SAND2019-5094C

Automating Burp Suite Application Security Scanning

NLIT Summit, May 2019

PRESENTED BY

Roy Life – ralife@sandia.gov

Gail Berry - glberry@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SANDS No:

Agenda

- Who Are We?
- Problem Statement
- Overview of Burp Suite
- Project Overview
- Architecture Flow Diagram
- Development Environment and Burp API
- Lessons Learned
- Future Work
- Q&A: Closing Discussion

Who are we?

Roy Life and Gail Berry

- Role
- Background
- Experience

Problem Statement

- Improve software quality and security
- Lower cost and time
- Utilize CI/CD pipeline and automation
- Improve the efficiency of deployments
- Strategic objective within our Center (org 9350) to leverage DevOps and automated testing
- Burp as a Service
  - Requires specialized knowledge of Burp

(or)

  - Requires application knowledge

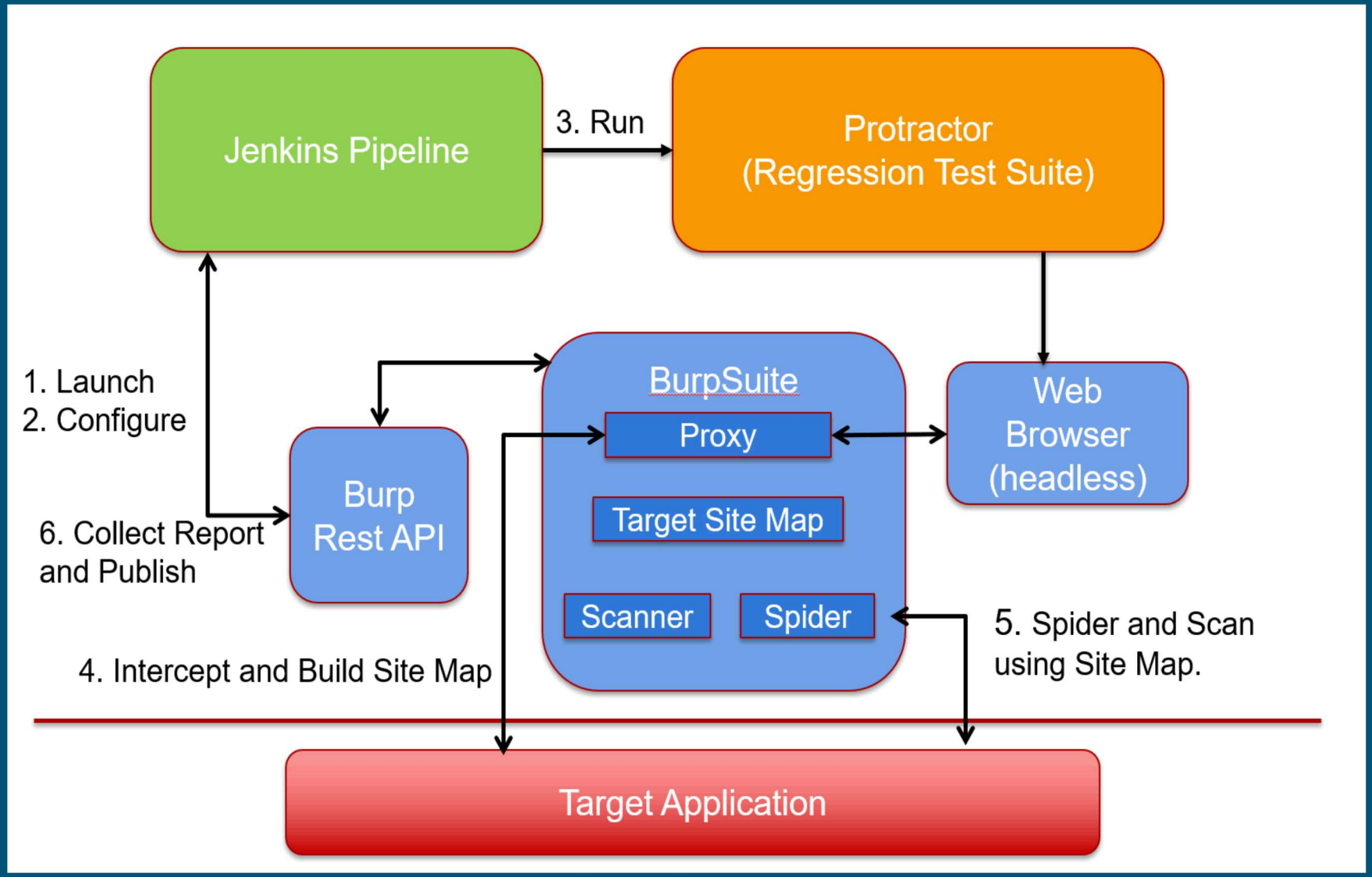
Overview of Burp Suite

- Developed by PortSwigger
- DAST (Dynamic Application Security Testing) / Penetration Testing Tool
- Integrated platform for performing security testing of web apps
  - Developed to provide comprehensive solution for web app security checks. Functionality includes proxy server, scanner, intruder, spidering, a repeater, a decoder, comparer, an extender, and sequencer
- 3 Different Products
  - *Community edition*
    - Free version – very limited options of the tool available, manual pen testing focus (scanner not available)
  - *Professional edition*
    - Paid Version – contains 'Scanner' tool for performing automated vulnerability scans of web applications as well as full functionality of advanced options (see above)
  - *Enterprise edition*
    - Designed for automated scanning at scale, and integration with software development processes

Project Overview

- Desired an automated approach to running Burp Suite security scanning
  - Manual security scanning is time consuming (bottleneck) due to application discovery/crawling process
  - Limited resources – Not many individuals with Burp knowledge
  - Limit knowledge of the application being tested
- Used Automated UI tests for application discovery
