up a list of URIs and APIs associated with the scanned asset. Click on any of the URI to view the all the detected vulnerability details, alternately, you can also filter the vulnerabilities data based on the severity by clicking on the severity type in the **Severity** column.

http://10.36.10.100/dvwa

[Select Multiple](#) [New x](#) +2

**Vulnerability Type:** Identification and Authentication Failures (CVSS Score: 5.8 Medium)

**Description :** The software allocates a reusable resource or group of resources on behalf of an actor without imposing any restrictions on the size or number of resources that can be allocated. It will be easy for an attacker to consume too many resource by rapidly making too many frequent requests, or causing larger resources to be used than is needed.

**Exploit Details :**

**Remediation :** Implement a timeout after a specific number of requests within a period of time or implement CAPTCHA mechanism on the form pages when the requests originate from a single source or use WAF to filter the traffic

**References :**

[New](#)

### Skipped Scans

FortiDAST skips scanning of URIs or files that are potentially safe.



The *Skipped Scans* section in *Scan Status* widget displays the total number of URIs detected for the following.

URI Type	Description
<b>Invalid status code</b>	FortiDAST skips URLs that have response status codes outside the following list: <i>200, 302, 403, 405, 500, 501, 502, 503, 504, 505, 506, 507, 508, 510, and 511.</i>
<b>Non-vulerable extensions</b>	FortiDAST skips file extensions that are considered non-vulnerable. Common examples of the file extensions that are skipped include <i>css, json, jpeg, png, mp3, mp4, pdf, and docx.</i>
<b>Third party links</b>	FortiDAST skips scanning of the links from external domains which are part of the target URL.

To view detailed information, click the number of URIs associated with each skipped scan type.

URI LIST
API LIST

URI's

---

http://testasp [redacted] /showthread.asp?id=73

---

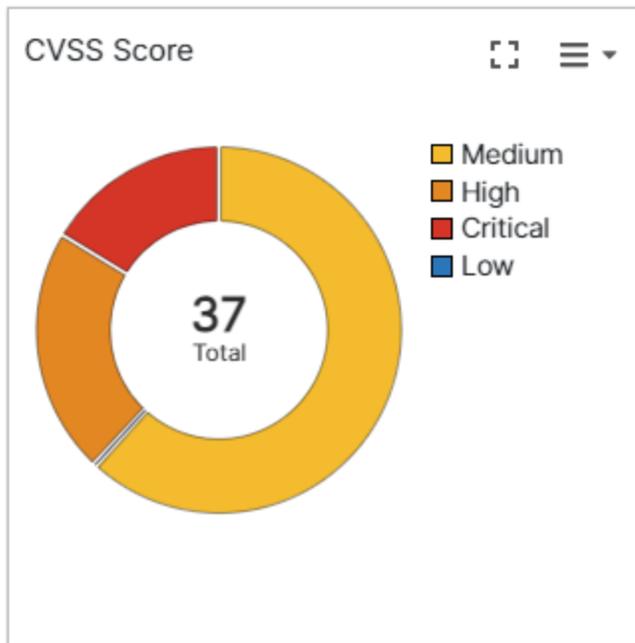
http://testasp [redacted] /showthread.asp?id=69

---

http://testasp [redacted] /showthread.asp?id=46

### CVSS Score

The detected URIs are scanned for vulnerabilities and are categorized based on severity. The vulnerability statistics displayed on the chart represent the total number of vulnerabilities found in the scanned URIs (center of the chart) with each swedge of the chart representing the count/percentage of vulnerabilities with **Critical**, **High**, **Medium**, and **Low** severity. The severity categorization is done based on the CVSS score.



Clicking on the chart or on the displayed statistics brings up a list of URIs categorized based on the detected vulnerabilities. Expand any of the displayed vulnerability categories to view details such as the CVSS score and the suggested remediation.

CVSS Score

[URI LIST](#)   [API LIST](#)

---

▶ Cryptographic Failures

---

▶ Identification and Authentication Failures

---

▶ Security Misconfiguration

---

▶ Vulnerable and Outdated Components

---

You can rescan vulnerable URIs that score a particular CVSS severity, select the required severity bar and then click **Rescan**, a partial scan is triggered.

CVSS Score

[URI LIST](#)   [API LIST](#)

▼ Cryptographic Failures

URI	CVSS SCORE	REMEDIATION
http://10.36.100.100/dvwa/login.php	5.3	Issue and apply SSL/TLS Certificate from a trusted authority.

