WebInspect Automation Workflows

WebInspect automation workflows use build automation tools to manage the dynamic scanning ecosystem, including QA testing and cloud deployments.

Guide

www.microfocus.com

Guide

Security

MICRO FOCUS

Automation Allows Tests to Be Run Simultaneously and at Scale

One of the goals of Securing DevOps (DevSecOps) is to build security testing into the development process. Integrating automated security testing makes it simpler for developers, QA staff, and security teams to work in sync across silos. Security testing can be part of the testing stack with similar frequency and integration as unit, integration, compatibility, and performance testing.

Dynamic analysis (DAST), combined with static analysis (SAST), provides more thorough coverage, but automating dynamic is more complex. You can either build your own tech stack, or borrow a framework. This guide helps you accelerate your automation by using existing test automation scripts/frameworks that other enterprises have already created as part of their DevOps practices.

Automating WebInspect into existing DevOps systems and processes allows security tests to be run simultaneously and at scale.

Maven Plugin for WebInspect and WebInspect Enterprise on GitHub

1) Instantiate a WebInspect proxy, 2) route the traffic from integration tests, 3) save the proxy traffic as a workflow macro (and shut down the proxy), 4) configure a new scan, and 5) run the scan.

- **https://github.com/rsenden/fortify-integration-maven-webinspect/tree/webinspect-maven-plugin-2.1**

Disclaimer: This information is provided as part of a community effort to share approaches to automation. The information is provided as a guidance and is not an endorsement for any particular solution. There may be no Fortify QA and Support for content within this page.

WebInspect Automation—General Workflow

Automation workflows use a build automation tool that manages the scanning ecosystem via the following steps:

A. Security team sets up the security scanning steps as a "security task" that is called after a build and after app deployment, via the build automation tool.

B. Development teams submit code changes to the build automation tool and after the set operational period, the security task is triggered after the build and app deployment is complete.

C. On completion of the security task, the automation tool is set up to either pass or fail the build job based on the security risk defined by the security team.

D. The security vulnerability findings are captured in Fortify Software Security Center (SSC), from where they can be optionally moved to bug repositories via the integrations available on SSC.