  - Used protractor (selenium) UI automated tests to act as the event based trigger for application discovery while using Burp Proxy to intercept all Requests/Responses
  - Used Jenkins (CI/CD) to run the automated UI tests
- Burp Suite Professional (v.1.7.34) with 'Burp-REST-API'
  - Used 'Bupr-REST-API' to develop Burp Configuration scripts using javascript
  - Used Jenkins (CI/CD) to run configuration scripts
  - Automated the spidering, proxy, intercepts request/response from UI tests, then kicks off security scan, then generates Burp security scan report

Architecture Flow Diagram



Development Environment and Burp API

- Development Environment
  - Angular/NodeJS Web application
  - Protractor/Jasmine – Automated UI tool
  - Jenkins CI/CD pipeline
- Burp Suite External API
  - https://github.com/vmware/burp-rest-api
  - Version 1.0.3

Lessons Learned

- Burp isn't infallible
  - Need multiple security tools to provide comprehensive security posture
  - Using multiple approach to testing applications
    - Manual Penetration Testing
    - DAST (Web App Scanning Tools)
      - Burp
      - Accunetix
      - Netsparker
    - SAST (DevSecOps Tools)
      - Checkmarx
      - Fortify
      - SonarQube
- Stable development environment
  - Testing in a quality environment is key
- Control over Jenkins Server
  - Needed admin access for debugging and troubleshooting

Future Work

- Burp Profession v2.0 (Beta) API Integration
  - Interested in using Burp API in Beta to determine functionality and performance vs external API currently used
- Burp Enterprise API Integration
  - Burp Enterprise comes with CI/CD integration
  - Scalable security scanning using distributed agents
  - Interested in using Burp API in Enterprise to determine functionality and performance vs external API currently used
- Compare Burp Pro (v.1.3.37) scan results with Burp Enterprise
  - Compare security scan results for current environment vs. security scan results from Enterprise version

Closing Discussion, Q&A

Discussion

- What lessons can you share?
- What are your issues?

Questions?

Contact Us

- Roy Life (ralife@sandia.gov)
- Gail Berry (glberry@sandia.gov)