DefectDojo, vidljivost ranjivosti na jednom mjestu

Dubravko Sever

14 studenoga 2020

1. Introduction
2. Problem
3. DefectDojo
4. Installation
5. Structure
6. Demo: manual import
7. Automation

8. Demo: automated import

9. Development

10. Question?

Who am I?

My past

- 12 University Computing Center (SRCE),
- 2 in integration industry, automation,
- since 2 years ago Pan-Net Deutsche telekom (Senior Security Spec)
- contractor for gew companies

What do I do

- Security in and of cloud solutions
- Cloud architecture, with security in focus
- Security in microservice architecture
- Compliance and standardization (CSA, ISO, PSA...)
- ...

Section 1

Introduction

Basics

Vulnerability assessment

SANS

```
Vulnerabilities are the gateways by which threats are manifested
```

RAPID7

```
A threat refers to the hypothetical event wherein
an attacker uses the vulnerability.
```

Vulnerability assessment: to determinate does vulnerability exists and can harm to one of the CIA principles
System that is 100% safe doesn't exist, some vulnerabilities are always there
Conclusion: if there are not vulnerabilities, than system doesn't exist

Basics

Vulnerability assessment

- One cycle of addressing and processing of vulnerabilities
- Processing not only to find out vulnerabilities, includes followup by doing prioritization, propose way of handling

Classification (MITRE):

- CVE-Common Vulnerability Exposure (CVSS v2 3 classes,v3 5 classes)
- CWE-Common Waekness Enumeration, groups (CWE-233: Improper Handling of Parameters)
- CAPEC-Common Attack Pattern Enumeration and Classification (CAPEC-66: SQL Injection)
- CPE - Common Platform Enumeration (cpe:2.3 kubernetes:kubernetes:1.14.0:-::::-::*)

CVE Search

Basics

Vulnerability management

What is the difference between assessment and management?

- Is never ending process,
- well defined holistic approach, build in into company processes

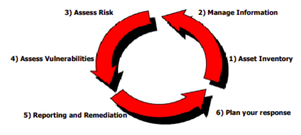


Figure 1: SANS, Vulnerability Management:Tools, Challenges and Best Practices

Basics

Discovery of vulnerabilities

Speed up the task, but automate it:

- SAST (Static Application Security Testing), white box, SonarQube, semgrep, KubeSec, DependencyChecker, ThunderScan, Snyk
- DAST (Dynamic Application Security Testing), black box, ZAP, Burp
- IAST (Interactive Application Security Testing), runtime testing
- Vulnerability Network Scanners, Nexpose, Nessus, OpenVas...
- Container Scanners, Clair, Anchore
- CIS CAT (Center for Information Security)

Discovering vulnerabilities by manual:

- Source code review
- Tracking system behavior
- Penetration testing (manual)

Section 2

Problem

How one cycle looks like

Discovering

Life cycle (levels):

- code scanning,
- behavior scanning
- scanning of infrastructure definitions (IaC)
- network scanning
- penetration testing...

Important, price to fix it increases by each level

Processing

Figure 2: 90% Vulnerability Management program

One cycle in product development

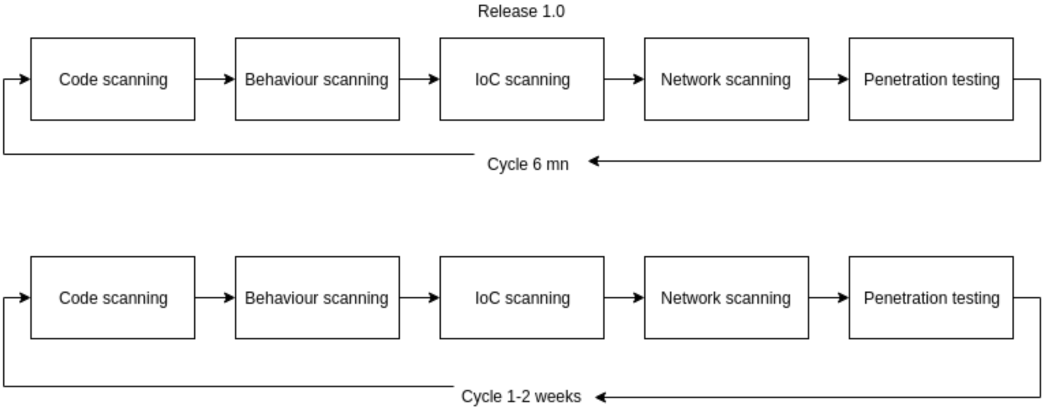
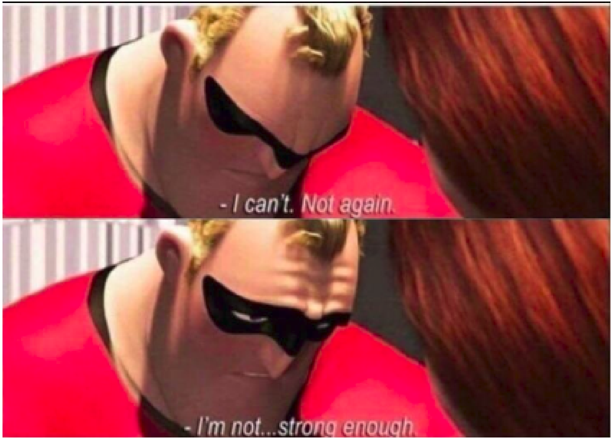


