



OWASP

Open Web Application
Security Project

OWASP中国区域安全论坛合肥站 暨2021年“安全赋能·数字合规”技术沙龙

- 主办单位：OWASP中国、OWASP中国安徽区域
- 承办单位：安徽省软件评测中心、合肥高创股份有限公司、合肥科技创新创业服务中心
- 协办单位：科大讯飞股份有限公司、开源网安技术有限公司、奇安信集团安徽区域



OWASP

Open Web Application
Security Project

OWASP DevSecOps成熟度模型解读

|| 目录



DevSecops 背景介绍



OWASP 推进开源工具项目

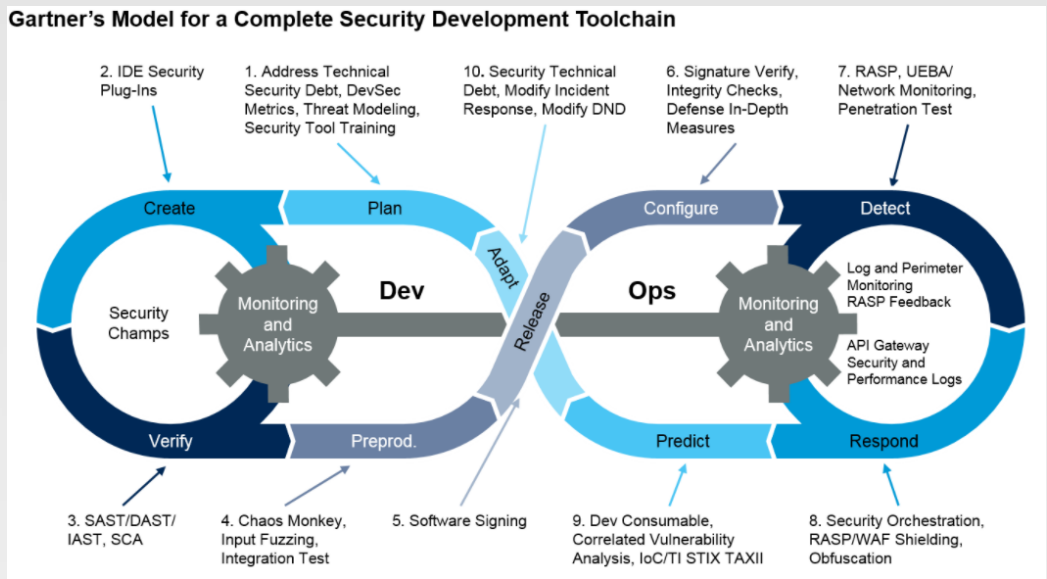
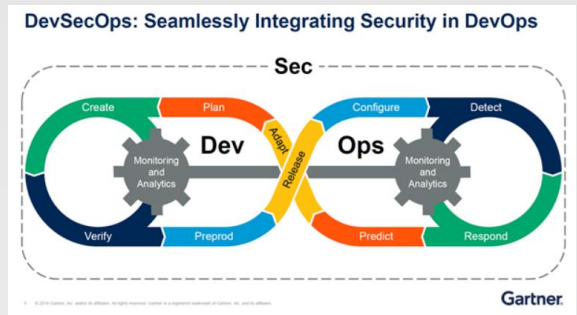