▶ Identification and Authentication Failures

▶ Security Misconfiguration

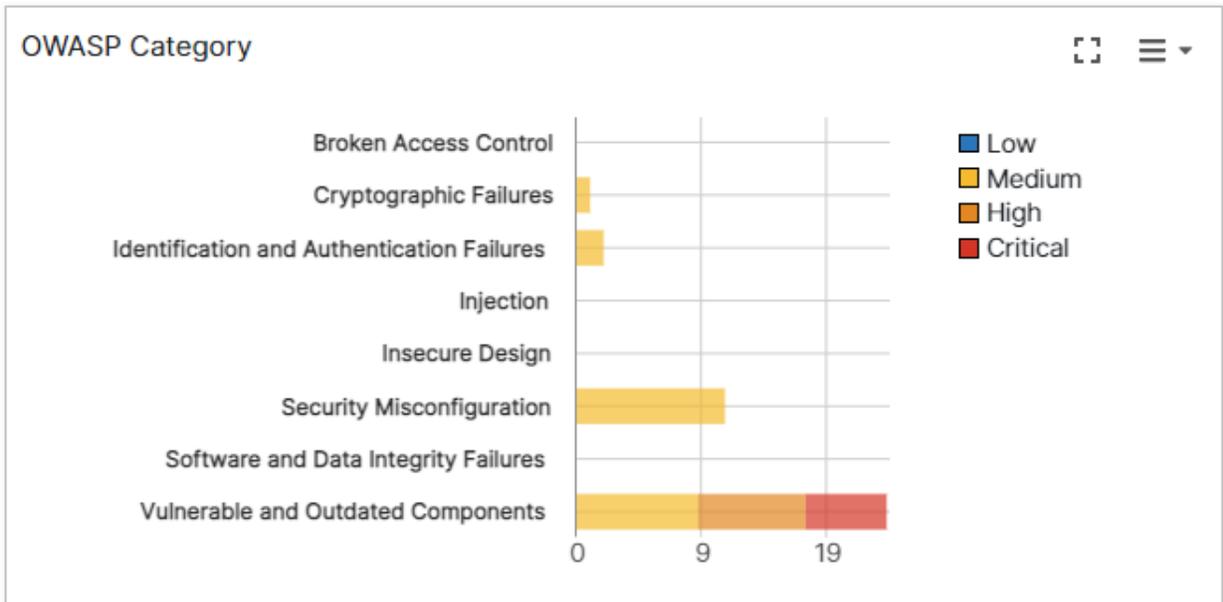
▶ Vulnerable and Outdated Components

RESCAN

## OWASP Category

The OWASP Top 10 - 2021 category based statistics found on the scanned asset are displayed on the chart. The category based statistics displayed on the chart represent the total number of vulnerabilities found (center of the chart) with each wedge of the chart representing the count/percentage of vulnerabilities. Clicking on this chart brings up a tabular view of the vulnerabilities categorized as **Critical**, **High**, **Medium**, and **Low**.

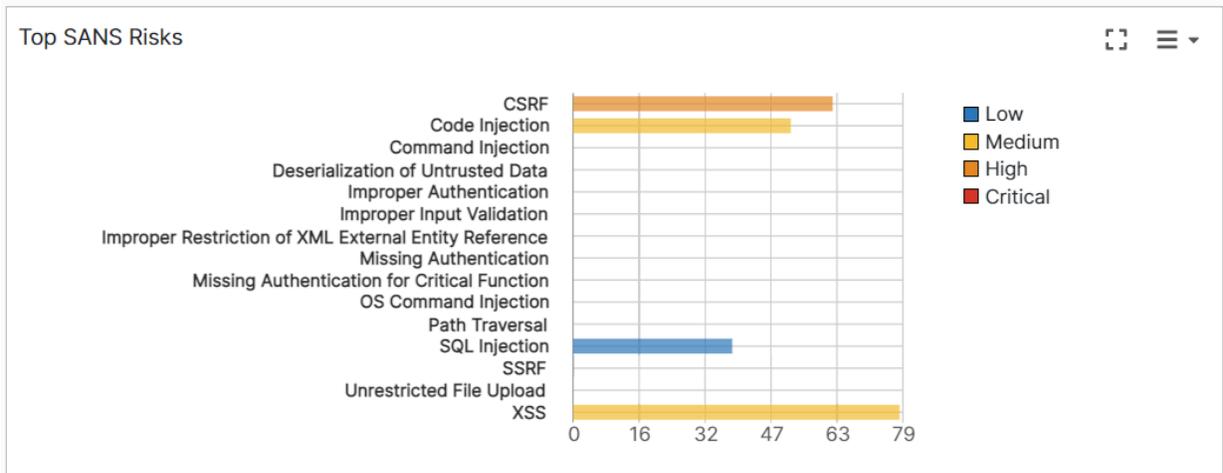
FortiDAST provides vulnerability assessment for the OWASP security risks. For details, see section [What is FortiDAST](#).



**Note:** To scan your asset for the **Insufficient Logging and Monitoring** category, email at [fgpt@fortinet.com](mailto:fgpt@fortinet.com) to engage for manual penetration testing.

### Top SANS Risks

The SANS category based statistics found on the scanned asset are displayed on the chart. The category based statistics displayed on the chart represent the total number of vulnerabilities found (center of the chart) with each wedge of the chart representing the count/percentage of vulnerabilities. Clicking on this chart brings up a tabular view of the vulnerabilities categorized as **Critical, High, Medium, and Low**.



Currently, 15 out of the SANS top 25 vulnerabilities are supported. The supported SANS categories are:

ID	Name
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') XSS

ID	Name
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-20	Improper Input Validation
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-434	Unrestricted Upload of File with Dangerous Type
CWE-502	Deserialization of Untrusted Data
CWE-287	Improper Authentication
CWE-862	Missing Authorization
CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')
CWE-306	Missing Authentication for Critical Function
CWE-918	Server-Side Request Forgery (SSRF)
CWE-611	Improper Restriction of XML External Entity Reference
CWE-94	Improper Control of Generation of Code ('Code Injection')

## Technologies, Ports, and Services

This tab displays the details of the scanned target asset, such as, technologies, ports, and services used by the server and any firewalls detected during vulnerability scanning. The data in the summary is generated by the reconnaissance engine.

Technologies found in the server 📌 🗨️ ☰

**Programming languages**

php **PHP**

**Operating systems**

 **UNIX**

**Web servers**

 **Apache**

▼

Port	Service	Version
22/tcp	ssh	OpenSSH/7.6p1 Ubuntu 4ubuntu0.6
8000/tcp	irdmi	
8002/tcp	teradataordbms	
8080/tcp	http-alt	Apache/2.4.7 (Ubuntu)
8090/tcp	opsmessaging	
8540/tcp		
8541/tcp		
8843/tcp		
8880/tcp	cddbp-alt	
9501/tcp		

**Notes:**

- The technologies on the target asset are displayed only for full scans (*Licensed*).
- Quick scan performs only partial port scan (*Evaluation, FortiCloud Premium Trial, and Licensed*).
- The port scan results vary based on the target infrastructure/environment (target behind firewall, WAFs).

