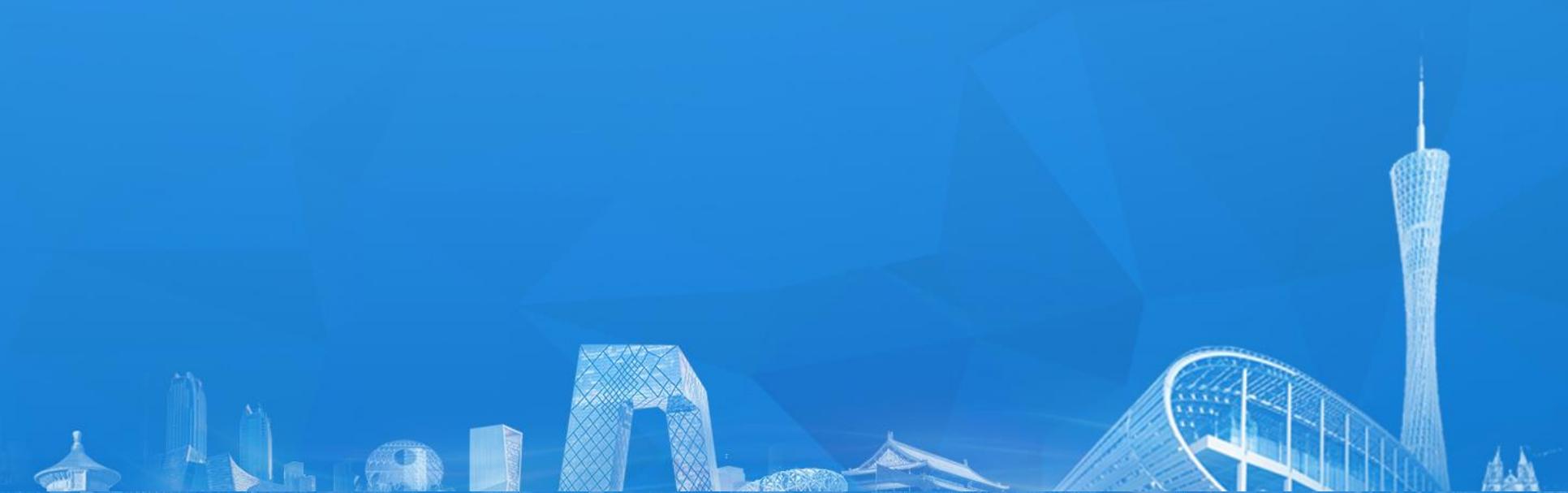


DevSecOps在大型企业的落地实践



内容

- DevSecOps简介
- DevSecOps实现模型
- DevSecOps工具
- DevSecOps成熟度模型

DevSecOps简介

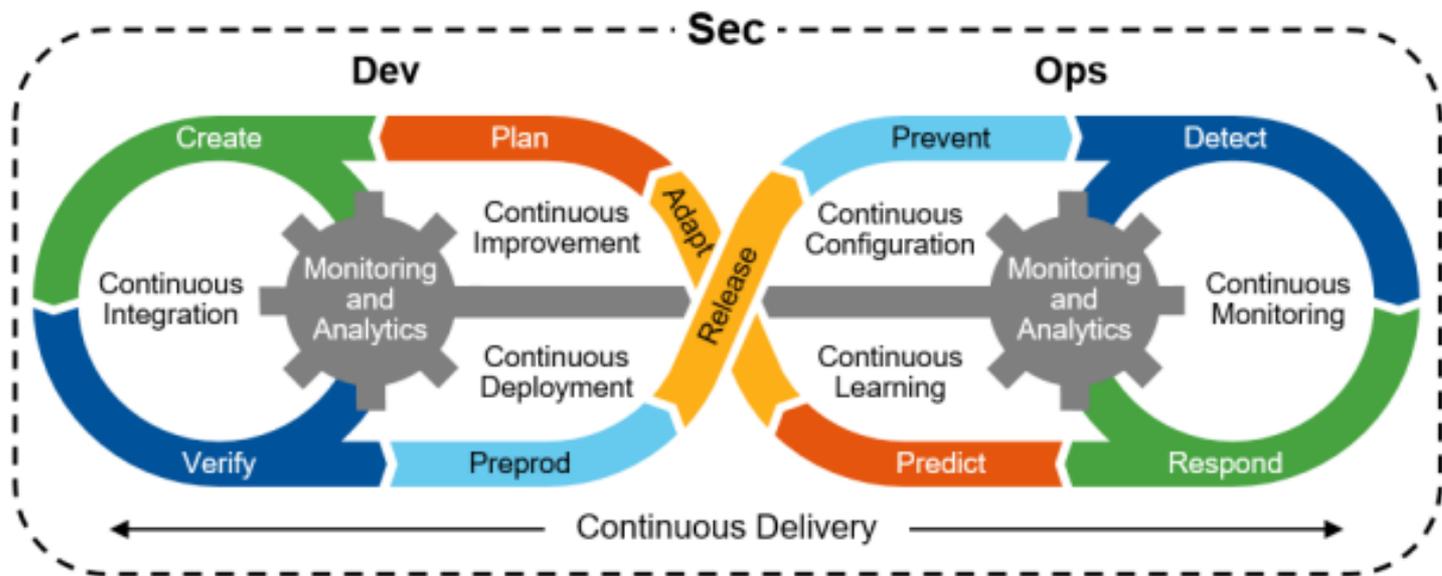
什么是DevSecOps

- 2012年，Gartner创建了“DevSecOps”概念，其原始术语为“DevOpsSec”
- 2017 RSA峰会后，DevSecOps开始成为世界热门话题
- DevSecOps是一种新方法，有助于在开发过程早期而不是在产品发布后识别安全问题，这意味着DevSecOps将安全性从被动转变为主动
- DevSecOps的目标是让每个人都对信息安全负责，而不仅仅是信息安全部门



什么是DevSecOps

DevSecOps旨在将安全性嵌入开发过程的每个部分。这意味着将应用安全思维模式左移到开发团队。

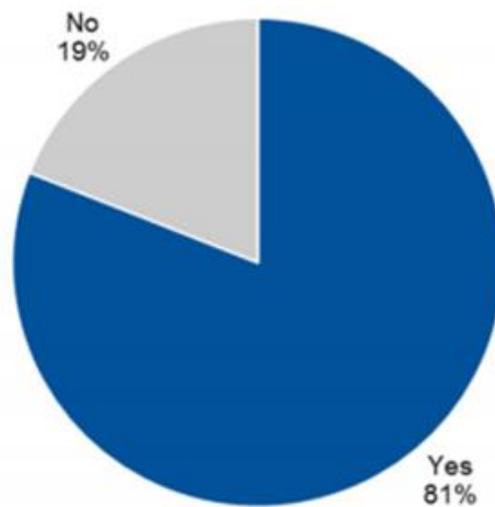


© 2017 Gartner, Inc.

为什么需要DevSecOps

- 应用程序层是2016年数据泄露的主要原因
