

CSO Conference & Awards

Produced by **CSO** |  **IDC**

FioriDAST

Reinvent dynamic security scans for modern Web-based
Cloud applications

Matthias Ems, SAP SE
Vladislav Dexheimer, SAP SE

CSO Conference
& Awards



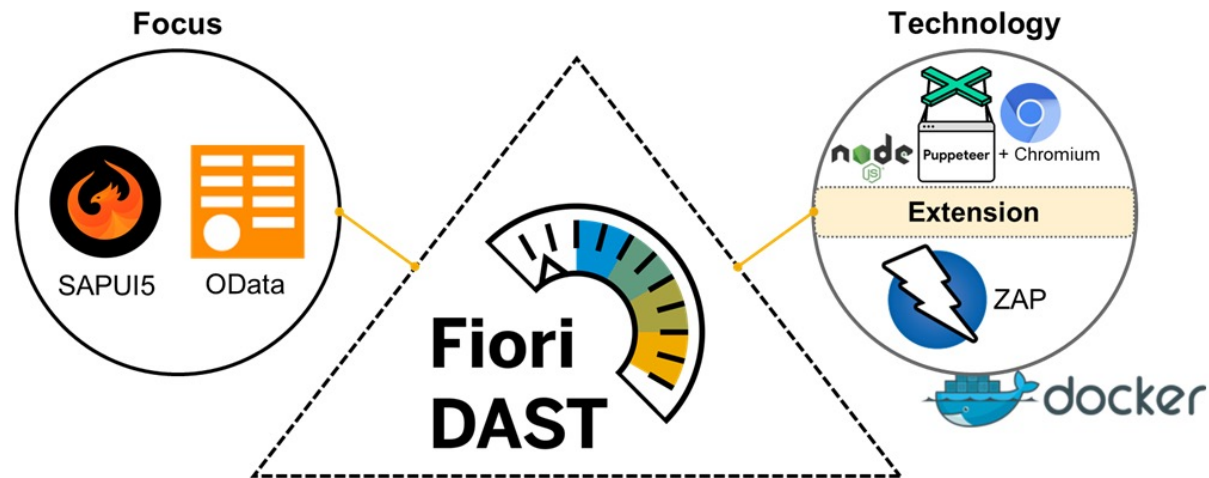
Produced by **CSO** |  **IDC**

Context & Challenges

- SAP delivers business software with many Web applications, especially within the SaaS ERP “SAP S/4HANA Cloud Public Edition”
- Code scans (SAST) cover only a subset of vulnerabilities and are blind to security issues like misconfigurations and authorization check inconsistencies
- Penetration testing comes late in the lifecycle and doesn't fully scale (→ shift left)
- Other dynamic security test (DAST) tools lack in-depth capabilities for SAP-specific UI and API technologies
- High-level automation and scalability is a must due more than 1000 applications in scope

Solution Overview

In-house security tool for **automated exploration and attacks** against **application interfaces at runtime** aiming to detect **Web-related vulnerabilities**



Capabilities

1. Configuration & Orchestration

Customizability via UI, API, CLI

...

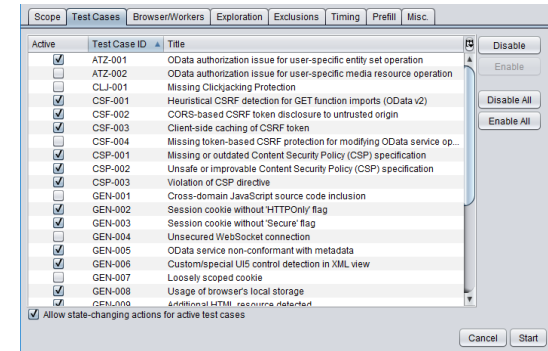
Command-line parameters

```
Scan depth level as a number between 0 and 5 which gets translated into more granular exploration settings or a string with a valid JSON object defining exploration strategy parameters directly
```

- Adapts the scan thoroughness but also influences the scan time during the exploration/testing
- Use higher levels only for small-sized apps and services
- Available exploration strategy parameters: `MaxActionPathDepth` (Integer), `MaxActionsPerControlGroup` (Integer), `MaxActionsPerActionPath` (Integer), `InspectValueHelpDialogs` (Boolean), `InspectOnlyDeltaChanges` (Boolean), `IgnoreRedundantControlActions` (Boolean), `TermPreferenceList` (String), `MaxUnsuccessfulValueStateRetries` (Integer), `OptimizeActionPaths` (Boolean)

```
-D (or --scandepth) <value>  
optional, since 0.0.1
```

Test case selection



Filtering and amendments of alerts

```
# Local browser storage usage will be ignored by external.server.com  
url(*https://external.server.com*)&test(GEN-008):confidence(f);  
# Missing Clickjacking protection will be considered as a high risk  
test(CLJ-001):risk(h);  
# Solution for CSRF-related alerts in path /services/v1/ will be adjusted  
url(*services/v1/*)&test(CSF):solution(Check if anti-CSRF token is applied);
```

