Event

Security Meetup:

Catching Vulnerabilities & AI-Driven DevSecOps

Google Developer Group
Stuttgart

22 January 2026

CGI

22.01.2026

Agenda

1. Introduction: Testing Pyramid & DevSecOps Tooling
2. Why E2E Tests are not enough
3. Introducing DAST in the SDLC
4. ZAP - *One Proxy to attack them all*
5. CI/CD Integration with GitHub Actions + **Demo**
6. Bonus: Pipeline Hardening

© 2026 CGI Inc.

Introduction

Testing Pyramid & DevSecOps Tooling

© 2026 CGI Inc.

DevSecOps Tooling

Burp Suite

ZAP

DEPENDENCY-CHECK

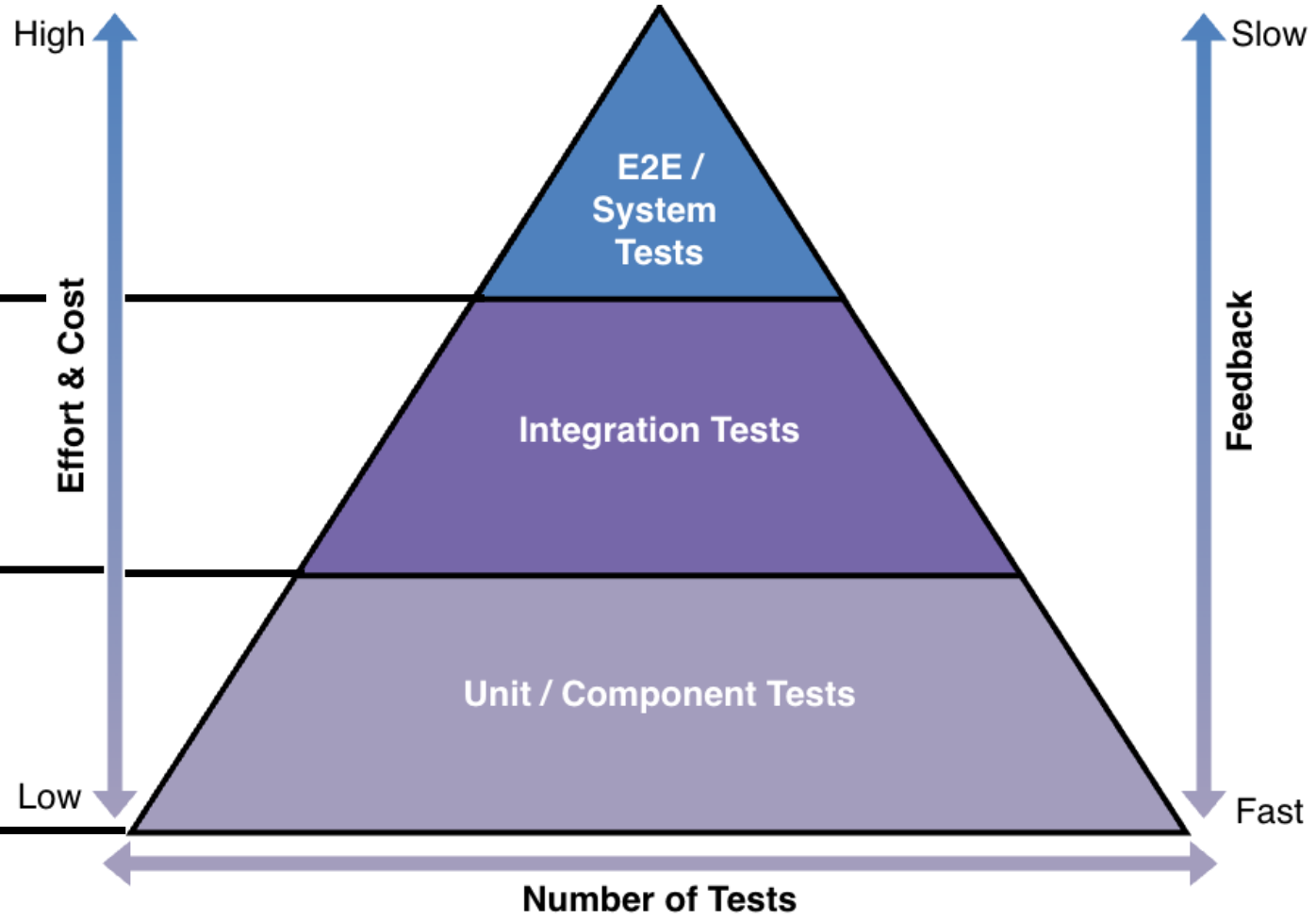
trivy

SpotBugs

sonarqube

Semgrep

Testing Pyramid

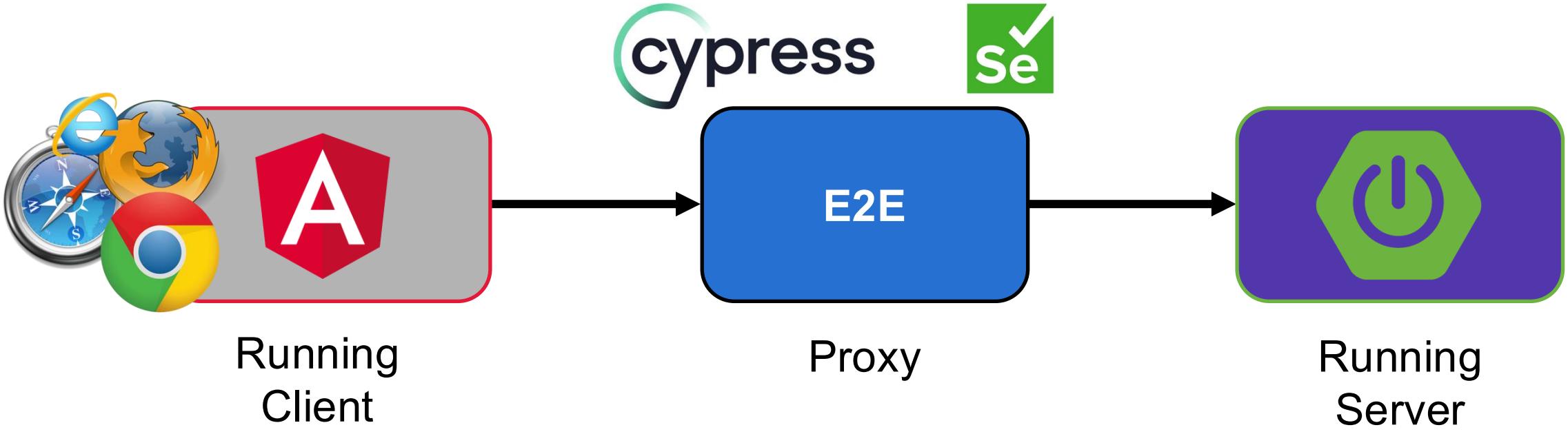


© 2026 CGI Inc.

Why E2E Tests are not enough

© 2026 CGI Inc.

Dynamic Application Security Testing (DAST)



© 2026 CGI Inc.

E2E Tests

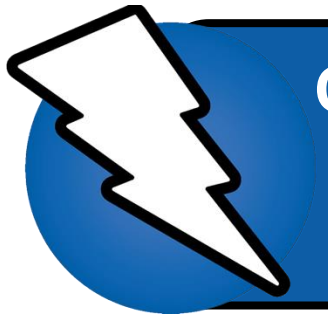
Does

- ✓ validate real user journeys
- ✓ verify system integration
- ✓ test expected behaviour
- ✓ use known inputs

Can a user sign up / login, and complete a purchase?