- DevOps为开发团队带来更快，更好的协作，可以在几周到几天内交付和部署软件。然而快速创新与安全性的冲突，让安全性成为DevOps的瓶颈
- 许多公司里，直到整个软件开发生命周期结束才解决网络安全问题

Figure 2. IT Operations Professionals:
Do You Believe Your Information Security
Policies/Teams Are Slowing IT Down?



n = 93

Source: Gartner (September 2016)

实现DevSecOps的挑战

- 美国威胁检测公司 Threat Stack 针对北美大中小企业200多名安全，开发和运维专业人员的一项新调查和报告表明，DevSecOps 仍然停留在理论阶段
- 造成这种情况的主要原因是缺少高层的支持，业务领导者甚对此并不鼓励。52% 的公司承认会削减安全措施，以便在截止日期前完成目标。报告作者表示：速度成为开发业务中的首要目标，往往需要牺牲安全。

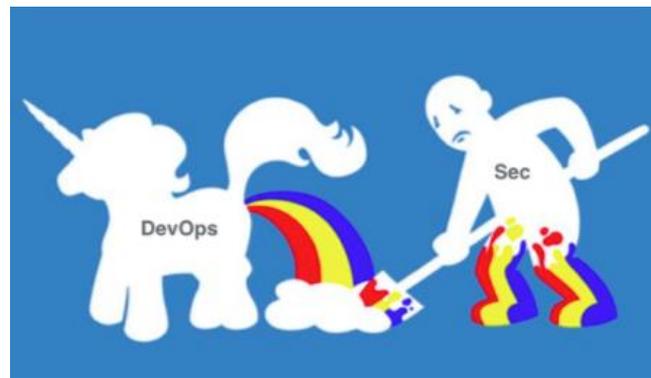


实现DevSecOps的挑战

➤ 62% 的受访者表示，DevOps 不利于在产品中实现安全技术部署；57% 的受访者认为 DevOps 阻碍了安全最佳实践。主要原因都是安全考量与高速开发互相掣肘。