Figure 3: Waterfall vs Agile

Reality



Section 3

DefectDojo

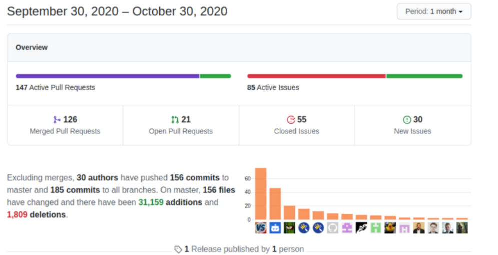
DefectDojo basics

What is DefectDojo

Open source application, dedicated to manage of vulnerabilities:

- OWASP supported
- written in python (django)
- consolidation of finding into single platform

Community



September 30, 2020 – October 30, 2020

Period: 1 month ▾

Overview

147 Active Pull Requests

85 Active Issues

126 Merged Pull Requests

21 Open Pull Requests

55 Closed Issues

30 New Issues

Excluding merges, **30 authors** have pushed **156 commits** to master and **185 commits** to all branches. On master, **156 files** have changed and there have been **31,159 additions** and **1,809 deletions**.

1 Release published by **1** person

Figure 4: Contribution

Features

Major features

Manages application security	Multiple levels of tagging
Application inventory	Activity calendar
Metadata	Archiving of previous assessments
Archiving of results	REST API/Swagger
Password repository	Reporting
Metrics	Data filtering
OWASP ASVS benchmark	Multiple way of data import
Jira, email, slack	Deduplication and FP detection
SAML support	SLA trekking

Tools and reports

How about 63!

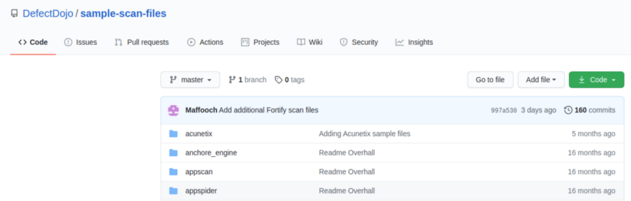
appspider
CHECKMARX
DEPENDENCY-CHECK
snyk
CSV
CONTRAST SECURITY
NMAP
Nessus
OpenVAS
Node Security Platform
VERACODE
WAS
Trustwave TrustKeeper
Qualys
SSL LABS
nexpose
arachni

Tools and reports

Tools

- More than 80 tools (maybe even to much),
- built in integration (Burp, SonarQube, Nmap scanner),
- Google Sheets Integration.

Scan samples



Maffooch Add additional Fortify scan files 997a530 3 days ago **160** commits

acunetix	Adding Acunetix sample files	5 months ago
anchore_engine	Readme Overhall	16 months ago
appscan	Readme Overhall	16 months ago
appspider	Readme Overhall	16 months ago

Figure 5: Samples

Section 4

Installation

Components

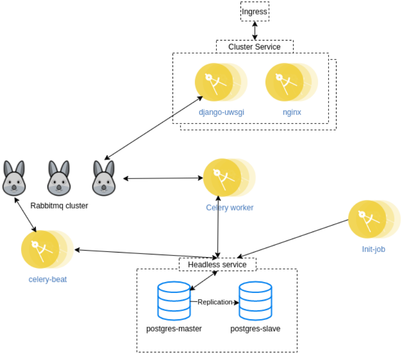


Figure 6: Architecture

Ways of installation

Ways

- From the source code,
- using docker-compose,
- using kubernetes HELM

Supported backends

- Brokers, Redis or Rabbit,
- databases, Postgresql or Mysql.

Section 5

Structure

Structure

Terms

- Product, project, program, (wordpress...)
- Product type, location, part of organization (internal, security...)
- Engagement, period of testing (Beta, Release XYZ)
- Test Type, type of test related to Engagement (Security, Functional)
- Environment, environment under the testing (production, staging...)
- Test, group of activities (Burp Scan from to)
- Finding, item thas been discovered (e.g. OpenSSL vulnerability)