Doesn‘t do

- fuzzing of inputs
- enumerate endpoints
- abuse auth logic
- chain attacks



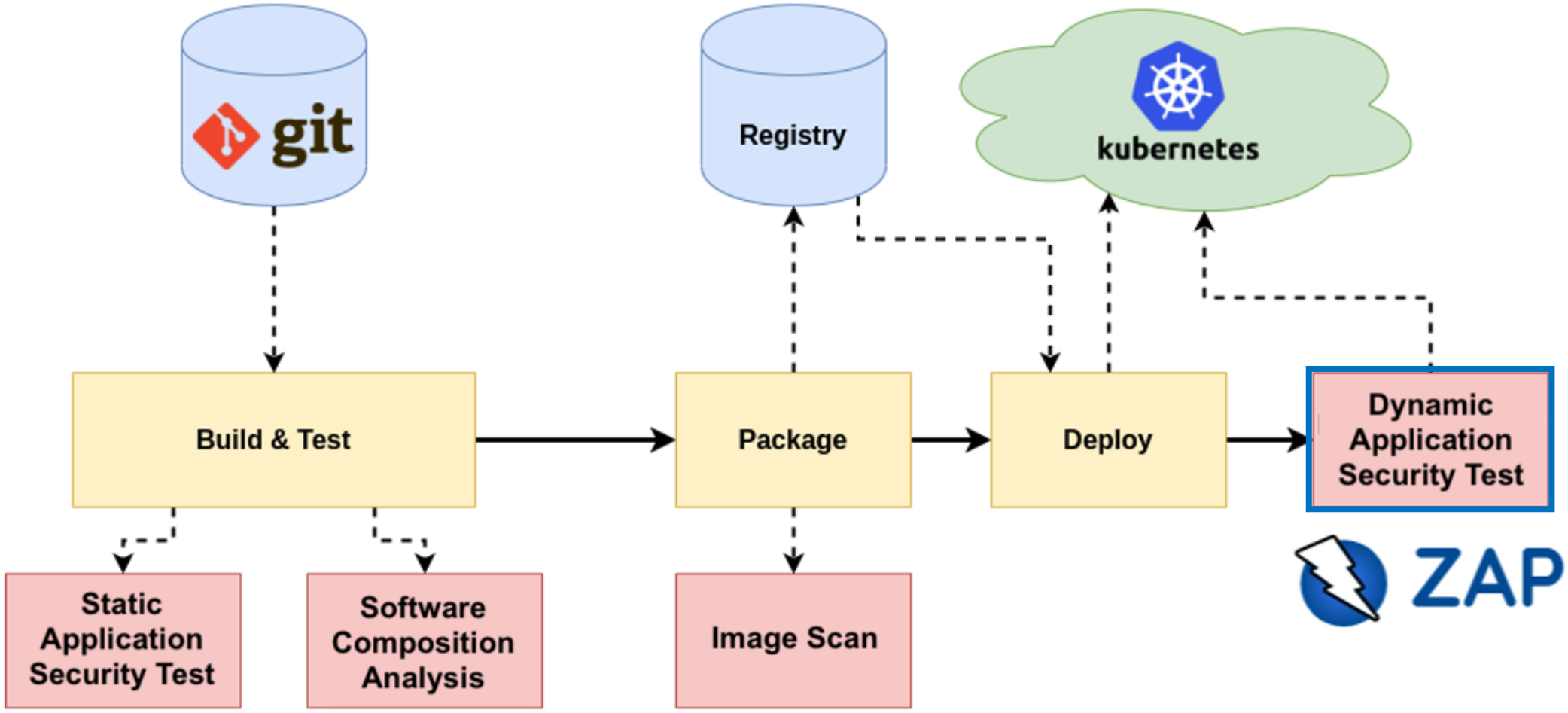
Can an attacker use an user flow to manipulate the checkout process and gain unintended benefits?

© 2026 CGI Inc.