## Exploit Engine

Displays the vulnerabilities detected by the configured FortiDAST [Exploit Engine](#).

Exploit Engine 📌 🗖️ ☰

Select Multiple New x +2

---

🔗 URI: <http://10.36.234.202:8014/> [\(CVSS Score: 8.8\)](#)

📄 Description : The XML Data Archiving Service (XML DAS) in SAP NetWeaver AS Java does not check authorization, which allows remote authenticated users to obtain sensitive information, gain privileges, or possibly have unspecified other impact via requests to (1) webcontent/cas/cas\_enter.jsp, (2) webcontent/cas/cas\_validate.jsp, or (3) webcontent/aas/aas\_store.jsp, aka SAP Security Note 1945215.

🔍 Exploit Details : Vulnerability Title:CVE-2015-8840 SAP NetWeaver J2EE Data Archiving Service Unauthorized Access  
 Vulnerability State:Exploitable  
 Script Name:cve-2015-8840-sap-netweaver-j2ee-das-unauthorized-access.fse

🛠️ Remediation : Apply latest patch from the vendor

🌐 References : <https://erpscan.io/advisories/erpscan-15-017-sap-netweaver-j2ee-das-service-unauthorized-access/>,<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-8840>

New ⌵

## Vulnerabilities

The URI/API list displays the scan result per URI/API for you to analyze and remediate the issues found. Click on each row to view details such as the vulnerability description, reasons for failure, CVSS score, EPSS percentile, suggested remediation, and outbreak alerts.

Summary		Vulnerabilities				Outbreak Alerts		
URI LIST		Search						
URI	Scan Status	Start Time	End Time	Total	Severity			
<input type="checkbox"/> http://[redacted]/login.php	Success	2023/06/28 15:49:54	2023/06/28 16:02:42	74	17	28	29	0

Click on the URI to view and modify the status of each vulnerability or of all vulnerabilities, select **Select Multiple** and set the status. You can also filter the vulnerabilities based on the assigned status.

Click **Outbreak Alerts** to view asset specific outbreak alerts detected after a successful scan.

### Notes:

- The EPSS percentile is displayed for vulnerabilities that have an associated CVE ID. This value is displayed only for vulnerable and outdated components.
- You are required to re-scan the previously scanned assets (from older releases) to review the associated vulnerabilities and to obtain the *Proof of Exploit* in the result dashboard and reports.
- Click outbreak alert link to navigate to the FortiGuard Outbreak Alert page for in-depth analysis.

http://10[redacted]:8090/s/cfx/\_;/WEB-INF/web.xml

**Select Multiple** **New x** +2

**Vulnerability Type:** Security Misconfiguration [\(CVSS Score: 4.3 Medium\)](#)

**Description:** Information disclosure refers to revealing sensitive information, such as personally identifiable information, to parties that are not supposed to have access to the subject matter in normal circumstances.

**Exploit Details:** A GET request was sent to the target http://10[redacted]:8090/s/cfx/\_;/WEB-INF/web.xml. FortiDAST's Secret Finder found Arbitrary File Read : /json/startheartbeactivity.action /admin/appTrustCertificate /images/icons /rpc/xmlrpc in the HTTP Response body leading to Information Disclosure.

**Remediation:** Do not embed information in webpages if it is not supposed to be available to the public.

**References:**

**Outbreak Alerts:** [Atlassian Confluence and JIRA Server Vulnerabilities \(CVE-2021-26085\)](#)

**New**

The following status categories are supported.

- **New:** This is a new vulnerability detected by the scan.
- **Confirmed:** This is a real vulnerability and requires a fix.
- **In Review:** This vulnerability is currently in review/looked into for further action.
- **Reviewed:** This vulnerability review is complete.
- **Fixed:** This vulnerability is fixed and does not appear in the next scan result.
- **Risk Accepted:** This vulnerability is an accepted risk and continues to exist without any potential damage.

- **False Positive:** This vulnerability is a potential flaw in the scanner or is indicative of a unique feature of the application.
- **Removed:** This vulnerability is overlooked in the application.
- **Reopened:** This is a fixed vulnerability detected again in the rescan and requires to be addressed. This state is assigned by FortiDAST.

URI	Scan Status	Start Time	End Time	Total	Severity
http://10.36.234.29514/fi_json	Success	2022/06/24 15:18:57	2022/06/24 15:20:05	3	0 3 0 0
http://10.36.234.29514/fi_xml	Success	2022/06/24 15:18:57	2022/06/24 15:20:54	4	0 3 1 0
http://10.36.234.29514/fi_path/{payload}	Success	2022/06/24 15:18:56	2022/06/24 15:19:14	0	0 0 0 0

## Virtual Patching

This tab displays a list of supported vulnerabilities available for virtual patching discovered during the last scan. For each vulnerability, *URL*, *Vulnerability Name*, *Severity*, *Details*, *Type (Parameter or URL)*, and *Signature* is displayed.

URL	Vulnerability Name	Severity	Details	Type	Signature
eli_pentest	Signature1				
https://.../injection-get.jsp (Patched)	eli_pentest	High		parameter	{(w*ld*)=(%24 %23 %2524 %2523)}{0\$
https://.../injection-post.jsp (Patched)	eli_pentest	High		parameter	{(w*ld*)=(%24 %23 %2524 %2523)}{0\$

Click **Details** icon to view detailed information.

A label next to the URL indicates the patch status.

- **New:** Newly identified vulnerability after a scan.
- **Patched:** Patch has been applied in FortiAppSec Cloud WAF.
- **Not Patched:** Patch was deleted in FortiAppSec Cloud WAF but is still present in FortiDAST.

