Adopting a DevSecOps mindset

union of people, process, technology

Digvijay Patil (DevOps Engineer)

26.07.2022

Agenda

- Industry trend
- Driving the initiative
- Challenges
- Shift left- Continuous Testing
- Continuous Security
- Shift left- Continuous Automation
- Security Automation
- Secure Product Lifecycle
- Hands-on Experience
- Demo
- Maturity Journey

Industry Trends

- Digital Transformation

- Cyber space

- In-house software development to speed up development cycle

>> Leads to adopt right practices

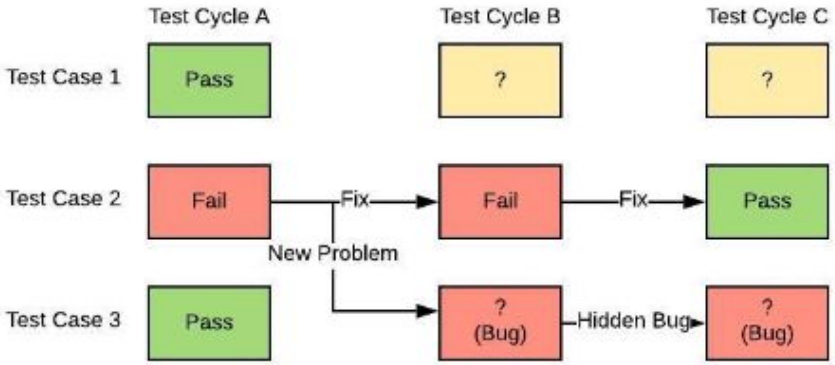
Driving the Initiative..

- Define Processes
- Bridge a gap between teams (DevOps, Security, Product team)
- Evaluate Tools (modern s/w architecture, APIs, shallow learning curve, seamless integration)
- Start Automation
- Modify existing SDLC to Secure SDLC
- Start with small step and shift-left gradually
- Improve the security posture of organization
- Maintain Comprehensive Documentation

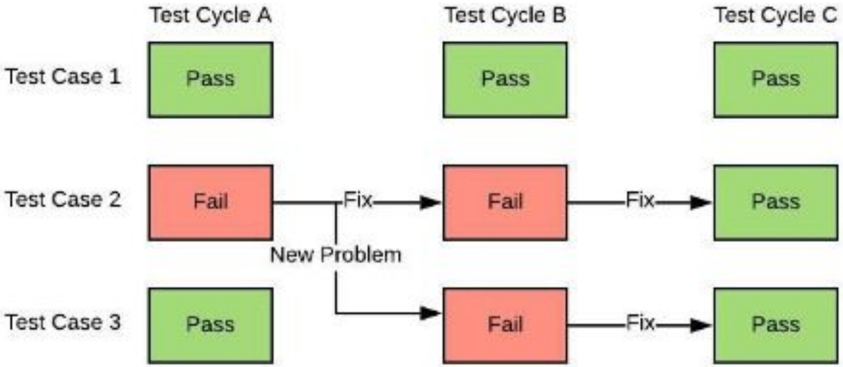
Challenges

- DevOps Engineer
    1. Right choice of tools for integration
    2. Frictionless integration
    3. Define Process
- Security Personnel
    1. Right choice of tools for security
    2. Up-to-date with security knowledge
    3. Make product team comfortable for security practices
    4. Define Process

Shift Left - Continuous Testing



	Test Cycle A	Test Cycle B	Test Cycle C
Test Case 1	Pass	?	?
Test Case 2	Fail	Fail	Pass
Test Case 3	Pass	? (Bug)	? (Bug)



	Test Cycle A	Test Cycle B	Test Cycle C
Test Case 1	Pass	Pass	Pass
Test Case 2	Fail	Fail	Pass
Test Case 3	Pass	Fail	Pass

