INSTITUTEOFTECHNOLOY

SCHOOL OF COMPUTING

DEPARTMENTOFSOFTWAREENGINEERING

Coursetitle:SoftwareToolsandPractices

Course code:SEng3051

INDIVIDUALASSIGNMENT

Summitted To: Mr.Esmael

SummitedDate:20/07/2016E

C

SoftwareEngineeringProblemsCausesForTheInitiationOf DevSecOps

• Increasing frequency and complexity of cyber threats: The rise of sophisticated cyberattacks and data breaches highlighted the need for better security measures in software development.
• Growing demand for rapid software delivery: Agile development methodologies and DevOps practices emphasized faster software delivery, but this often came at the expense of security.
• Lack of security expertise in development teams: Traditional software development teams often lacked the necessary security knowledge and skills to effectively address security risks.
• Disconnected security and development processes: Security was often treated as an afterthought, bolted on at the end of the software development process, leading to vulnerabilities and security gaps.
• Manual and time-consuming security testing: Traditional security testing methods were manual and time-consuming, slowing down software delivery and hindering innovation.
• Poor communication and collaboration between development and security teams: Lack of communication and collaboration between development and security teams led to misunderstandings, delays, and security risks.

DevSecOps was initiated to address these problems by integrating security practices into the software development process, enabling teams to identify and remediate security issues early and continuously, without compromising on speed and agility.

WhatisDevSecOps?

integrates security practices into the DevOps pipeline, aiming to prioritize security throughout the software development lifecycle.

DevSecOps works by incorporating security controls, automated testing, and continuous monitoring into the DevOps pipeline. This enables development and operations teams to identify and remediate security issues early in the development process, thus reducing security vulnerabilities

and improving overall software security.

Key principles of DevSecOps include:

• Security as a shared responsibility: Security is not just the responsibility of the security team, but of everyone involved in the software development process.
• Automate security testing: Automated security testing tools can help identify vulnerabilities early and continuously, reducing the risk of security breaches.
• Continuous monitoring: Continuous monitoring of software systems helps detect and respond to security threats in real-time.
• Collaboration and communication: Effective collaboration and communication between development, security, and operations teams is essential for successful DevSecOps implementation.

DevSecOps offers a range of benefits, including:

• Improved software security and reduced risk of data breaches
• Faster identification and remediation of security vulnerabilities
• Greater collaboration between development, operations, and security teams
• Enhanced compliance with regulatory requirements
• Increased efficiency and speed of software delivery

Overall, DevSecOps helps organizations build more secure software, faster and with greater confidence

DevSecOps, which stands for development, security, and operations, is a framework that integrates security into all phases of the software development life cycle. It is a methodology thatintegratessecuritypracticeswithintheDevOps(DevelopmentandOperations)process.It focuses on incorporating security early in the software development lifecycle, rather than treating it as an afterthought. Organizations adopt this approach to reduce the risk of releasing code with security vulnerabilities. Through collaboration, automation, and clear processes,teamssshareresponsibilityforsecurity,ratherthanleavingittotheendwhenissues canbemuchmoredifficultandcostly toaddress. DevSecOps isa criticalcomponentofa multicloud security strategy.

BrieflyExplainDevSecOpsLifecycle

ChatGPT4 | Midjourney:
The DevSecOps lifecycle involves integrating security practices at every stage of the software development process, from planning and coding to testing, deployment, and monitoring, ensuring security is built-in from the start rather than bolted on at the end.

1. Planning:
• Define security requirements and threat models
• Establish security policies and standards
• Integrate security tools into the development pipeline

2. Coding:
• Implement secure coding practices
• Use automated security testing tools
• Perform peer code reviews

3. Testing:
• Conduct security testing throughout the development process
• Use automated vulnerability scanners and penetration testing tools
• Integrate security testing into continuous integration (CI) pipelines

4. Deployment:
• Deploy software securely using automated deployment tools
• Configure security settings on servers and infrastructure
• Monitor software for security vulnerabilities

5. Monitoring:
• Continuously monitor software for security threats
• Use security information and event management (SIEM) tools
• Respond to security incidents promptly

By integrating security practices into every stage of the software development lifecycle, DevSecOps helps organizations build more secure software

HowDevSecOpsWorks?

DevSecOps is a practice that integrates security measures at every stage of the software development lifecycle.it involves collaborationbetween development, security andoperations teams toensure that securityisconsideredfromthebeginningofaprojectratherthanbeingaddedasanafterthought.Soit does like the following:

1. IntegrationofSecurity

In DevSecOps, security is integrated throughout the softwaredevelopment pipeline. Security considerationsaremadeateverystage,fromplanningandcodingtotestinganddeployment.