### Applying Virtual Patch



The target application must be added in FortiAppSec Cloud WAF before applying virtual patches. See [FortiAppSec Cloud User Guide > WAF setup](#).

Perform the following steps to apply a virtual patch.

1. Ensure the FortiAppSec Cloud WAF is integrated with FortiDAST. See [FortiAppSec Cloud WAF Virtual Patching](#).
2. Go to **Scans Overview**.
3. Select a asset and click **Virtual Patching** tab.
4. Select a vulnerability and click **Apply**.
5. You can review the applied patch by navigating to **FortiAppSec Cloud > WAF > Application > Advanced Applications > Custom Rule**. See [FortiAppSec Cloud User Guide > Custom Rule](#).

## Deleting Virtual Patch

Perform the following steps to delete a virtual patch.

1. Go to **Scans Overview**.
2. Select a asset and click **Virtual Patching** tab.
3. Select a vulnerability and click **Delete**.



Deleting a patch in FortiDAST will only delete the filter in FortiAppSec Cloud WAF custom rule.

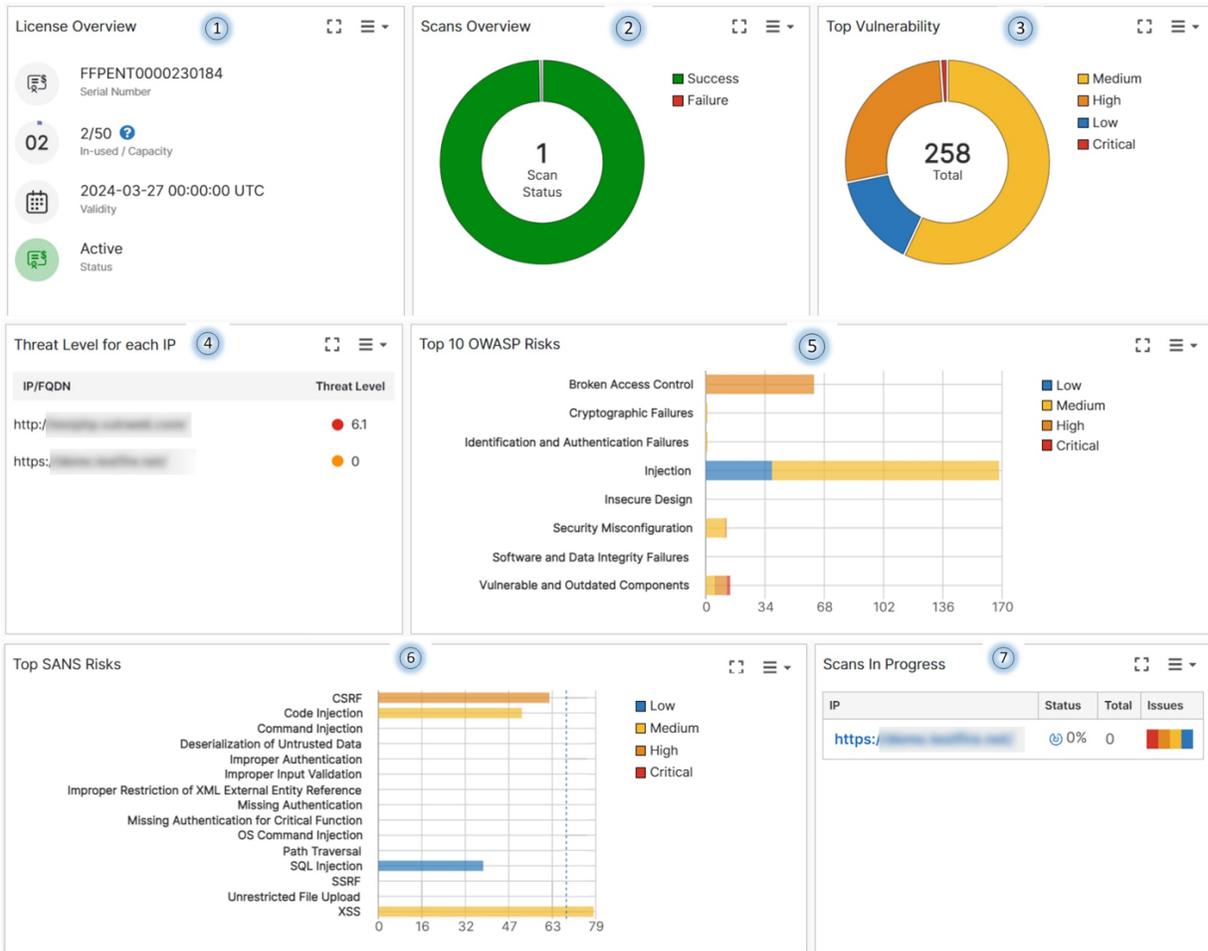
---

## Synchronization

Click **Sync** to update the latest patch status from FortiAppSec Cloud WAF. Patch that was deleted in FortiAppSec Cloud WAF but present in FortiDAST are marked **Not Patched**.

## Dashboard

The dashboard provides an insight into the scanned assets based on the vulnerabilities, threat levels, and OWASP Top 10 - 2021 vulnerabilities' categorization. Based on the vulnerability scanning, details for the top 5 assets are provided in the dashboard. These details are for the assets listed in the **Threat Level for each IP**.



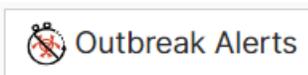
The dashboard is divided into donut/pie charts with each color coded wedge of the chart representing a particular count/percentage. Hover over different parts of the chart to view details.