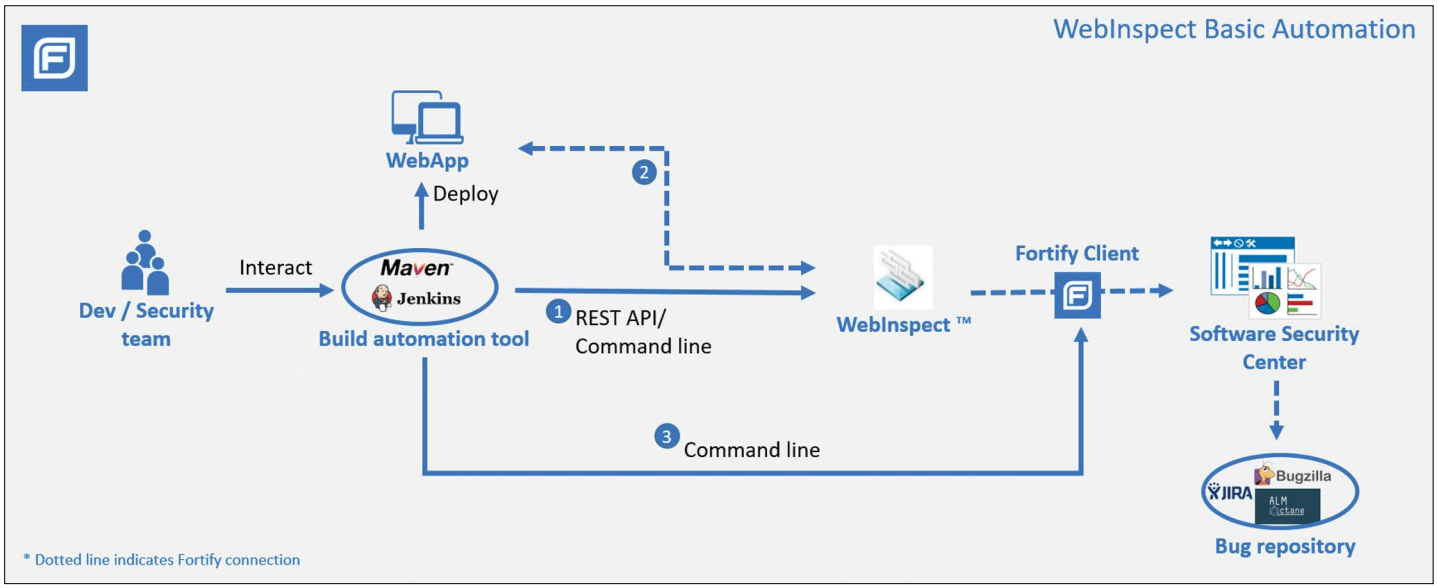

Figure 1. Basic Security Task—WebInspect

Basic Security Task—WebInspect

1. Health check the WebInspect sensor to ensure the scanner is available to schedule a scan.
2. Call the WebInspect REST API/ or command line to initiate a scan. This involves passing the necessary URL, settings file and login information.
3. Polling the sensor to check the status of scan and trigger the next steps on scan completion.
4. On scan completion, export findings as an FPR to a server containing Fortify Client and upload to SSC via the Fortify Client.

WebBreaker

Target solves Dynamic Application Security Test Orchestration (DASTO) with WebBreaker tool on GitHub. This open-source project utilizes WebInspect to provide greater agility and flexibility to deliver improved integration into the SDLC pipeline, Git workflows, etc.

- **https://github.com/target/webbreaker**

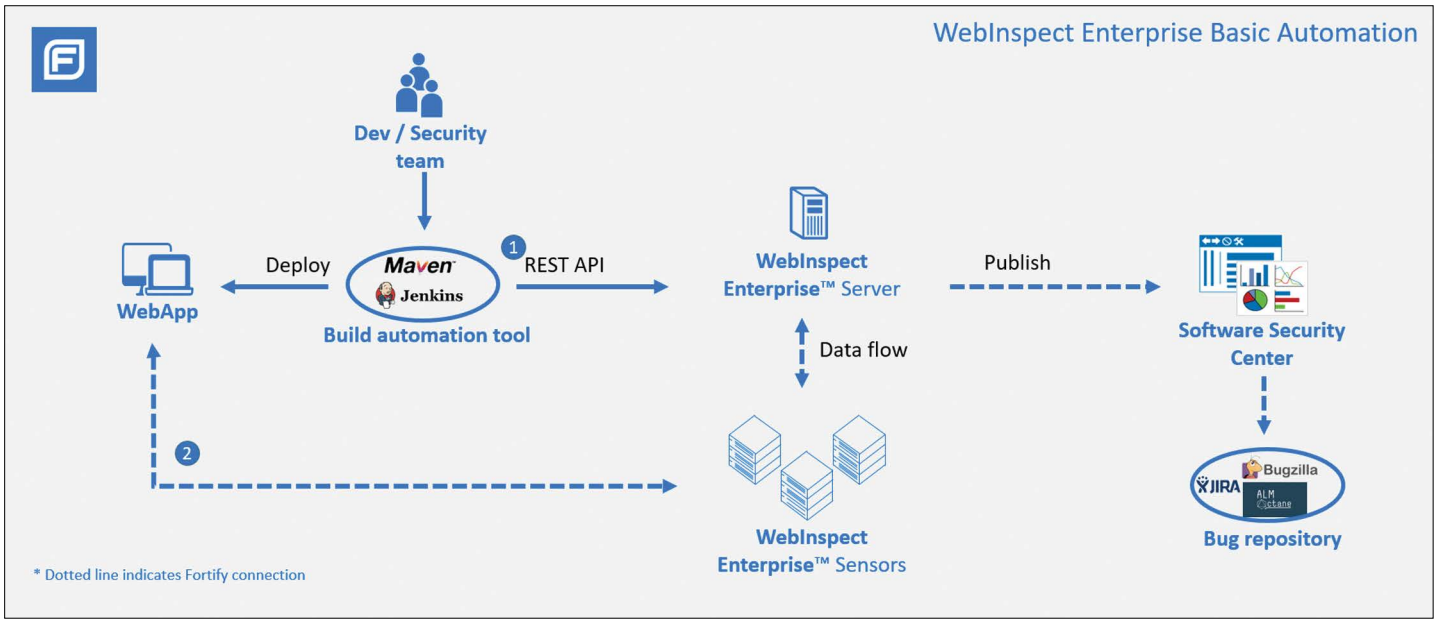

Figure 2. Basic Security Task—WebInspect Enterprise

Basic Security Task—WebInspect Enterprise (WIE)

This is simpler because WIE manages the scheduling and polling to identify availability of a sensor. WebInspect Enterprise also automatically publishes results to Fortify Software Security Center.