Introducing DAST in the SDLC

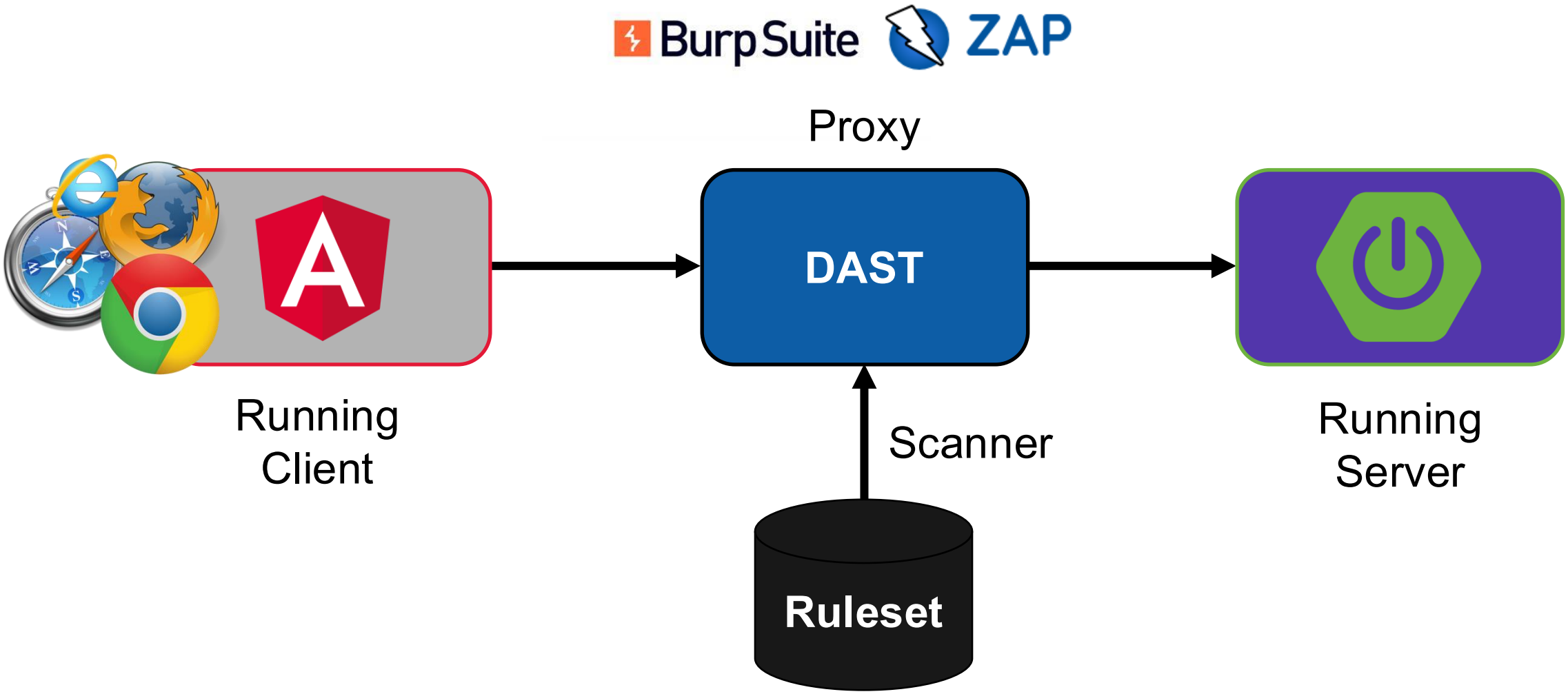
© 2026 CGI Inc.

ZAP in CI-/CD-Pipeline



© 2026 CGI Inc.

Dynamic Application Security Testing (DAST)



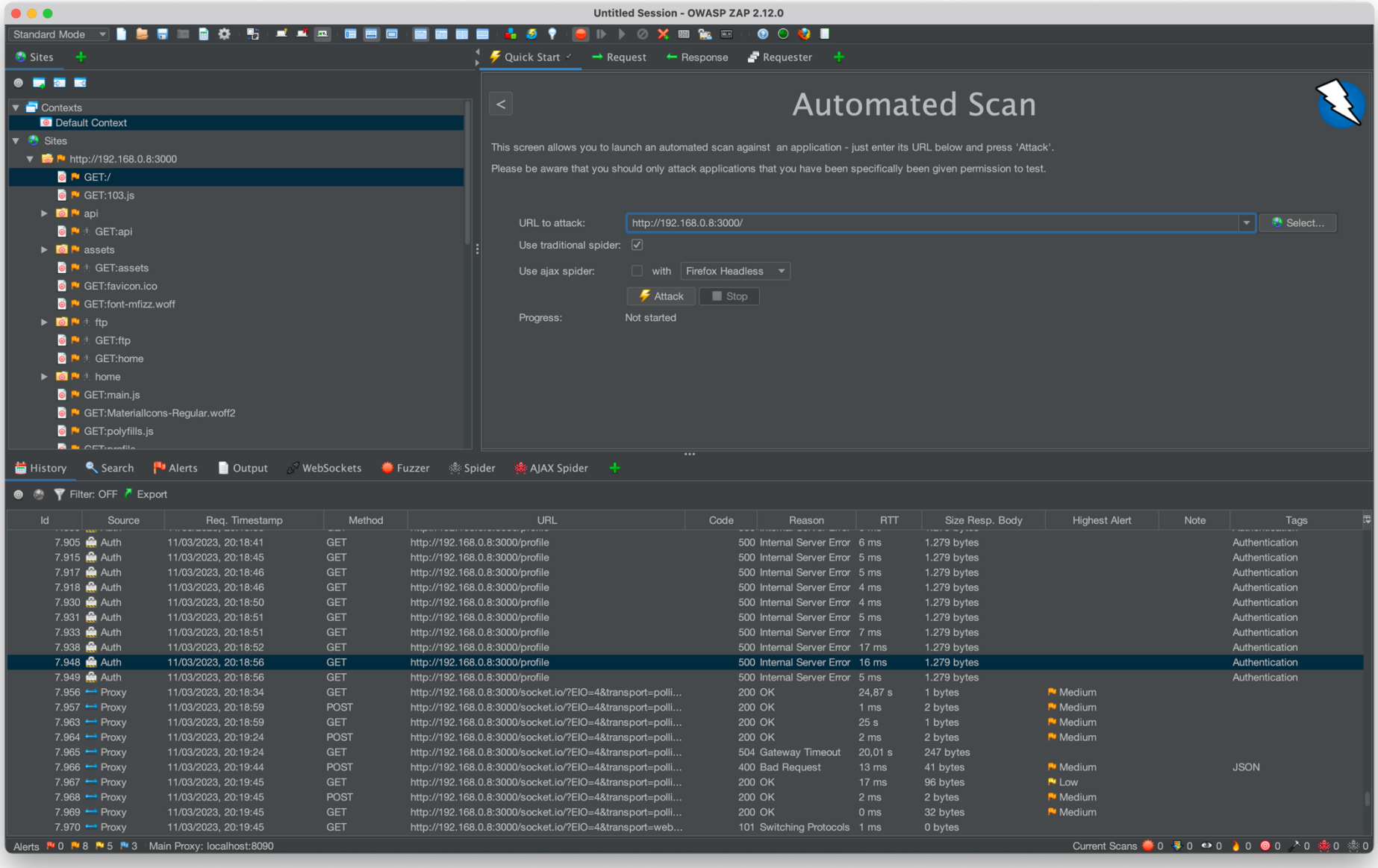
© 2026 CGI Inc.