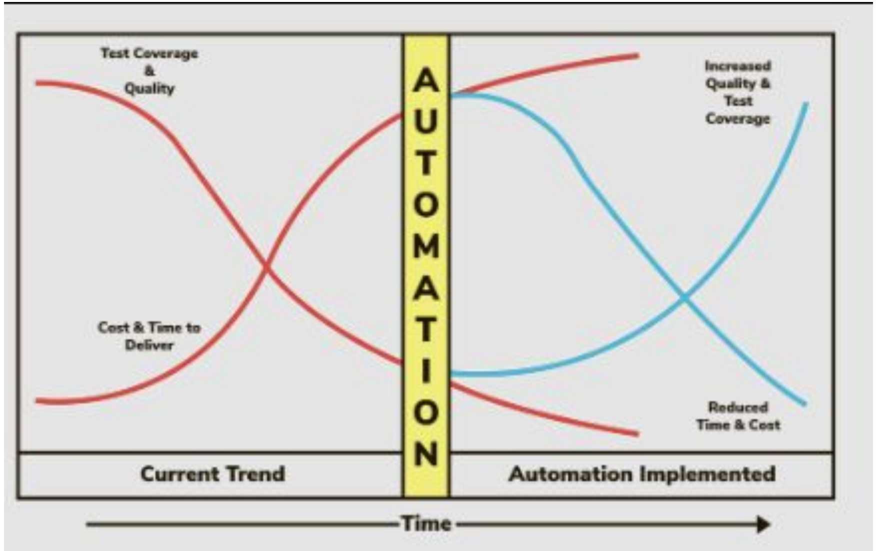
Source: Randy Rice. Software Testing Training - Regression Testing. Youtube. 2013. Url: `https://www.youtube.com/watch?v=5A7J5cM2e7c`

Continuous Security

- Security testing/Security audit on final source code takes 4-5 days.
- We can take small chunk of source code and do the testing on top of it.
- Integrating security in DevOps will help to accommodate this testing.
- **Continuous Security** i.e. CI/**CS**/CD

Shift Left - Continuous Automation

Aim : Discover defects in a short time



Source: `https://www.sealights.io`

The right choice of a tool plays an important role in adopting automation

Security Automation

Why ?

- Overloaded with abundant security alerts
- Various tools for detection, investigation, remediation >> Lots of consoles
- Poor documentation of security processes
- Shortage of talent

Solution

- Plug your security tools into CI/CD practises using Vendor provided APIs/CLI
- Run automated scripts to fetch consolidated data from security tools

Source: Libby Reichenberg. What is Security Orchestration and Automation. Youtube. 2018. Url: `https://www.youtube.com/watch?v=RJafMxfQ_IY&t=361s`