- Call the WebInspect Enterprise server API to schedule a scan with URL and settings file/template information.

Proxy and QA Automation

Automation can utilize artifacts generated during QA functional tests (for example Selenium scripts to automate WI/WIE scans). The advantage of this approach is:

A. The functional testing often involves a sequence of actions that have a business logic associated with them, whereas it is impossible to model from a blind WebInspect automatic crawl.

B. The possibility to utilize the login sequence used during the functional testing and not create a separate WebInspect Login Macro. This involves configuring settings to exclude the login page from WI crawl or audit, and also that a logout doesn't occur during security scan.

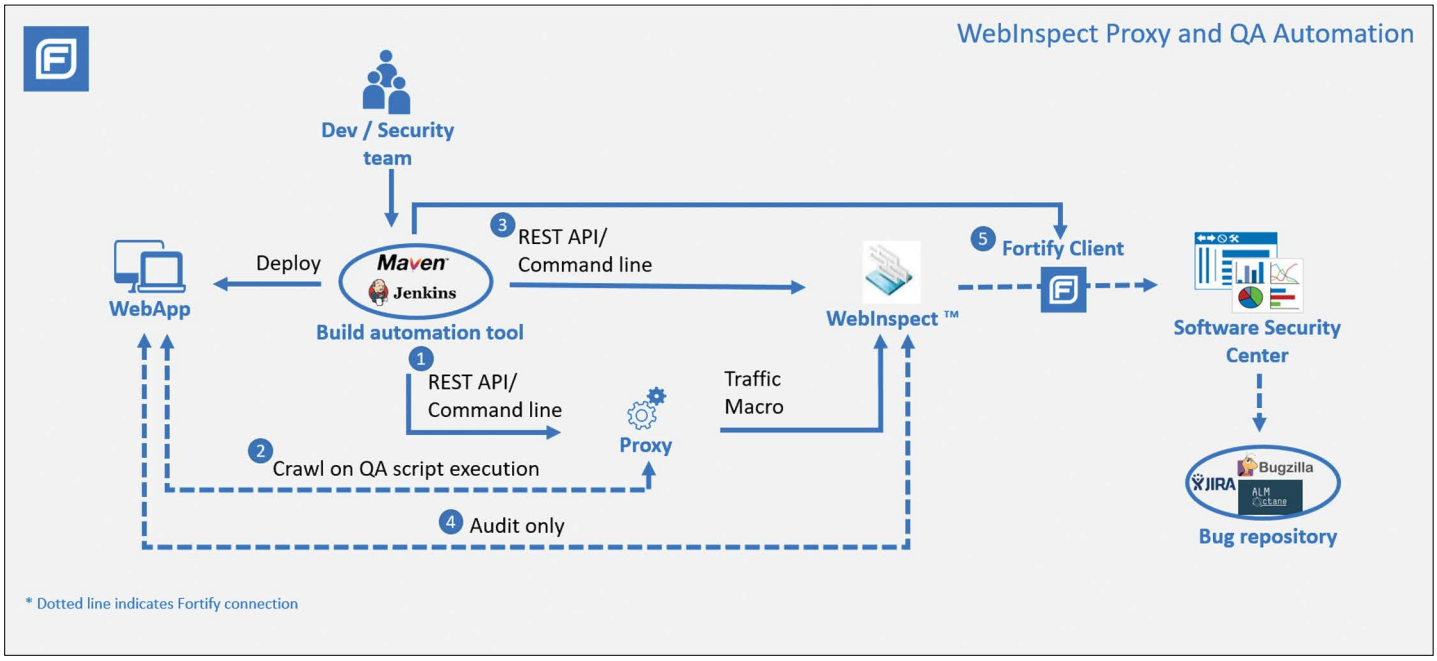

Figure 3. QA Security Task—WebInspect

QA Security Task—WebInspect

Add these steps to the Basic Security Task—WebInspect:

1. Spinning up a WI proxy via REST API and replaying the captured QA artifact to generate a traffic file. The traffic file is then saved as a WebMacro.
2. Using Command line/ REST API to modify default settings file. The settings file is overridden] a Workflow macro saved from the traffic file in step 1.