DSOMM 模型详解



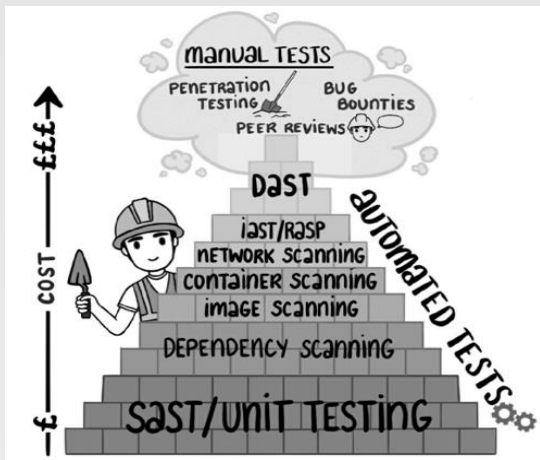
总结: 怎么判断一个企业适合开展DevSecOps

DevSecOps & Gartner



PPTR=人+流程+技术+资源

DevSecOps 体系结构



安全工具链金字塔

来源: DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement



DevOps 基础底座



周边生态环境

总体规划、分步实施

根据企业的规模，大中小不同形态，采取不同的构建方式



建立流程、关键卡点

管理机制、流程，人力为主



工具化、管道建设

工具+平台、安全能力原子化、补短板



自动化、生态运营

自动化编排、数据运营、稳定性保障

|| OWASP 推荐开源安全工具

开源项目地址: https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools

开源项目名称: **Free for Open Source Application Security Tools**

Author: Dave Wichers

Contributor(s): Sherif Koussa, Dirk Wetter

开源工具名录:

SAST

静态代码安全检测

- ① GitHub code scanning
- ② Coverity Scan Static Analysis
- ③ HCL AppScan CodeSweep

DAST

动态安全检测

- ① OWASP ZAP
- ② StackHawk
- ③ Arachni

IAST

交互式安全检测

- ① Contrast Community Edition (CE)
- ② Open RASP/IAST
- ③ 火线~洞态IAST

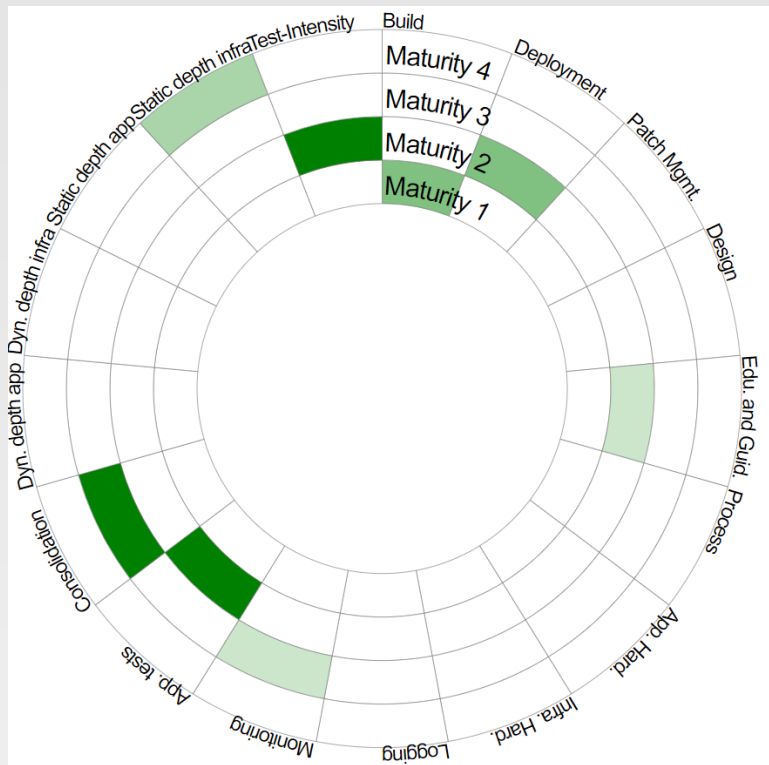
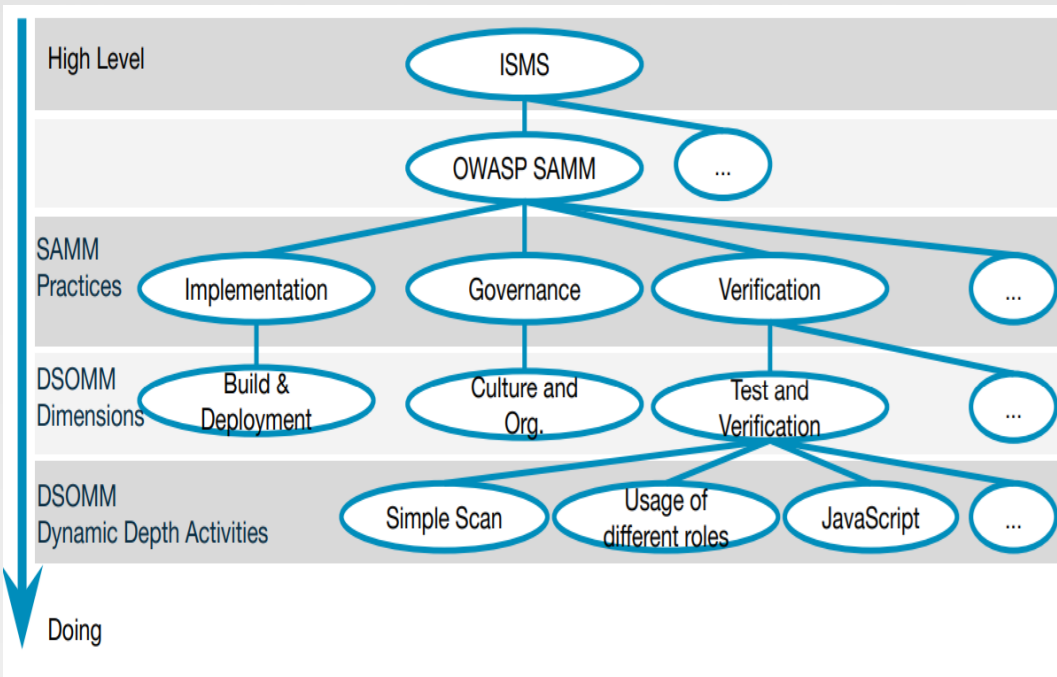
SCA