Structure

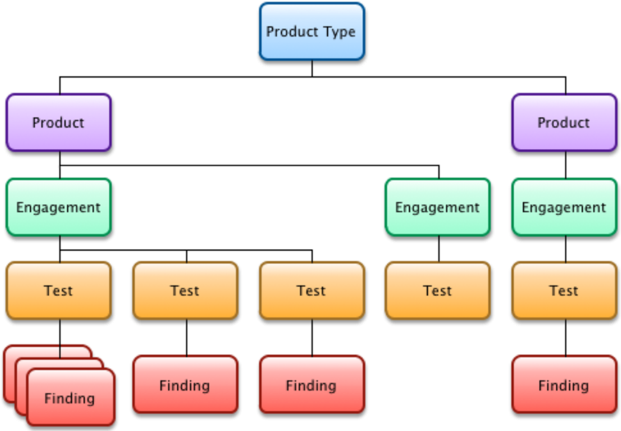
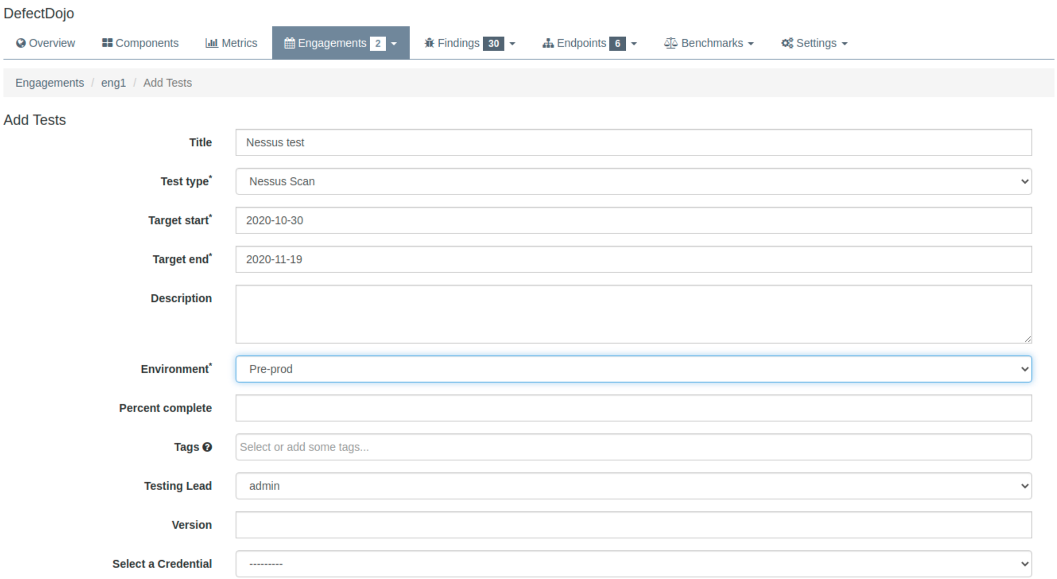


Figure 7: Hierarhija

Section 6

Demo: manual import

Manual import



Engagements / eng1 / Add Tests

Add Tests

Title	Nessus test
Test type*	Nessus Scan
Target start*	2020-10-30
Target end*	2020-11-19
Description	
Environment*	Pre-prod
Percent complete	
Tags ❓	Select or add some tags...
Testing Lead	admin
Version	
Select a Credential	---------

Section 7

Automation

Automation

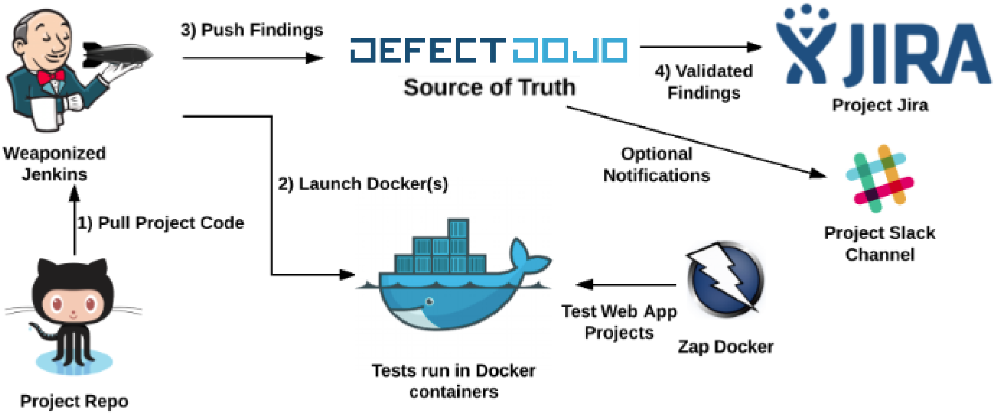


Figure 9: pipline

Section 8

Demo: automated import

Section 9

Development

Advantages

- Process achievement,
- Automation
- Deduplication and notifications
- Good community support
- Agile community

Downsides

- Authorization concept
- Code quality
- Sometimes big imports are slow
- API doesn't reflecting GUI 100%

Resources

Resources

- Official page
- GitHub django
- Sample files
- Documentation
- OWASP SLACK Workspace #defectdojo #defectdojo-dev

Demo

- https://demo.defectdojo.org/login?next=/
- admin/defectdojo@demo#appsec

Section 10

Question?

Question?

- **Email:** Dubravko.Sever@gmail.com
- **Linkedin:** www.linkedin.com/in/dubravko-sever-900b892