ZAP

- One Proxy to attack them all

© 2026 CGI Inc.

Zed Attack Proxy (ZAP)



Passive & Active Scans
Script Integration
Automation
Reporting
Spidering
Fuzzing
...

Passive Scan

Focus

- Analyzes the given page/domain
- Identifies potential vulnerabilities
- Quick scan for:
  - Vulnerable HTML tags (Sinks)
  - Missing headers
  - Misconfiguration (Cookie, CSP)

© 2026 CGI Inc.

Active Scan

Focus

- Uses crawlers and web fuzzers
- Verifies identified vulnerabilities
- JS analysis for further vulnerabilities
- Fuzzes request bodies and parameters for:
  - Various injection attacks
    - XSS, SQLi, XXE, …
  - Other common web vulnerabilities.

© 2026 CGI Inc.

Report Generation

Contains

- Summary of all findings
- Finding details
  - Description
  - URL metadata
    - (HTTP method, param, etc.)
  - References (rfc)
  - CWE ID
  - Plugin ID

https://www.zaproxy.org/docs/desktop/addons/report-generation/

ZAP by Checkmarx

Site: https://saftladen.automatisier.bar

Generated on Tue, 25 Nov 2025 15:53:13

ZAP Version: D-2025-11-24

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	3
Low	4
Informational	5
False Positives:	0

Summary of Sequences

For each step: result (Pass/Fail) - risk (of highest alert(s) for the step, if any).

Alerts

Name	Risk Level	Number of Instances
SQL Injection	High	2
Content Security Policy (CSP) Header Not Set	Medium	1
Cross-Domain Misconfiguration	Medium	Systemic
Integer Overflow Error	Medium	1
A Server Error response code was returned by the server	Low	16

© 2026 CGI Inc.

CI/CD Integration
with GitHub Actions

© 2026 CGI Inc.

ZAP Baseline Scan *(Passive Scan)*

- Focuses on identifying potential vulnerabilities
- Designed to detect known security issues:
  - XSS Protection (Header)
  - SQL Injections
  - Path Traversal
  - HTTP Security Headers
  - Unvalidated Redirects and Forwards
  - Insecure Deserialization
  - …

```yaml
zap_baseline_scan:
  name: "ZAP Baseline Job"

  runs-on: ubuntu-latest

  steps:

    - name: "ZAP Baseline Scan"
      uses: zaproxy/action-baseline@v0.9.0
      with:
        token: ${{ secrets.GITHUB_TOKEN }}
        docker_name: 'ghcr.io/zaproxy/zaproxy:stable'
        target: 'https://example.com'
        rules_file_name: '.zap/rules.tsv'
        artifact_name: zap_baseline_scan
        cmd_options: '-a'
```

https://www.zaproxy.org/docs/docker/baseline-scan/

© 2026 CGI Inc.