组件成分分析

- ① OWASP Dependency Track
- ② Dependabot
- ③ WhiteSource

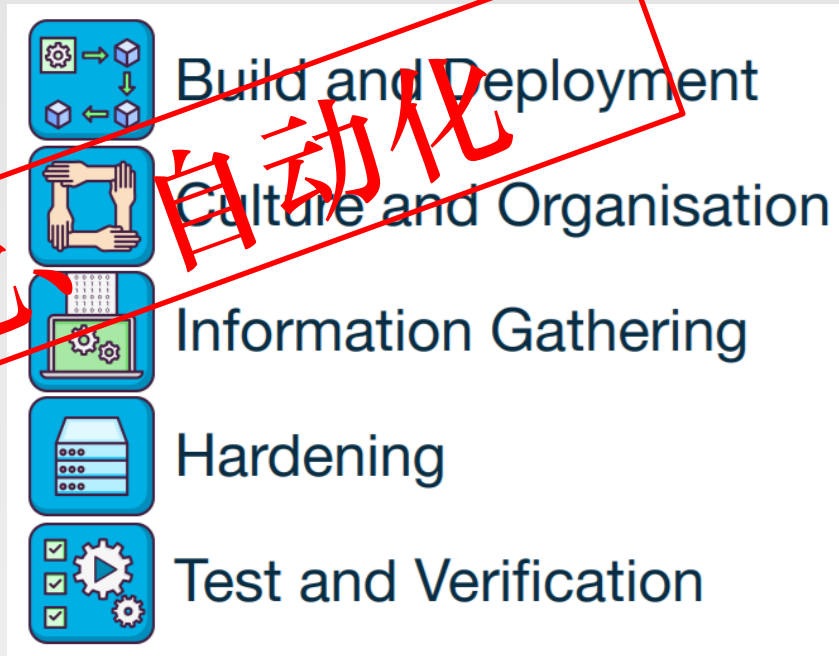
|| OWASP DevSecops成熟度模型

项目地址: <https://dsomm.timo-pagel.de/>






|| OWASP DevSecops成熟度模型构成

- **Level 1:** Basic understanding of security practices
- **Level 2:** Adoption of basic security practices
- **Level 3:** High adoption of security practices
- **Level 4:** Advanced deployment of security practices at scale