Tool Evaluation

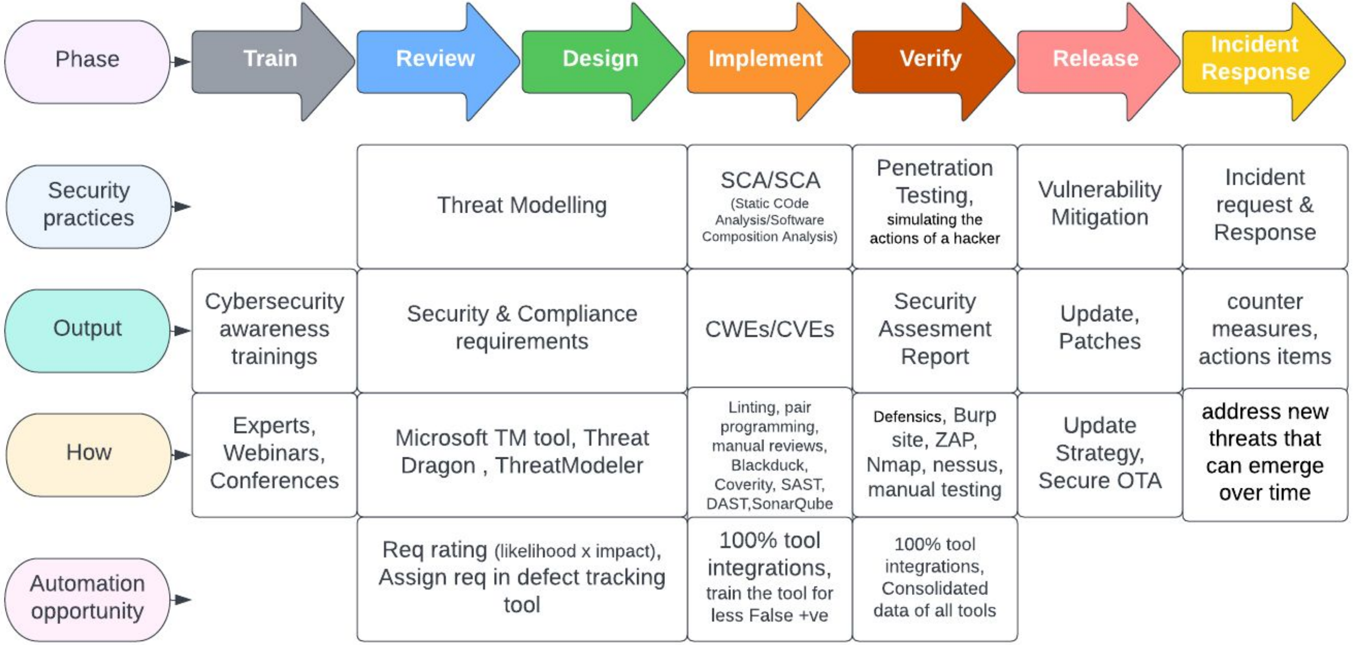
- **From Integration perspective**

1. Enable applications’ data and functionality to third party developers using API (RESTful) service of a application
2. Include authorization credentials, unique tokens, signatures, TLS encryption for API calls
3. Enable built-in command-line interface (CLI) or scripting/console interface support (command prompt, PowerShell, bash, remote terminal programs (PuTTY, SSH))