2. AutomationofSecurityProcesses

Securitypracticesareautomatedwhereverpossibletoensureconsistencyandefficiency.Thisinclu des automated security testing, vulnerability scanning, and compliance checks.

3. CollaborationandCommunication

DevSecOpspromotescollaborationbetweendevelopment,operations,andsecurityteams.Bybreakin
g
downsilosandfosteringcommunication,teamscanbetteridentifyandaddresssecurityvulnerabilities.

4. ContinuousMonitoring

ContinuousmonitoringisakeyaspectofDevSecOps.Teamsusemonitoringtoolstodetectandrespon
d to security threats in real-time, thus enhancing the overall security posture of the software.

5. ShiftLeftApproach

DevSecOps follows a shift left approach, which means addressing security concerns early in the developmentprocess.Byidentifyingandmitigatingsecurityissuesattheoutset,thelikelihoodof vulnerabilities slipping through to production is reduced.

6. SecurityasCode

Developerswritesecurityconfigurationsascode,justliketheydowiththeapplicationcode.Thispractice ensures that security policies are version controlled, automated, and auditable.

7. SharedResponsibility

DevSecOps emphasizes that security is everyone's responsibility, not just the security team's.Developers,operations,andsecurityprofessionalsallplayaroleinensuringthesecurityofthes oftware.

To sum up , DevSecOps aims to create a culture of security awareness and accountability within the organization.Byembeddingsecuritypracticesintoeverystageofthesoftwaredevelopmentlifecycle, DevSecOps helps in building secure, resilient, and high-quality software products.

ExplainwellknownDevSecOpstools

TheDevSecOpsapproachrequirestheuseofvarioustoolsandstrategiestoidentifyand address

security risks.

DevSecOps

StaticAnalysis Audit

SoftwareCompositionAnalysis(SCA)Tools:

1. ***OWASP Dependency-Check***:OWASP Dependency-Check is a software compositionanalysistoolthatidentifiesknownvulnerabilitiesinprojectdependencies.
2. ***Retire.js***:Retire.jsisascannerthatdetectsvulnerableJavaScriptlibrariesinyourweb application.
3. ***WhiteSourceBolt***:WhiteSourceBoltisanopen-sourceSCAtoolthatscansyourproject dependencies for known vulnerabilities and provides actionable remediation steps.
4. ***Dependency-Track***:Dependency-Track is an open-source platform that tracks and monitorsyourproject'sdependencies,providinginsightsintotheirknownvulnerabilities.
5. **OSSIndex**.OSSIndexisanopen-sourcevulnerabilitydatabaseandanalysisplatformthat integrates with various development tools to provide real-time security intelligence on project dependencies

StaticApplicationSecurityTesting(SAST)Tools:

1. ***SonarQube***:SonarQubeisanopen-sourceplatformforcontinuouscodequalityinspection that includes static code analysis for identifying security vulnerabilities.
2. ***Bandit***:BanditisaPython-focusedSASTtoolthatanalyzesPythoncodeforcommon security issues and vulnerabilities.
3. ***SpotBugs***:FindBugsisanopen-sourcestaticanalysistoolforJavaapplicationsthat detects common coding errors, potential vulnerabilities, and performance issues.

4. ***RIPS***:RIPSisanopen-sourcePHPsecurityanalysistoolthathelpsidentifysecurity vulnerabilities and coding flaws in PHP applications.

5. ***PMD***:PMD is an open-source source code analyzer for various programming languages, including Java, JavaScript, and XML, which identifies potential bugs, dead code, and security vulnerabilities.

DynamicApplicationSecurityTesting(DAST)Tools:

1. ***OWASPZAP***:OWASPZAP(ZedAttackProxy)isanopen-sourcewebapplicationsecurity scanner that helps you identify vulnerabilities in web applications.
2. ***Nikto***:Niktoisanopen-sourcewebserverscannerthatperformscomprehensivetests against web servers to identify potential vulnerabilities.
3. ***Wapiti***:Wapitiisanopen-sourcewebapplicationvulnerabilityscannerthatauditsthe security of web applications by performing black-box testing.
4. ***Arachni***:Arachniisanopen-source,modularwebapplicationsecurityscannerthatchecks for a wide range of vulnerabilities and provides comprehensive reports.
5. ***Grabber***:Grabberisanopen-sourcewebapplicationscannerthatdetectssecurity vulnerabilities by crawling and scanning web pages.