Section	Description
1	Provides an overview of your subscription status. The license status, validity, serial number, and the number of scans allowed/conducted are displayed.
2	The total number of scans conducted and the status of the scans, whether successful or failed.
3	Categorizes the detected vulnerabilities as medium, high, critical, and low. Click on a section of the chart to view the vulnerability details. See <a href="#">CVSS Score</a> .
4	The overall threat level score for a scanned asset to prioritize asset remediation. See <a href="#">Compare Scans on page 73</a> .
5	The OWASP Top 10 - 2021 category based vulnerability statistics displayed on the chart. See <a href="#">OWASP Category</a> .

Section	Description
6	The SANS category based vulnerability statistics displayed on the chart. See <a href="#">Top SANS Risks</a> .
7	Displays the vulnerability scans in progress with the status and the vulnerabilities detected.

## Outbreak Alerts

Click **Outbreak Alerts** in the top right corner of the dashboard page to view all the outbreak alerts detected in the top 5 assets scanned.



**Outbreak Alerts** x

**Redigo Attack (CVE-2022-0543)** 8 Dec, 2022   Severity Critical

Go based malware that targets Redis server's vulnerability CVE-2022-0543 allowing threat actors to drop the Redigo malware and gain server access.

Click on the name of the alert to access additional information. Each alert in the Outbreak Alerts pane includes:

- Name of the alert
- Severity level
- Last revised date
- Description of the vulnerability

Following are the supported vulnerabilities for outbreak alerts in FortiDAST.

CVE	Vulnerability
CVE-2021-26085	Atlassian Confluence Server Pre-Authorization Arbitrary File Read Vulnerability.
CVE-2021-26086	Atlassian JIRA Path Traversal Vulnerability.
CVE-2022-0543	Debian-specific Redis Server Lua Sandbox Escape Vulnerability.
CVE-2022-22963	Spring Cloud Function 3.1.6, 3.2.2 and older Remote Code Execution Vulnerability.
CVE-2022-22965	Spring Framework 5.2.x/5.3.x Remote Code Execution Vulnerability.
CVE-2022-22980	Spring Data MongoDB SpEL Expression injection vulnerability through annotated repository query methods.
CVE-2022-35914	GLPI PHP code injection via httmlawed module.
CVE-2023-23752	Joomla improper access check.

CVE	Vulnerability
CVE-202-22205	Remote Code Execution vulnerability in Gitlab CE/E.
CVE-2021-44228	Apache Log4j JNDI Injection (aka Log4Shell).
CVE-2021-22005	VMware vCenter Server 6.7 - 6.7 Update 3o and 7.0 - 7.0 Update 2c Customer Experience Improvement Program (CEIP) service unauthenticated arbitrary file upload vulnerability.
CVE-2021-21974	VMWare ESXi OpenSLP Unauthenticated Remote Code Execution.
CVE-2021-45046	Apache-log4j-jndi-injection-log4shell-bypass.
CVE-2021-40539	Zoho ManageEngine ADSelfService Plus authentication bypass vulnerability.
CVE-2021-41773	Apache HTTP Server Path Traversal.
CVE-2021-42013	Apache HTTP Server 2.4.50 Path Traversal.
CVE-2022-41082	Microsoft Exchange Proxynotshell Remote Code Execution.
CVE-2022-46169	Cacti command injection vulnerability.
CVE-2023-28121	WooCommerce Payment WordPress Plugin authentication bypass to gain administrative privileges.
CVE-2023-35078	MobileIron Core Unauthenticated API Access Vulnerability.
CVE-2023-27350	PaperCut MF/NG Improper Access Control Vulnerability.
CVE-2021-35394	Realtek Jungle SDK Remote Code Execution Vulnerability.
CVE-2023-33246	Apache RocketMQ Remote Code Execution Vulnerability.
CVE-2023-1389	TP-Link Archer AX-21 Command Injection Vulnerability.
CVE-2023-42793	JetBrains TeamCity Authentication Bypass Attack.
CVE-2022-29303	SolarView Compact Command Injection Vulnerability.
CVE-2017-11317	Progress Telerik UI Attack.
CVE-2023-4966	Citrix Bleed Attack.
CVE-2023-26360	Adobe ColdFusion Deserialization of Untrusted Data Vulnerabilities.
CVE-2018-9995	TBK DVR Authentication Bypass Attack.
CVE-2023-20887	VMware Aria Operations for Networks Command Injection Vulnerability.
CVE-2024-20767	ColdFusion versions 2023.6, 2021.12 and earlier are affected by an Improper Access Control.
CVE-2024-27198	A critical authentication bypass vulnerability in the web component of JetBrains TeamCity versions before 2023.11.4.
CVE-2024-3400	Palo Alto Networks PAN-OS Command Injection Vulnerability.