Useful Links for Automation

1. **FoD BugTrackerUtility. Fully automated command-line utility for batch submission of SSC and FoD vulnerabilities to various external systems.**
   - **https://github.com/fod-dev/FoDBugTrackerUtility/tree/processrunner-bugtracker-root-3.2**
2. WebInspect APIs
   - **http://<hostname:port>/webinspect/swagger/ui/index#/**
3. Maven repository for Fortify-related artifacts
   - **https://github.com/rsenden/FortifyMavenRepo**

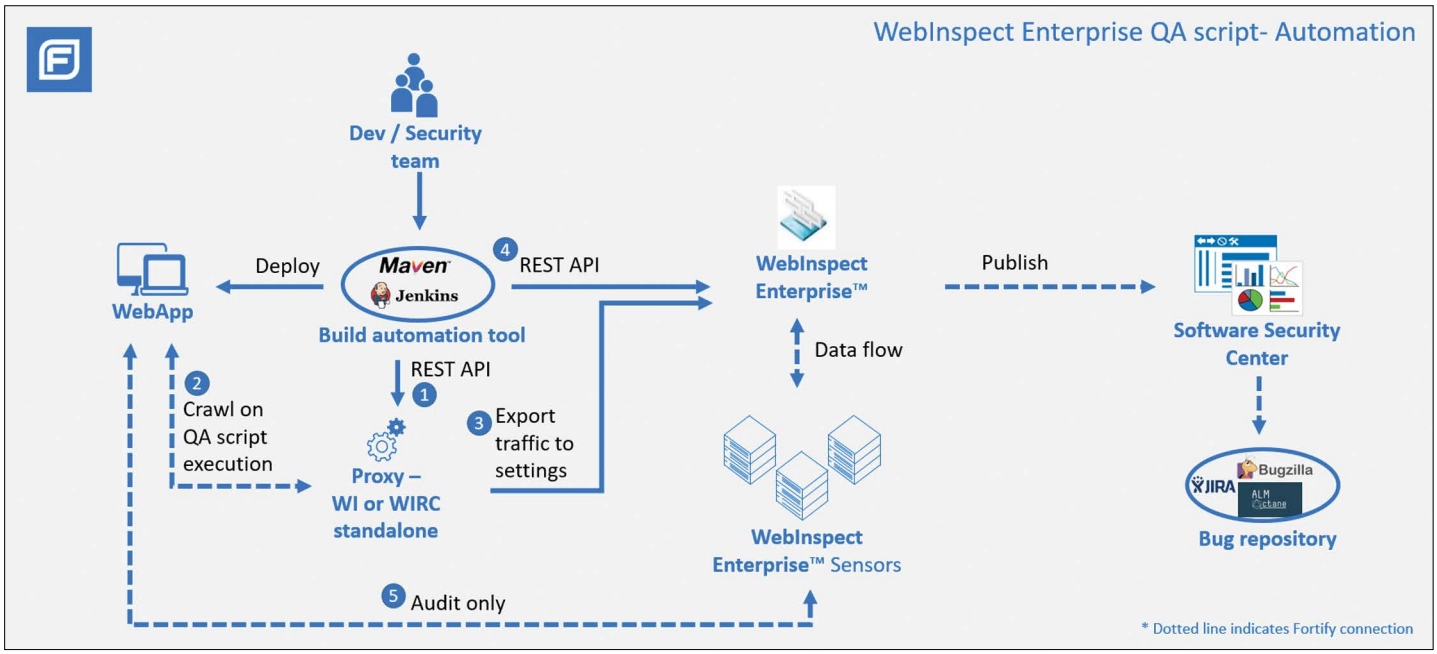

Figure 4. QA Security Task—WebInspect Enterprise

QA Security Task—WebInspect Enterprise

1. Same additional steps as for WI. For customers who don't have access to WI desktop to spin up a proxy, download a license-free instance of a proxy available at the Fortify Marketplace.
2. After creating a settings file, the process of initiating a scan for WIE involves additional steps found in the Creating Scan Guide.