ContainerSecurityTools:

1. ***Clair***:Clairisanopen-sourcecontainervulnerabilityscannerthatanalyzescontainer images and provides reports on known vulnerabilities.
2. ***Trivy***:Trivyisanopen-sourcevulnerabilityscannerforcontainersandotherartifacts,such as operating system packages and application dependencies. It scans containerimagesandprovidesdetailedreportsonanyvulnerabilitiesdetected,including

theirseverityandremediationsteps.

3. *AnchoreEngine*:AnchoreEngineisanopen-sourcetoolforanalyzingcontainerimages for vulnerabilities, policy violations, and best practices.

4. *SysdigFalco*:SysdigFalcoisanopen-sourcebehavioralactivitymonitoringtooldesigned specifically for containers and Kubernetes. It detects and alerts on anomalous behavior and potential security threats in real-time. Falco uses rules and policies to define expected container behavior and raises alerts when deviations occur.

InfrastructureSecurityTools:

1. *OpenSCAP*:OpenSCAPisanopen-sourceframeworkforcompliancecheckingand vulnerability management, which includes capabilities for assessing and securing infrastructure systems.

2. *Lynis*:Lynisisanopen-sourcesecurityauditingtoolthatassessesthesecurity configuration of Linux and Unix-based systems.

3. *Dagda*:Dagdaisanopen-sourcecontainersecurityanalysistoolthatperformsstatic analysis of container images to detect security issues and vulnerabilities.

4. *ScoutSuite*:ScoutSuiteisanopen-sourcemulti-cloudsecurityauditingtoolthatassesses the security posture of containerized infrastructure in public cloud environments.

ComplianceTools:

1. *OpenSCAP*:OpenSCAPisaSecurityContentAutomationProtocol(SCAP)frameworkfor compliance checking, vulnerability management, and measurement.

2. *OpenVAS*:OpenVAS (Open Vulnerability Assessment System) is a full-featured

vulnerabilityscannerthatcandetectsecurityvulnerabilitiesinsystemsandnetworks.

3. **Wazuh**:Wazuhisanopen-sourcehost-basedintrusiondetectionsystem(HIDS)that helps with compliance monitoring, file integrity monitoring, and log analysis.

DashboardTools:

1. ***Grafana***.Grafanaisanopen-sourceanalyticsandmonitoringplatformthatallowsyouto create customizable dashboards for visualizing various metrics and data sources.
2. ***Kibana***:Kibanaisanopen-sourcedatavisualizationdashboardforElasticsearch,usedfor exploring, analyzing, and visualizing data stored in Elasticsearch indices.
3. ***Metabase***:Metabaseisaneasy-to-useopen-sourcebusinessintelligenceandanalytics tool that allows you to create dashboards and visualize data from various sources.

VulnerabilityTrackingTools:

1. ***OWASPDefectDojo***.DefectDojoisanopen-sourcevulnerabilitymanagementtoolthat helps you track and manage vulnerabilities in your applications and infrastructure.
2. ***TheHive***:TheHiveisanopen-sourceincidentresponseandcasemanagementplatform that includes features for tracking and managing vulnerabilities.

Inconclusion,open-sourcetoolsplayacrucialroleinthefieldofcybersecurity,offeringawide rangeofsolutions fordifferent categories suchas SoftwareCompositionAnalysis (SCA),Static ApplicationSecurityTesting(SAST),DynamicApplicationSecurityTesting(DAST),Container Security, and Infrastructure Security. These tools provide valuable support in identifying vulnerabilities, assessing security risks, and ensuring compliance.

WhatAreTheBenefitsOfDevSecOps?

Benefits of DevSecOps:

• Improved software security and reduced risk of data breaches: DevSecOps helps organizations build more secure software by integrating security practices into every stage of the software development lifecycle. This reduces the risk of security vulnerabilities and data breaches.
• Faster identification and remediation of security vulnerabilities: Automated security testing and continuous monitoring tools help identify and remediate security vulnerabilities early in the development process, reducing the time it takes to fix security issues.
• Greater collaboration between development, operations, and security teams: DevSecOps fosters collaboration between development, operations, and security teams, breaking down silos and improving communication. This leads to better decision-making and more secure software.
• Enhanced compliance with regulatory requirements: DevSecOps helps organizations meet regulatory compliance requirements by providing visibility into the security posture of software systems and automating security controls.
• Increased efficiency and speed of software delivery: By automating security testing and integrating security into the development process, DevSecOps reduces the time it takes to deliver secure software.