- **From performance perspective**

1. Provide right scan policy, risk assessment, less false positives
2. Provide good API documentation

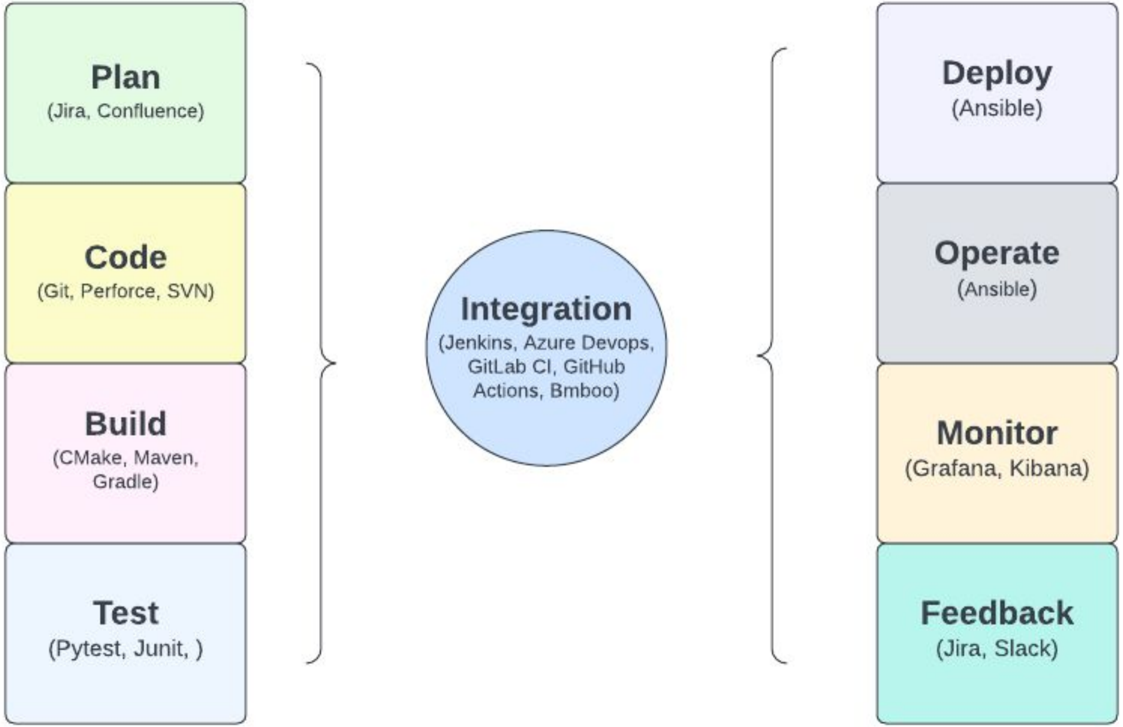
Secure Product Development Lifecycle



Phase	Train	Review	Design	Implement	Verify	Release	Incident Response
Security practices		Threat Modelling		SCA/SCA (Static COde Analysis/Software Composition Analysis)	Penetration Testing, simulating the actions of a hacker	Vulnerability Mitigation	Incident request & Response
Output	Cybersecurity awareness trainings	Security & Compliance requirements		CWEs/CVEs	Security Assesment Report	Update, Patches	counter measures, actions items
How	Experts, Webinars, Conferences	Microsoft TM tool, Threat Dragon , ThreatModeler		Linting, pair programming, manual reviews, Blackduck, Coverity, SAST, DAST,SonarQube	Defensics, Burp site, ZAP, Nmap, nessus, manual testing	Update Strategy, Secure OTA	address new threats that can emerge over time
Automation opportunity		Req rating (likelihood x impact), Assign req in defect tracking tool		100% tool integrations, train the tool for less False +ve	100% tool integrations, Consolidated data of all tools		

Source: Eaton. Secure development lifecycle. 2018. Url: shorturl.at/rvxZ0.

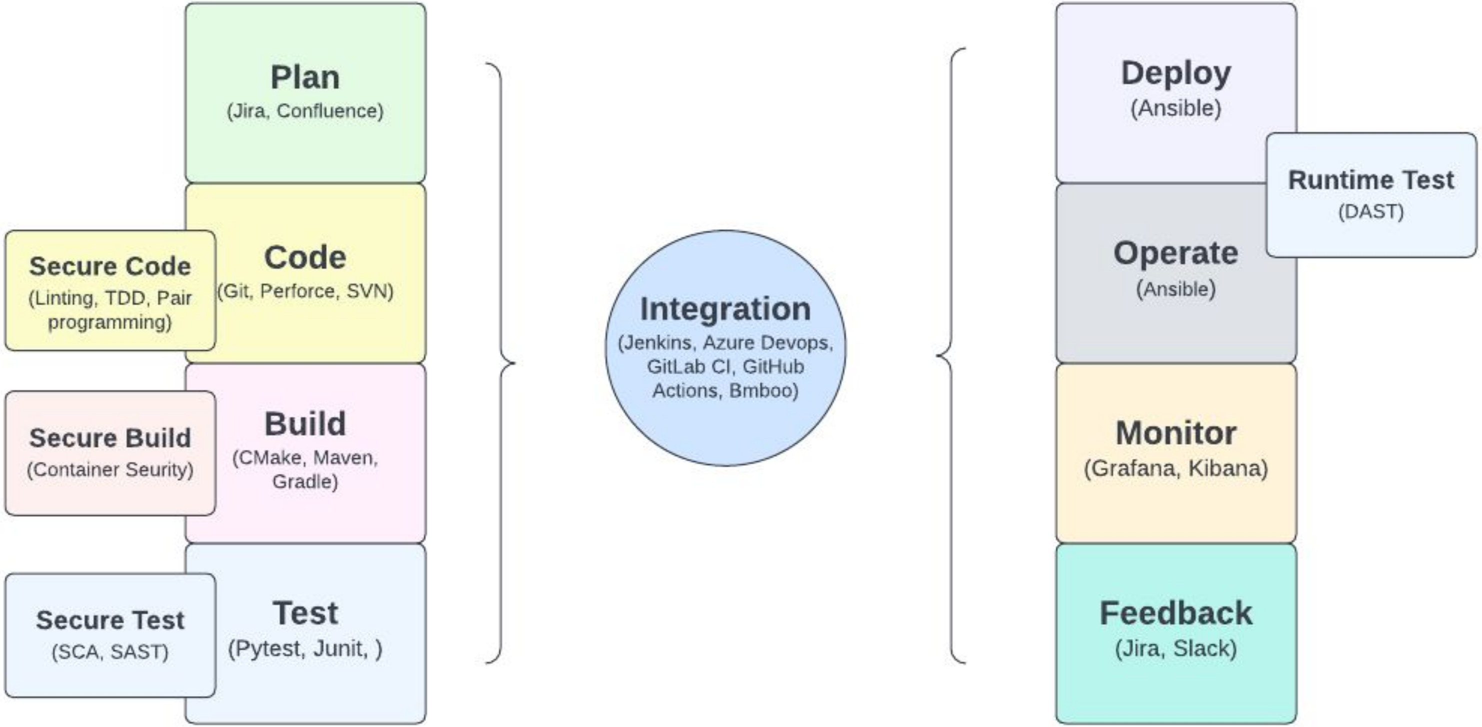
DevOps



Plan
(Jira, Confluence)
Code
(Git, Perforce, SVN)
Build
(CMake, Maven, Gradle)
Test
(Pytest, Junit,)
Integration
(Jenkins, Azure Devops, GitLab CI, GitHub Actions, Bmboo)
Deploy
(Ansible)
Operate
(Ansible)
Monitor
(Grafana, Kibana)
Feedback
(Jira, Slack)