Automation in the Cloud

Another use case is automation in the cloud by deploying the sensors for both WI and WIE, and dynamically scaling the sensor installation around the scale of application security testing under process.

1. Security team accesses the scan request pipeline and determines scaling/descaling of N Sensors. Assign licenses based on this requirement.
2. Security teams use the general workflow described in the general workflow and then loop through steps 1 and 2.

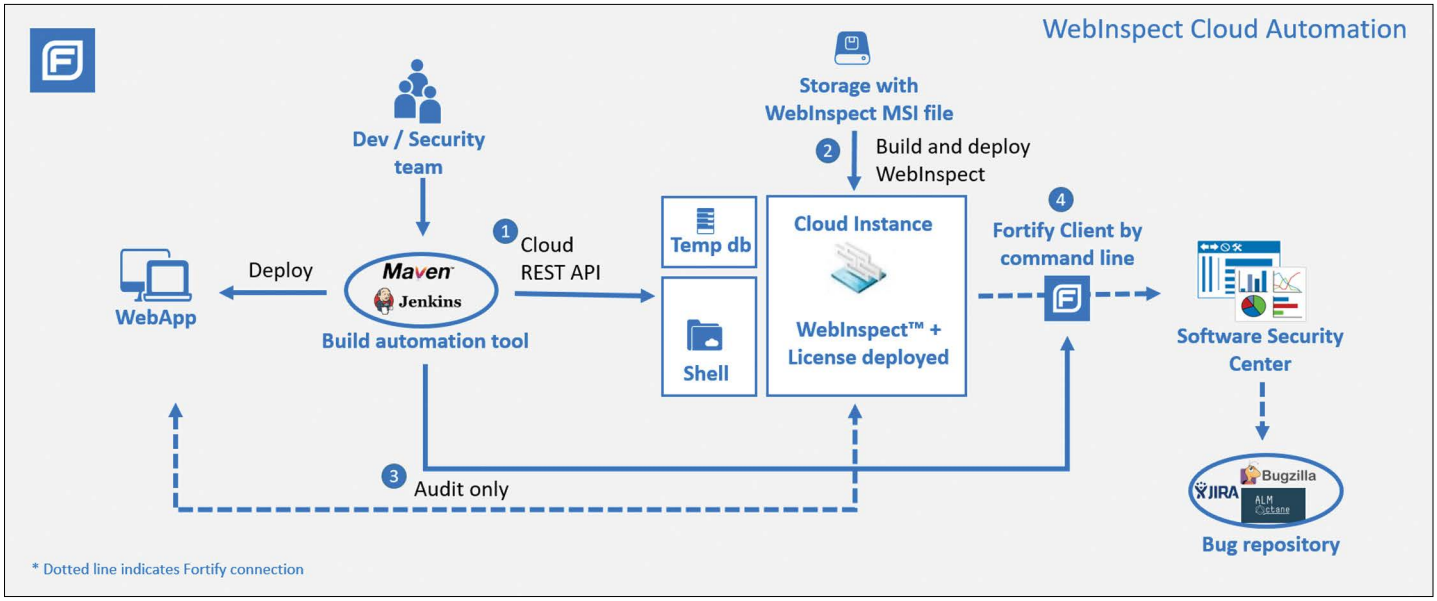

Figure 5. Cloud Security task—Scaling for WebInspect sensors

Cloud Security Task—Scaling for WebInspect Sensors

1. A WebInspect installation MSI is stored in cloud storage and ready for deployment. [call location: cloud_memory]
2. Security team calls the cloud API to create a windows instance and uses the command line of the instance (C_Instance) to do a headless installation of WebInspect sensor from location: cloud_memory.
3. Necessary settings and macro files are deployed over the instance
4. A scan is triggered in the command line (C_Instance) using the REST APIs of WebInspect in that instance.
5. On scan completion, export findings as an FPR to a server containing Fortify Client and upload to SSC via the Fortify Client.