Overall, DevSecOps helps organizations build more secure software, faster and with greater confidence.

ThebenefitsofimplementingDevSecOpsinsoftwaredevelopmentinclude:

Rapid,cost-effectivesoftwaredelivery

When software is developed in a non-DevSecOps environment, security problems can lead to hugetimedelays.Fixingthecodeandsecurityissuescanbetime-consumingandexpensive.The rapid, secure delivery of DevSecOps saves time and reduces costs by minimizing the need to repeat a process to address security issues after the fact.
Thisprocessbecomesmoreefficientandcost-effectivesinceintegratedsecuritycutsout duplicative reviews and unnecessary rebuilds, resulting in more secure code.

Improved,proactivesecurity

DevSecOps introduces cybersecurity processes from the beginning of the development cycle. Throughout the development cycle, the code is reviewed, audited, scanned and tested for
securityissues.Theseissuesareaddressedassoonastheyareidentified.Securityproblemsare fixed before additional dependencies are introduced. Security issues become less expensive to fix when protective technology is identified and implemented early in the cycle. Additionally, better collaboration between development, security and operations teams improves an organization’s response to incidences and problems when they occur. DevSecOps practices reduce the time to patch vulnerabilities and free up security teams to focus on higher value work. These practices also ensure and simplify

compliance,savingapplicationdevelopmentprojectsfromhavingtoberetrofittedforsecurity.

Acceleratedsecurityvulnerabilitypatching

AkeybenefitofDevSecOpsishowquicklyitmanagesnewlyidentifiedsecurityvulnerabilities.As DevSecOps integrates vulnerability scanning and patching into the release cycle, the ability to identify and patch common vulnerabilities and exposures (CVE) is diminished. This capability limits the window that a threat actor has to take advantage of vulnerabilities in public-facing production systems.

Automationcompatiblewithmoderndevelopment

Cybersecuritytestingcanbeintegratedintoanautomatedtestsuiteforoperationsteamsifa n organizationusesacontinuousintegration/continuousdeliverypipelinetoshiptheirsoftwa re. Automation of security checks depends strongly on the project and organizational goals.
Automated testing can ensure that incorporated software dependencies are at appropriate patchlevels,andconfirmthatsoftwarepassessecurityunittesting.Plus,itcantestandsecure code with static and dynamic analysis before the final update is promoted to production.

Generally, DevSecOps enables organizations to build and deliver secure software at a faster pace, with reducedsecurityrisksandimprovedcollaborationacrossteams,ultimatelyleadingtomoreresilientand secure applications.

DevOpscareerpaths

Local and international DevSecOps career opportunities:

DevSecOps is a rapidly growing field, with high demand for skilled professionals. Career opportunities exist in a variety of industries, including technology, finance, healthcare, and government.

Local DevSecOps career opportunities:

- DevSecOps engineer
- Security automation specialist
- Security architect
- Compliance analyst

• Cloud security engineer

International DevSecOps career opportunities:

• DevSecOps engineer
• Security automation specialist
• Security architect
• Compliance analyst
• Cloud security engineer
• Site reliability engineer (SRE)
• DevOps engineer with security focus

Career path in DevSecOps:

Individuals interested in a career in DevSecOps typically have a background in software development, security, or operations. Common career paths include:

• Software developer: Gain experience in software development and security best practices.
• Security engineer: Gain experience in security testing, vulnerability management, and incident response.
• Operations engineer: Gain experience in system administration, network security, and cloud computing.

Certifications:

Obtaining relevant certifications can enhance your career prospects in DevSecOps. Some popular certifications include:

• Certified DevSecOps Engineer (CDSE)
• Certified Kubernetes Security Specialist (CKS)
• Certified Information Systems Security Professional (CISSP)
• Certified Ethical Hacker (CEH)

Tips for advancing your DevSecOps career:

• Gain experience in both software development and security.
• Obtain relevant certifications.
• Build a strong network of professionals in the DevSecOps community.
• Stay up-to-date on the latest DevSecOps trends

There are several DevOps career paths you can pursue in this exciting and indemandfield.HerearesomeexamplesofthetopDevOpscareerpathsand what each entails, plus what you can expect in terms of your DevOps salary.

DevOpsSoftwareTester

DevOpsSoftwareTesterstestsoftwareapplicationstomakesuretheymeet

stakeholderexpectations.ThisDevOpscareerinvolvesresponsibilitiessuchas:

- Testplanning.
- Designingandimplementingautomatedtestingframeworks.
- Implementcontinuoustestingprocessesandworkflows.
- Qualityassurance.