CVE	Vulnerability
CVE-2022-4257	Unauthorized RCE vulnerability in the C-Data web management system.
CVE-2024-24919	Arbitrary File Read in Check Point SVN, which allows an attacker to read certain information on Check Point Security.
CVE-2024-1709	ConnectWise ScreenConnect 23.9.7 and prior are affected by an Authentication Bypass to RCE.
CVE-2024-3273	A vulnerability in D-Link NAS devices that allows remote attackers to execute arbitrary commands via a crafted HTTP request to the cgi-bin/nas_sharing.cgi endpoint.
CVE-2021-40655	Sensitive information disclosure vulnerability in D-Link dir-605 Hardware.
CVE-2014-100005	D-Link DIR-600 routers with firmware before 2.17b02 has Cross-Site Request Forgery.
CVE-2024-4577	A critical PHP remote code execution vulnerability.
CVE-2022-40881	SolarView Compact 6.00 was discovered to contain a command injection vulnerability via network_test.php.
CVE-2024-4879	Jelly Template Injection on ServiceNow.
CVE-2024-5217	ServiceNow - Incomplete Input Validation.
CVE-2023-29298	Adobe ColdFusion Access Control Bypass.
CVE-2023-38205	
CVE-2024-36104	Apache OFBiz - Path Traversal.
CVE-2024-38856	Apache OFBiz - Remote Code Execution.
CVE-2024-22024	A XXE vulnerability in the SAML component of Ivanti Connect Secure, Ivanti Policy Secure and ZTA gateway.
CVE-2024-21893	A server-side request forgery vulnerability in the SAML component of Ivanti Connect Secure.
CVE-2019-7256	Nice Linear eMerge Command Injection Vulnerability.
CVE-2021-26084	Confluence Server OGNL RCE.
CVE-2021-33044	An Authentication Bypass Vulnerability in Dahua Products.
CVE-2021-33045	
CVE-2024-23897	Jenkins Arbitrary File Read Vulnerability.
CVE-2022-26138	Atlassian Questions for Confluence App Hardcoded Credentials Vulnerability.
CVE-2024-5910	Palo Alto Expedition Missing Authentication Vulnerability.
CVE-2024-9463	

CVE	Vulnerability
CVE-2024-9474	Palo Alto Networks Management Interface Attack.
CVE-2024-1212	Progress Kemp LoadMaster OS Command Injection Vulnerability.
CVE-2024-41713	Path Traversal vulnerability in Mitel MiCollab.
CVE-2024-8963	Path Traversal Vulnerability in Ivanti Cloud Service Appliance.

**Notes:**

- To detect Debian-specific Redis vulnerability - *CVE-2022-0543* as an outbreak alert, FortiDAST Scripting Engine (FSE) must be enabled.
- To detect *CVE-2021-26085* or *CVE-2021-26086* as an outbreak alert, you must add the base URL of Atlassian Confluence or JIRA as target respectively.

# Integrations

FortiDAST supports the following integrations.

- Web Application Firewall (WAF)
  - [FortiWeb Appliances](#)
  - [FortiAppSec Cloud WAF](#)
- DevOps
  - Native integrations
    - [GitLab Setup](#)
    - [Jenkins Setup](#)
  - Proxy based scans
    - [FortiDAST Proxy Server](#)
- Issue tracking
  - [Jira](#)

## WAF Integration

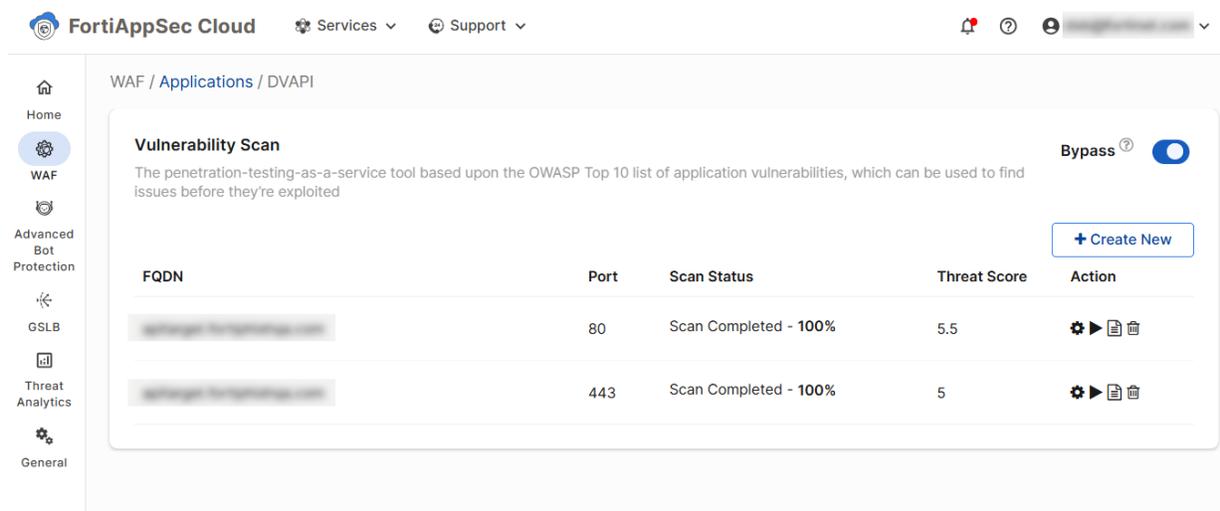
FortiDAST supports the following Web Application Firewall (WAF) integrations.

- Vulnerability scanning of applications configured behind FortiAppSec Cloud WAF. See [FortiAppSec Cloud WAF](#).
- Automatic and manual generation of WAF rules to FortiWeb on-premises. See [FortiWeb Appliances](#).

## FortiAppSec Cloud WAF

You can perform FortiDAST vulnerability scans within FortiAppSec Cloud WAF. See [Vulnerability Scan](#) in *FortiAppSec Cloud User Guide* for more information.

See [FortiAppSec Cloud WAF Virtual Patching](#) to validate API Secret generated from FortiAppSec Cloud.



## FortiWeb Appliances

FortiDAST integration with FortiWeb appliances deployed on-premises offers two options for generating WAF rules.

- Automatic WAF rule generation
  - a. In the [WAF Configuration](#) page, specify the FortiWeb appliance details like IP address, username, password, and VDOM name.
  - b. Upon completion of the scan, an XML file containing identified vulnerabilities is generated.
  - c. FortiWeb appliance automatically parses the XML and dynamically creates corresponding WAF rules based on the configured actions and supported vulnerabilities.
- Manual WAF rule generation
  - a. Navigate to **Scans Overview > Summary > Overview** and download the report in XML format. See [Exporting Scan Result to FortiWeb WAF](#).
  - b. You can manually upload the downloaded XML report to FortiWeb appliance to create WAF rules based on the reported vulnerabilities.