Learn more about dynamic application security testing software.

Contact us at:
www.microfocus.com

Like what you read? Share it.

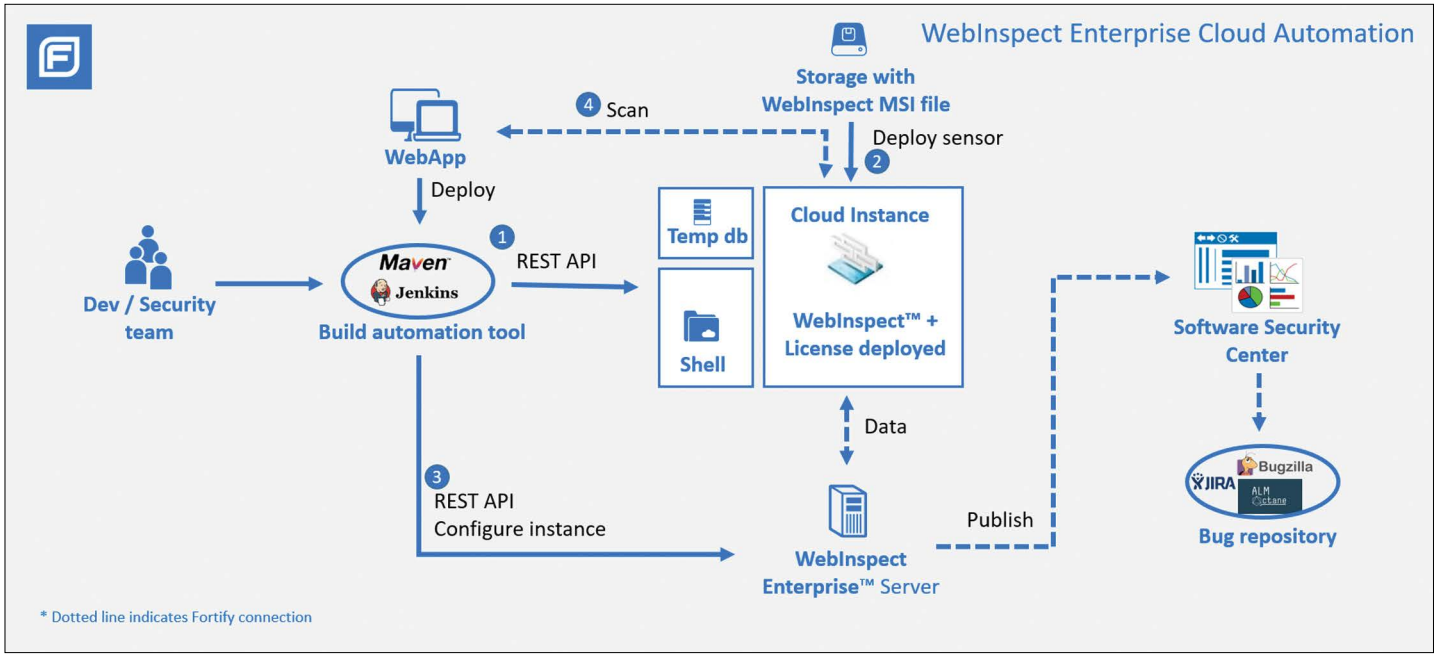

Figure 6. Cloud Secuirty Task—Scaling for WebInspect Enterprise Sensors

Cloud Security Task—Scaling for WebInspect Enterprise Sensors

Additional steps are required to connect and configure the sensor to connect to WIE Server management layer and assign necessary permissions for access to the sensor.

1. A WebInspect installation MSI is stored in cloud storage and ready for deployment.
2. Security team calls the cloud API to create a windows instance and uses the command line of the instance (C_Instance) to do a headless installation of WIE sensor from location
3. Using the command line : C_Instance, the sensor is configured to connect to WIE management server. Invoke the REST APIs of the WIE server management layer to provide permissions and security group access to the WIE sensor. Watch the WebInspect Enterprise Automation demo on the Fortify Marketplace.
4. Once the WIE sensor installation is complete, call the WIE server API to schedule a scan with URL and settings file/template information.
5. On scan completion, findings are automatically synched to SSC.