DevSecOps

Plug-in Security Testing methods into Devops



Plan
(Jira, Confluence)

Secure Code
(Linting, TDD, Pair programming)

Code
(Git, Perforce, SVN)

Secure Build
(Container Seurity)

Build
(CMake, Maven, Gradle)

Secure Test
(SCA, SAST)

Test
(Pytest, Junit,)

Integration
(Jenkins, Azure Devops, GitLab CI, GitHub Actions, Bmboo)

Deploy
(Ansible)

Runtime Test
(DAST)

Operate
(Ansible)

Monitor
(Grafana, Kibana)

Feedback
(Jira, Slack)

Hands-on Experience

- Configured Authorization flows (Jenkins, GitHub, Slack, Redmine)
  - OAuth 2.0 Protocol Integration

- Coordinated with ITOps & NetOps Team for CI infrastructure maintenance
  - Asset Management tool (Lansweeper)
  - security management tool (Falcon Sensor)

- DAST tool integration

Demo

Perform DAST scan on demo website http://demo.testfire.net

- **Tools**

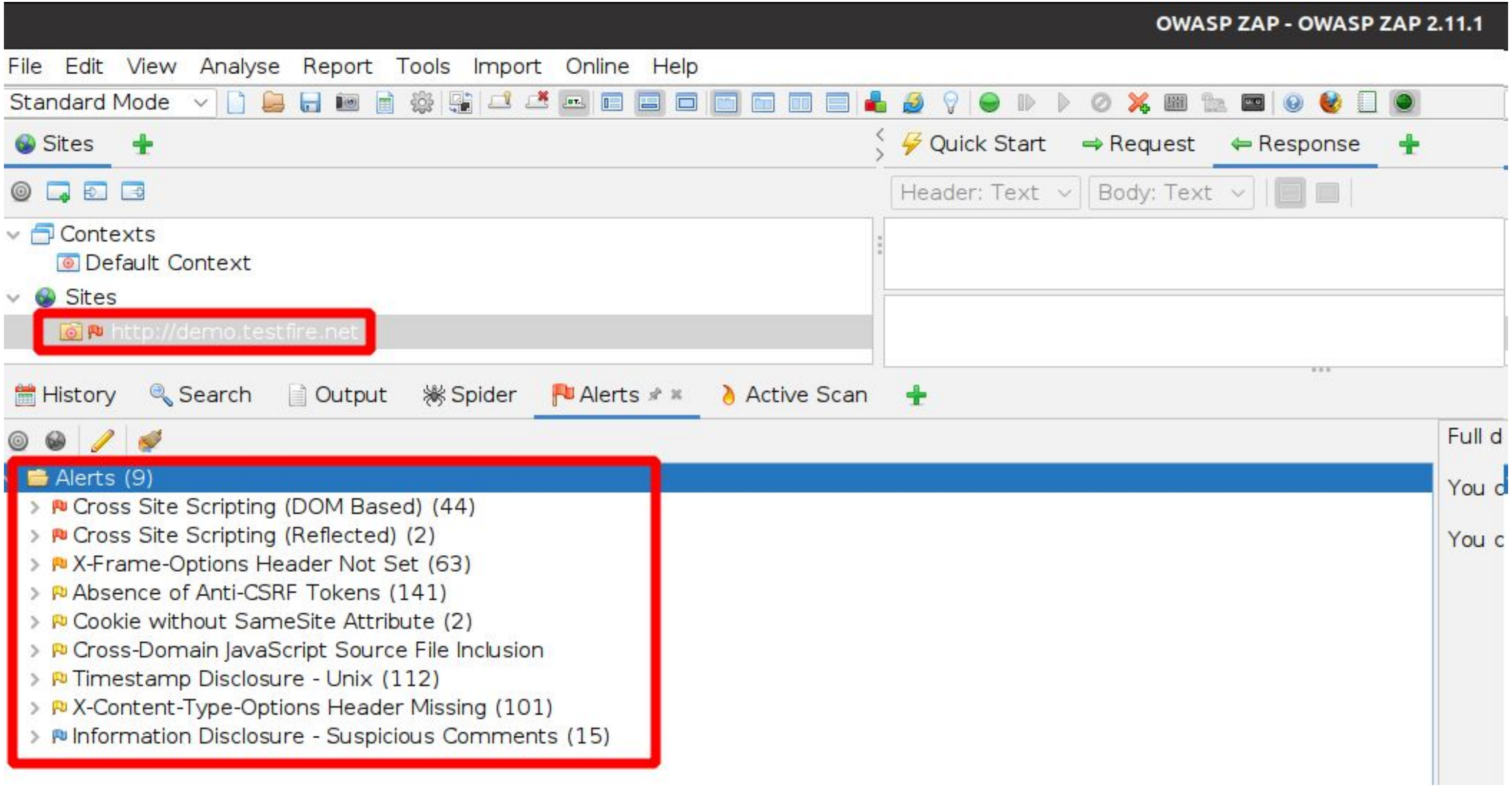
1.	Jenkins	v.2.346.2
2.	PowerShell	v1.7
3.	Git	v2.37.1
4.	ZAP	v2.11.1