**Note:** Automatic WAF rule generation is currently only supported for FortiWeb appliances deployed on-premises.

## DevOps Integration

You can integrate FortiDAST plugin with Jenkins and with GitLab for Continuous Integration/Continuous Deployment to trigger automated vulnerability assessment scans.

The FortiDAST Jenkins plugin and GitLab CI/CD establish a connection with the FortiDAST server using an API key. Ensure that your asset is successfully authorized before automated scans are triggered for each build in FortiDAST.

- [Jenkins Setup](#)
- [GitLab Setup](#)

You can scan your internal web applications that are not exposed to the internet, by integrating FortiDAST proxy into CI/CD pipelines. See [FortiDAST Proxy Server](#).

## Jenkins Setup

Integrate the FortiDAST plugin with Jenkins to trigger vulnerability assessment scans as part of the build process. Click [here](#) to download the FortiDAST plugin.

Perform the following steps to **install** the FortiDAST plugin for Jenkins.

1. In the Jenkins GUI, navigate to *Manage Jenkins > Manage Plugins*. The **Plugin Manager** page is displayed.
2. On the Jenkins **Plugin Manager** page, click on the **Available** tab and search **FortiDAST**.
3. Select the FortiDAST plugin.
4. Click on **Install without restart**.

(Optional) When the installation is complete, select **Restart Jenkins when installation is complete and no jobs are running**.

Perform the following steps to add the FortiDAST SSL certificate to the Jenkins.

1. Download FortiDAST SSL certificate and locate the certificate store path to install it in the Jenkins CA store.
2. Identify the *JAVA\_HOME\_FOLDER* used by the Jenkins service (The *executable* tag *Jenkins.xml* file in the installation directory gives the location of Java folder used by Jenkins).
3. To install the SSL certificate in Windows, navigate to the *bin* folder and run the following command.  
**keytool -import -trustcacerts -alias FortidastCA -keystore "C:\Program Files\Java\jdk-11.0.10\lib\security\cacerts" -file "<FortiDAST SSL Certs Location Path>"**.

Perform the following steps to **integrate** FortiDAST and Jenkins.

1. In the Jenkins GUI, navigate to *Dashboard > Manage Jenkins > Configure System* and scroll down to the **FortiDAST** plugin.
2. Verify the **FortiDAST API URL** that is auto-populated after the plugin installation. The default URL is *https://fortidast.com/api/v1.0*.
3. Enter the FortiDAST server **UserName**.
4. Validate the user name and click **Apply**.

The screenshot shows a configuration form titled "FortiPenTest". It contains two text input fields. The first field is labeled "FortiPenTest API URL" and contains the text "https://fortipentest.com/api/v1.0". The second field is labeled "FortiPenTest UserName" and contains the text "test1303@qa@test.com". To the right of each field is a blue question mark icon. At the bottom right of the form is a "Validate" button.

Perform the following steps to add the FortiDAST **Build Step** in a Jenkins Job. You can modify an existing job or create a new job. Obtain the API key from FortiDAST GUI. See section [REST API on page 70](#).

**Note:** Only the **Privileged** key is supported.

1. In the Jenkins job, scroll down to **Build**, select **FortiDAST** as the **Build Step** and update the following.
  - **Scan Type** - Select **Quick Scan** or **Full Scan** for your asset.
  - **API Key** - Click **Add** to add credentials and update; select **Secret text** as the **Kind**, enter the **API key** in the **Secret** field, and enter a unique **ID**.

**Jenkins Credentials Provider: Jenkins**

**Add Credentials**

Domain: Global credentials (unrestricted)

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: [Masked]

ID: test1303

Description: [Empty]

2. Select the **ID** from the **API Key**, the assets associated with the FortiDAST account (user name) are populated in the **Scan Target** field.
3. Select the asset to be scanned in the **Scan Target** field.

**Build**

**FortiPenTest**

Scan Type: Quick Scan

Scan Target: http://10.36...

API Key: test1303 [Add]

[Add build step]

4. Click **Save**.

Click **Build Now**, a scan is triggered on the asset. After the scan is complete, you can view the **Scan Summary Report** in Jenkins.

## OWASP CATEGORY

Scan URL - <http://10.36.234.202:6601>

Vulnerability Type	Enabled	Total	Critical	High	Medium	Low	None
Injection	true	0	0	0	0	0	0
Broken Access Control	true	0	0	0	0	0	0
Cryptographic Failures	true	1	0	0	1	0	0
Security Misconfiguration	true	13	0	0	12	1	0
Vulnerable and Outdated Components	true	40	11	18	11	0	0
Identification and Authentication Failures	true	1	0	0	1	0	0
Software and Data Integrity Failures	true	0	0	0	0	0	0
Insecure Design	true	0	0	0	0	0	0

## GitLab Setup

Integrate the FortiDAST with GitLab for CI/CD.

Perform the following steps to configure FortiDAST with GitLab.