四个成熟度级别之间的联系和区别-Build and Deployment

Dimension	Sub-Dimension	Level 1: Basic understanding of security practices	Level 2: Adoption of basic security practices	Level 3: High adoption of security practices	Level 4: Advanced deployment of security practices at scale
 Build and Deployment	Build	<ul style="list-style-type: none">• Continuous integration• Defined build process	<ul style="list-style-type: none">• Building and testing of artifacts in virtual environments• Pinning of artifacts	<ul style="list-style-type: none">• Signing of artifacts• Signing of code	
 Build and Deployment	Deployment	<ul style="list-style-type: none">• Defined deployment process	<ul style="list-style-type: none">• Environment depending configuration parameters (secrets)• Usage of trusted images	<ul style="list-style-type: none">• Handover of confidential parameters• Inventory of running artifacts• Rolling update on deployment• Same artifact for environments• Usage of feature toggles	<ul style="list-style-type: none">• Blue/Green Deployment
 Build and Deployment	Patch Management	<ul style="list-style-type: none">• A patch policy is defined• Automated PRs for patches	<ul style="list-style-type: none">• Nightly build of images (base images)• Reduction of the attack surface• Usage of a maximum lifetime for images		<ul style="list-style-type: none">• Usage of a short maximum lifetime for images

四个成熟度级别之间的联系和区别-Culture and Organization

 Culture and Organization	Design	<ul style="list-style-type: none">• Conduction of simple threat modeling on technical level	<ul style="list-style-type: none">• Conduction of advanced threat modeling• Conduction of simple threat modeling on business level• Creation of simple abuse stories• Creation of threat modeling processes and standards	<ul style="list-style-type: none">• Creation of advanced abuse stories
 Culture and Organization	Education and Guidance	<ul style="list-style-type: none">• Ad-Hoc Security trainings for software developers• Security code review• Security consulting on request	<ul style="list-style-type: none">• Each team has a security champion• Regular security training for all• Regular security training of security champions• Reward of good communication• Simple mob hacking	<ul style="list-style-type: none">• Conduction of build-it, break-it, fix-it contests• Conduction of collaborative security checks with developers and system administrators• Security-Lessonned-Learned• Aligning security in teams• Conduction of collaborative team security checks• Conduction of war games• Regular security training for externals
 Culture and Organization	Process	<ul style="list-style-type: none">• Definition of simple BCDR practices for critical components• Source Control Protection	<ul style="list-style-type: none">• Approval by reviewing any new version• Definition of a change management process• Prevention of unauthorized installation	

四个成熟度级别之间的联系和区别-Implementation & Information Gathering

 Implementation	Application Hardening	<ul style="list-style-type: none"> Application Hardening Level 1 	<ul style="list-style-type: none"> App. Hardening Level 2 	<ul style="list-style-type: none"> App. Hardening Level 3 	<ul style="list-style-type: none"> Full Coverage of App. Hardening Level 3
 Implementation	Infrastructure Hardening	<ul style="list-style-type: none"> Isolated networks for virtual environments Simple access control for systems Usage of test and production environments 	<ul style="list-style-type: none"> Applications are running in virtualized environments Backup Checking the sources of used libraries Filter outgoing traffic The cluster is hardened Usage of security by default for components Virtual environments are limited 	<ul style="list-style-type: none"> 2FA Immutable Infrastructure Infrastructure as Code Role based authentication and authorization Versioning 	<ul style="list-style-type: none"> Limitation of system calls in virtual environments Microservice-Architecture Production near environments are used by developers Usage of a chaos monkey
 Information Gathering	Logging	<ul style="list-style-type: none"> Centralized system logging Logging of security events PII logging concept 	<ul style="list-style-type: none"> Visualized logging 	<ul style="list-style-type: none"> Centralized application logging 	<ul style="list-style-type: none"> Correlation of security events
 Information Gathering	Monitoring	<ul style="list-style-type: none"> Simple application metrics Simple system metrics 	<ul style="list-style-type: none"> Alerting Visualized metrics 	<ul style="list-style-type: none"> Advanced availability and stability metrics Advanced webapplication metrics Deactivation of unused metrics Grouping of metrics Targeted alerting 	<ul style="list-style-type: none"> Coverage and control metrics Defense metrics Metrics are combined with tests Screens with metric visualization