ZAP API *& Full Scan (Active Scans)*

- Focuses on scanning RESTful APIs
- Supported formats:
  - Openapi (DEFAULT)
  - Soap
  - GraphQL
- Searches for a variety of vulnerabilities:
  - SQL Injections
  - Authentication issues
  - Insecure direct object references (IDOR)
  - Broken Access Control
  - Sensitive Operartions Without Confirmation
  - …

```yaml
zap_api_scan:
  name: "ZAP API Job"

  runs-on: ubuntu-latest

  steps:

    - name: "ZAP API Scan"
      uses: zaproxy/action-api-scan@v0.5.0
      with:
          token: ${{ secrets.GITHUB_TOKEN }}
          docker_name: 'ghcr.io/zaproxy/zaproxy:stable'
          format: openapi
          target: 'https://example.com'
          rules_file_name: '.zap/rules.tsv'
          artifact_name: zap_api_scan
          cmd_options: '-a'
```

https://www.zaproxy.org/docs/docker/api-scan/
https://www.zaproxy.org/docs/docker/full-scan/

© 2026 CGI Inc.

With Docker

Different scans types:

- `zap-full-scan.py`
- `zap-baseline.py`
- `zap-api-scan.py`

Options:

`-t` target domains / APIs
`-f` file format
`-r` report file
`-g` config for the context
`-z` set options (e.g. Auth header)

https://www.zaproxy.org/docs/docker/about/
https://deepwiki.com/zaproxy/zaproxy/5.1-scan-rules

© 2026 CGI Inc.

```yaml
zap_docker_api_scan:
    name: "ZAP API Job (Dockerized)"

    runs-on: [ xentry, medium ]

    steps:

      - uses: actions/checkout@v2

      - name: "Pull ZAP Image"
        run:
          docker pull zaproxy/zap-weekly


      - name: "Run ZAP API Scan (json)"
        run: |
          set +e
          docker run -v $(pwd):/zap/wrk/:rw -t zaproxy/zap-weekly \
            zap-api-scan.py -t OpenAPI.json -f openapi \
            -r zap_report.html -g api-active-scan.conf -z options.prop
          echo "ZAP completed with exit code $?"
          set -e

      - name: Upload ZAP Report
        uses: actions/upload-artifact@v3
        with:
          name: zap-report
          path: zap_report.html
```

Demo

https://github.com/marcel-haag/dev-sec-ops-demos

© 2026 CGI Inc.

Bonus:
Pipeline Hardening

ZAP Baseline Job summary ...

Network Activity Monitored by StepSecurity Harden-Runner

Network calls made by the runner during this job run. These were automatically monitored and logged in real-time by StepSecurity Harden-Runner.

Process	Destination	Port	Status	Timestamp
java	saftladen.automatisier.bar	443	✅ Allowed	Nov 27 2025 11:27:15
java	raw.githubusercontent.com	443	✅ Allowed	Nov 27 2025 11:27:14
java	news.zaproxy.org	443	✅ Allowed	Nov 27 2025 11:27:13
java	cfu.zaproxy.org	443	✅ Allowed	Nov 27 2025 11:27:03
dockerd	production.cloudflare.docker.com	443	✅ Allowed	Nov 27 2025 11:26:32

Pipeline Hardening

Recommended Configuration to Harden the Runner

☑ Block egress traffic: Only allow calls to allowed endpoints

Initial baseline created. It will be updated as more job runs are monitored.

```yaml
- name: Harden Runner
  uses: step-security/harden-runner@95d9a5deda9de15063e7595e9719c11c38c90ae2 # v2.13.2
  with:
    egress-policy: block
    allowed-endpoints: >
      auth.docker.io:443
      cfu.zaproxy.org:443
      github.com:443
      news.zaproxy.org:443
      production.cloudflare.docker.com:443
      raw.githubusercontent.com:443
      registry-1.docker.io:443
      saftladen.automatisier.bar:443
      tel.zaproxy.org:443
```

https://docs.stepsecurity.io/harden-runner

Thank you for listening!

Questions?

Marcel Haag

marcel.haag@cgi.com

© 2026 CGI Inc.