- Using OWASP ZAP Standalone application
- Using OWASP ZAP CLI

```
java -jar zap-2.11.1.jar -cmd -quickurl http://demo.testfire.net -quickprogress -quickout report.xml
```

- Report back handful vulnerabilities

Standalone DAST

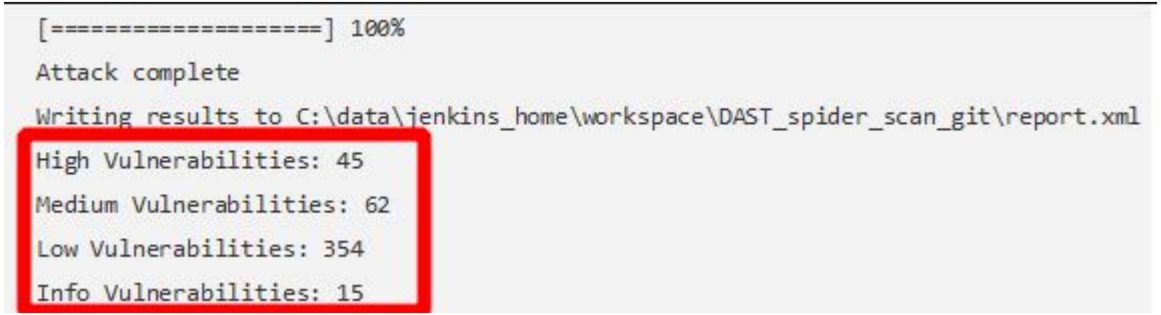


Alerts (9)
- Cross Site Scripting (DOM Based) (44)
- Cross Site Scripting (Reflected) (2)
- X-Frame-Options Header Not Set (63)
- Absence of Anti-CSRF Tokens (141)
- Cookie without SameSite Attribute (2)
- Cross-Domain JavaScript Source File Inclusion
- Timestamp Disclosure - Unix (112)
- X-Content-Type-Options Header Missing (101)
- Information Disclosure - Suspicious Comments (15)

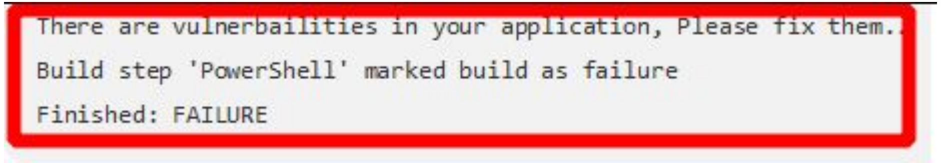
Automated DAST

- Install CI Server (Jenkins) on your machine
- Install Powershell plugin on CI Server
- Install Git, Java, Powershell on your machine
- Choose demo website for DAST scan
- Copy ZAP jar file from GitHub in Jenkins Workspace
- Run java command against demo website
- Export scan result into XML file
- Powershell script to fail/pass the build
- Archive Artifacts