OWASP DevSecops 能力构建

四个成熟度级别之间的联系和区别-Test and Verification

 Test and Verification	Application tests		<ul style="list-style-type: none">• Security unit tests for important components	<ul style="list-style-type: none">• Security integration tests for important components	<ul style="list-style-type: none">• High coverage of security related module and integration tests• Smoke Test
 Test and Verification	Consolidation	<ul style="list-style-type: none">• Definition of quality gates• Simple false positive treatment• Treatment of defects with severity high or higher	<ul style="list-style-type: none">• Simple visualization of defects	<ul style="list-style-type: none">• Integration of vulnerability issues into the development process• Treatment of defects with severity middle• Usage of a vulnerability management system	<ul style="list-style-type: none">• Advanced visualization of defects• Reproducible defect tickets• Treatment of all defects
 Test and Verification	Dynamic depth for applications	<ul style="list-style-type: none">• Simple Scan	<ul style="list-style-type: none">• Coverage of client side dynamic components• Usage of different roles	<ul style="list-style-type: none">• Coverage of hidden endpoints• Coverage of more input vectors• Coverage of sequential operations• Usage of multiple scanners	<ul style="list-style-type: none">• Coverage analysis• Coverage of service to service communication
 Test and Verification	Dynamic depth for infrastructure	<ul style="list-style-type: none">• Test for exposed services	<ul style="list-style-type: none">• Test network segmentation• Test of the configuration of cloud environments	<ul style="list-style-type: none">• Weak password test	<ul style="list-style-type: none">• Load tests• Test for unused Resources
 Test and Verification	Static depth for applications	<ul style="list-style-type: none">• Test of server side components with known vulnerabilities	<ul style="list-style-type: none">• Static analysis for important server side components	<ul style="list-style-type: none">• Static analysis for important client side components• Test of client side components with known vulnerabilities	<ul style="list-style-type: none">• Exclusion of source code duplicates• Static analysis for all components/libraries• Static analysis for all self written components• Stylistic analysis• Usage of multiple analyzers
 Test and Verification	Static depth for infrastructure	<ul style="list-style-type: none">• Stored Secrets	<ul style="list-style-type: none">• Check for image lifetime• Test cluster deployment resources• Test of virtualized environments• Test the cloud configuration• Test the definition of virtualized environments	<ul style="list-style-type: none">• Analyze logs• Check for malware• Check for new image version	<ul style="list-style-type: none">• Check for known vulnerabilities• Correlate known vulnerabilities in infrastructure with new image versions• Test of infrastructure components for known vulnerabilities
 Test and Verification	Test-Intensity	<ul style="list-style-type: none">• Default settings for intensity• High test intensity	<ul style="list-style-type: none">• Deactivating of unneeded tests• Regular tests	<ul style="list-style-type: none">• Creation and application of a testing concept	

|| 总结

怎么去评判企业是否适合做DevSecOps

研发管理成熟度是否适配

IT基础实施是否可自动化

人员能力是否匹配

实际情况是否需要



OWASP

Open Web Application
Security Project

THANKS!