Capabilities

1. Configuration & Orchestration

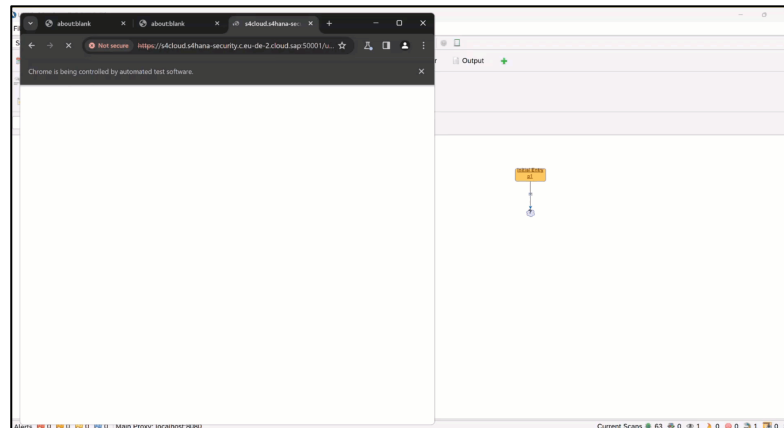
2. Exploration

Transparent
Authentication

**Web App
Crawling**

API Spidering

...



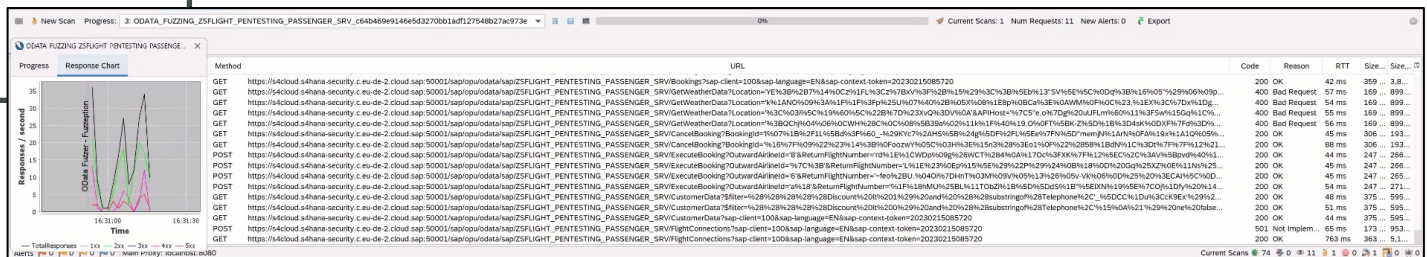
Capabilities

1. Configuration & Orchestration

2. Exploration

3. Security Testing

Client-side XSS
API Fuzzing
Insufficient Authorization Checks
Information Disclosure
...



CSO Conference
& Awards



Produced by



Capabilities

1. Configuration & Orchestration

2. Exploration

3. Security Testing

4. Reporting

Optimized Reporting

Different formats
(XML, SARIF,
PDF)

...

Fiorit DAST Scan Report

Created with Fiorit version 'Dev-DAST' and FioritDAST version '3.11.1' on Thu, 22 Feb 2018 07:43:19

Target Application Summary

Scan ID	LOBNDQMG
Scan	
Target URL	https://dev1.fiori-goat-dev.c.eu-de-1.cloud.sap:44300/#FioriGoat-open
App ID	sap-security-pentesting-fiori-goat
Title	FioriGoat
Description	A Vulnerable Application for Pen Testing
Version	1.0.1
Application Base URL	https://dev1.fiori-goat-dev.c.eu-de-1.cloud.sap:44300/#FioriGoat-open
Users	P1 P2
Services	ZFLIGHT_PENTESTING_PASSENGER_SRV ZFLIGHT_PENTESTING_SRV
Applied Test Cases	

Summary of Alerts

Risk Level	Number of Alerts
High	2
Medium	5
Low	6
Informational	2

High Risk (High Confidence) | Reflected Client XSS

Description: Found a reflected client XSS vulnerability inside route '___bypassed_'. Control class 'sap.security.pentesting.fiori-goat.control.ErrorText' wrote unsanitized output to the DOM (context: ELEMENT_CONTENT).

URL: <https://dev1.fiori-goat-dev.c.eu-de-1.cloud.sap:44300/#?sap-client=001&sap-language=EN#FioriGoat-open>

Method: GET

Parameter: The mandatory route parameter 'userName' can be used to inject and execute JS code

Attack: Attack payload '<script-prf(1)<script>' used within the route value 'personaldata<script-prf(1)<script>'

Generic ID: 248c1f70

Specific ID: 764f0d4a

Check snapshot

Achievements & Takeaways

- Automated scanning of > 600 applications in SAP S/4HANA Cloud daily with well-defined audit and correction processes (customer trust in SAP security)
- Broad tool adoption within other product areas at SAP
- Tool improvements via feedback loop from different channels

- Certain business challenges can only be tackled by custom/specialized solutions
- High efforts and engineering resources are required but the benefits can be paramount with automation in mind

Credits

- Management support
- International core team
- Cybersecurity students
- Close network of SAP security experts
- (Early) Adopters