Automated DAST

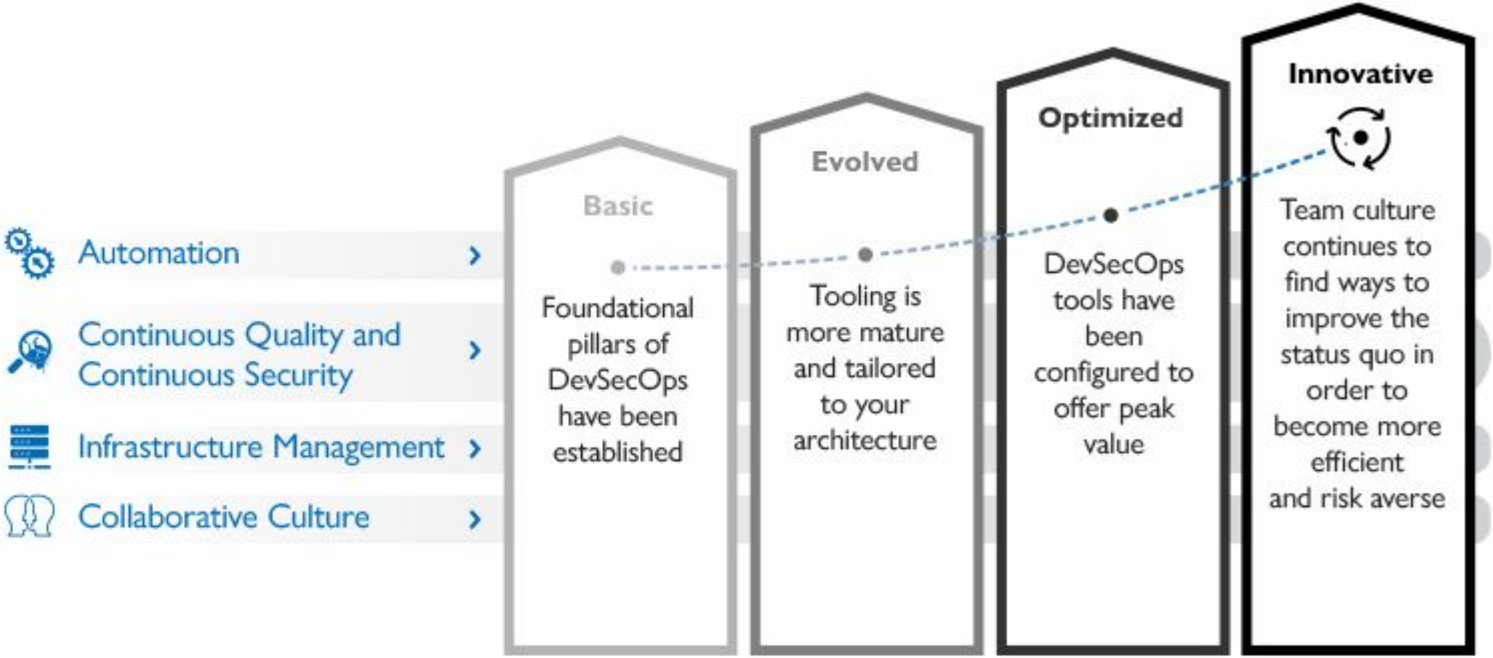


```
[===================] 100%
Attack complete
Writing results to C:\data\jenkins_home\workspace\DAST_spider_scan_git\report.xml
High Vulnerabilities: 45
Medium Vulnerabilities: 62
Low Vulnerabilities: 354
Info Vulnerabilities: 15
```



```
There are vulnerbailities in your application, Please fix them..
Build step 'PowerShell' marked build as failure
Finished: FAILURE
```

DevSecOps Maturity Journey



Automation

Continuous Quality and Continuous Security

Infrastructure Management

Collaborative Culture

Basic

Foundational pillars of DevSecOps have been established

Evolved

Tooling is more mature and tailored to your architecture

Optimized

DevSecOps tools have been configured to offer peak value

Innovative

Team culture continues to find ways to improve the status quo in order to become more efficient and risk averse

Source: SAIC. DevSecOps Journey. 2021. Url:
https://www.saic.com/features/devsecops-journey-how-to-go-from-a-strong-foundation-to-maturity-and-maximized-results

DevSecOps is not a destination.

It is long-term Journey……………….

....

....

....Thank you!