TobeaDevOps SoftwareTester,befamiliarwithDevOps,softwaredevelopment, and testing principles.Also know our way around testing frameworks, continuous testing tools, and quality assurance frameworks. we can learn more about the various DevOps tools and software by reading our product highlight.

JuniorDevOpsEngineer

One of the most common entry-level positions in this field is the Junior DevOps Engineer.AJuniorDevOpsEngineerworksunderSeniorDevOpsEngineersand has several responsibilities, such as:

- Troubleshootingissues.
- Writingscripts.
- Completingstandardsystemadministrationtasks.

JuniorEngineersmayalsobetaskedwithenhancingandmaintainingDevOps processes.

Tobecome aJunior DevOpsEngineer,you should have a solid understandingof operatingsystems,cloudinfrastructure,andprogramminglanguages.Youshould also be well-versed in DevOps principles and practices, including automation, continuous integration and deployment, monitoring, and source code management. DevOps Engineer

ADevOps Engineer builds, maintains, and enhances DevOps processes and infrastructure.Theyoftenworkalongsidedevelopment,testing,andoperations teams, ensuring the software delivery pipeline is smooth and

efficient.Managerialinnature,theDevOpsEngineerpositionabsorbsseveral roles and responsibilities.

Performthefollowingtasksrepeatedly

- Writingscriptsthatdeploy.
- Debug,andtestsoftware.
- Buildingreusablecodeforyourorganization.
- Collaboratewithdevelopers,gettingfeedbacktodeterminesoftwarecondition.

You will also need to keep projects on track by troubleshooting issues as they

popupwhilealsokeepingteammembersmotivatedtomeetgoals.Andyoumay also need to adapt to changes on the fly usingAgile principles, make sure that computer systems and networks are running as they should, and, most importantly,promoteaculturethatleadstothetimelydevelopmentofhighquality software.

DevOps engineers should have extensive technical knowledge in scripting and languages like Python, Ruby, or JavaScript. They should also be comfortable workingwithconfigurationmanagementtools, automationframeworks,andLinux environments or shells. Many employers require at least a bachelor’s degree in softwaredevelopment,softwareengineering,computerprogramming,orasimilar field. Beyond those technical requirements, soft skills like collaboration, time

management,andleadershipcanbehelpfulduringyourDevOpscareerasan engineer.

DevOpsArchitect

ADevOpsArchitect is in charge of designing and implementing DevOps processes and infrastructure to meet an organization's specific needs. Responsibilities of this DevOps career path begin with collaborating with developers,IToperations,executives,andotherstakeholderstodiscoverthe company's requirements and devise a DevOps strategy that fulfills them.

DevOpsArchitects work with development teams to ensure infrastructure matchessoftwareapplicationneedswhilebeingscalable.Additionally,theyare responsible for:

- Designingandimplementingsystemsfortesting.
- Deploymentandmonitoringtoenhancesoftwaredeliveryprocesses.
- Evaluatingandselectingnewtechnologiesandtoolstooptimize DevOps pipelines.

ADevOpsArchitect should have a broad knowledge of system administration, infrastructuredesign,andsoftwaredevelopment.Theyshould bewell-versedin how cloud infrastructure, containerization, and orchestration work, while also having familiarity with automation tools and frameworks that can help enhance software delivery.Tobe able torecommend propertechnology tostakeholders, DevOpsArchitectsmuststayup-to-dateonthelatestDevOpstrends.Andsince they must foster collaboration between teams and stakeholders, DevOps Architects should also have strong communication skills.

DevOpsReleaseManager

ADevOps Release Manager manages the release of software to ensure it is deliveredontime,uptopar,andwithinbudget.ChoosethisDevOpscareer,and you will plan and coordinate software releases by working with development, testing, and operations teams.

Toensurethatsoftwarereleasesremainreliable,predictable,andrepeatable, DevOps Release Managers must:

- Designandimplementautomatedreleaseprocesses.
- Managechangerequests,ensuringthatanychangesaremadeinamanner that is both auditable and controlled. • Identify and mitigate potential risks that could negatively impact release, plus create contingency plans to fix

References

https://www.mavhem.securitv/blog/the-devsecops-lifecycle-how-to-automate-securitv-in-softwaredevelopment. "Lifecycleof DevSecOps"

https://ranianiitian.medium.com/top-devsecops-tools-for-2023-open-source-solutions-for-enterprises-7c146f80b325."what are well known tools for DevSecOps?"