1. Login into the GitLab setup and select a project.
2. Click **CI/CD configuration**. This opens a CI/CD editor to update/create the yml file, *.gitlab-ci.yml*.
3. Copy the following contents in the editor and update the highlighted variables.

```
ScanJob:
  tags:
    - Your tag name

  before_script:
    - pip3 install requests

  variables:

    #input variable(string type)
    scanurl: "Your scan url"

    #input variable(string type)
```

```

uuid: "Your asset uuid"

#input variable(string type)-Example:https://fortidast.com/api/v1.0
apiUrl: "FortiDAST API Url"

#input variable(string type)
apikey: "Your API Key"

script:
- python scan.py $apiUrl $scanurl $uuid $scantype $apikey

#input variable(int type)Quick scan=0,Full scan=1
scantype: either 0 or 1

```

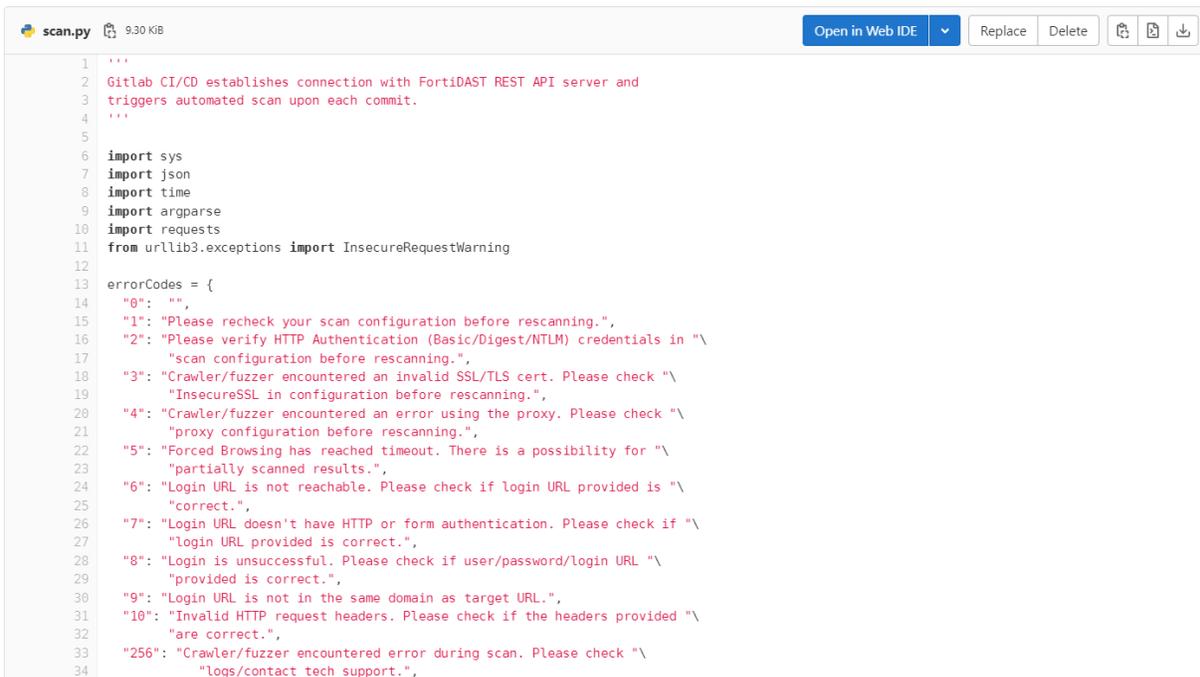
- Enter the **tag name** used while registering the runner. See <https://docs.gitlab.com/runner/register/>.
- **scanurl** - The URL of the asset to scan.
- **uuid** - The asset UUID. See [Asset Authorization on page 20](#).
- **scantype** - Type 0 for quick scan and 1 for full scan.
- **apiURL** - The FortiDAST URL.
- **apiKey** - Obtain the API key from FortiDAST GUI. See [REST API on page 70](#).

**Note:** Only the **Privileged** key is supported.

Click **Commit Changes**. The `.gitlab-ci.yml` file is created.

Commit the file `scan.py`; click [here](#) to download the file.

This image is a snapshot of the added file.



```

scan.py 9.30 KiB
Open in Web IDE Replace Delete
1 """
2 Gitlab CI/CD establishes connection with FortiDAST REST API server and
3 triggers automated scan upon each commit.
4 """
5
6 import sys
7 import json
8 import time
9 import argparse
10 import requests
11 from urllib3.exceptions import InsecureRequestWarning
12
13 errorCodes = {
14     "0": "",
15     "1": "Please recheck your scan configuration before rescanning.",
16     "2": "Please verify HTTP Authentication (Basic/Digest/NTLM) credentials in \"\
17     scan configuration before rescanning.",
18     "3": "Crawler/fuzzer encountered an invalid SSL/TLS cert. Please check \"\
19     InsecureSSL in configuration before rescanning.",
20     "4": "Crawler/fuzzer encountered an error using the proxy. Please check \"\
21     proxy configuration before rescanning.",
22     "5": "Forced Browsing has reached timeout. There is a possibility for \"\
23     partially scanned results.",
24     "6": "Login URL is not reachable. Please check if login URL provided is \"\
25     correct.",
26     "7": "Login URL doesn't have HTTP or form authentication. Please check if \"\
27     login URL provided is correct.",
28     "8": "Login is unsuccessful. Please check if user/password/login URL \"\
29     provided is correct.",
30     "9": "Login URL is not in the same domain as target URL.",
31     "10": "Invalid HTTP request headers. Please check if the headers provided \"\
32     are correct.",
33     "256": "Crawler/fuzzer encountered error during scan. Please check \"\
34     logs/contact tech support.",

```

Click **Commit Changes**.

The configured asset scan is triggered and a scan request to FortiDAST is triggered for all future commits.

## Jira

FortiDAST offers seamless integration with Jira to streamline vulnerability management. This integration allows you to:

- Automatically log all vulnerabilities identified by FortiDAST scans as Jira bugs.
- Each vulnerability becomes a traceable issue, facilitating efficient tracking and remediation.
- Subsequent scans on the same target automatically update existing Jira bugs with new findings or mark resolved vulnerabilities as closed (if no longer detected).

To integrate a Jira server with FortiDAST, ensure that you configure the Jira server URL, email ID, and the API key for authentication in the [Jira Integration](#) page.

You can enable integration in [Jira Integration](#) page.