➤ 安全与DevOps分割的三大关键因素

- 安全仍然是孤立的
 - 27%的运维团队配备了安全专家
 - 只有18%的开发团队配备了安全专家
- 开发与安全分割 - 44%开发人员没有接受过安全编码的培训
- 安全与运维分割 - 42%的运维人员没有接受过基本安全实践方面的培训



实现DevSecOps的挑战



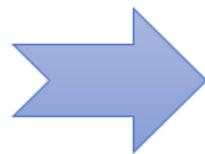
DevSecOps好处

更快 - DevSecOps通过自动化和反馈向需要在发布之前积极参与ISR/IT Sec的团队转移安全需求，从而缩短产品上市时间

控制风险 - 为漏洞识别和修复提供基准安全标准，使项目团队能够在可接受的风险标准内自我认证部署应用程序，从而降低业务和内部开发应用程序的整体风险

节省成本 - DevSecOps减少了对网络安全部门的整体依赖，以动手执行发布后代码和基础架构检查，从而整体降低补救成本

DevSecOps最佳实践



DevSecOps的最终目标是引入一套框架，解决持续快速交付和信息安全之间的矛盾

OWASP Top 10大安全漏洞

- ▶ OWASP是一个开源的、非盈利的全球性安全组织，致力于应用软件的安全研究。他的使命是使应用软件更加安全，使企业和组织能够对应用安全风险做出更清晰的决策。目前OWASP全球拥有250个分部近7万名会员，共同推动了安全标准、安全测试工具、安全指导手册等应用安全技术的发展
- ▶ OWASP在业界被视为web应用安全领域的权威参考。OWASP TOP 10为IBM APPSCAN、HP WEBINSPECT等扫描器漏洞参考的主要标准



T10 OWASP Top 10 应用安全风险-2017

A1:2017 注入	将不受信任的数据作为命令或查询的一部分发送到解析器时，会产生诸如SQL注入、OS注入和LDAP注入的注入缺陷。攻击者的恶意数据可以诱使解析器在没有适当授权的情况下执行非预期命令或访问数据。
A2:2017 失效的身份认证和会话管理	通常，通过错误使用应用程序的身份认证和会话管理功能，攻击者能够破解密码、密钥或会话令牌，或者利用其它开发中的缺陷来冒充其他用户的身份（暂时或永久）。
A3:2017 敏感数据泄露	许多Web应用程序和API都无法正确保护敏感数据，例如：财务数据、医疗保健数据和PII。攻击者可以窃取或修改这些未加密的数据，以进行信用卡诈骗、身份盗窃或其他犯罪。因此，我们需要对敏感数据加密，这些数据包括：传输过程中的数据、被存储的数据以及浏览器交互数据。
A4:2017 XML 外部实体 (XXE)	许多较早的或配置不佳的XML处理器评估了XML文档中的外部实体引用。外部实体可以通过URI文件处理器、在Windows服务器上未修复的SMB文件共享、内部端口扫描、远程代码执行来实施拒绝服务攻击，例如：Billion Laughs攻击。
A5:2017 失效的访问控制	未对通过身份验证的用户实施恰当的访问控制。攻击者可以利用这些缺陷访问未经授权的功能或数据，例如：访问其他用户的帐户、查看敏感文件、修改其他用户的数据、更改访问权限等。
A6:2017 安全配置错误	安全配置错误是数据中最常见的缺陷，这部分缺陷包含：手动配置错误、临时配置（或根本不配置）、不安全的默认配置、开启 S3 bucket、不当的 HTTP 标头配置、包含敏感信息的错误信息、未及时修补或升级（或根本不修补和升级）系统、框架、依赖项和组件。
A7:2017 跨站脚本 (XSS)	每当应用程序的新网页中包含不受信任的、未经过恰当验证或转义的数据，或者使用可以创建 JavaScript 的浏览器 API 更新现有的网页时，就会出现 XSS 缺陷。XSS 缺陷让攻击者能够在受害者的浏览器中执行脚本，并劫持用户会话、污染网站或将用户重定向到恶意站点。
A8:2017 不安全的反序列化	当应用程序接收到恶意的序列化对象时，会出现不安全的反序列化缺陷。不安全的反序列化会导致远程代码执行。即使反序列化缺陷不会导致远程代码执行，也可以重播、篡改或删除序列化对象以欺骗用户、进行注入攻击和提升权限。
A9:2017 使用含有已知漏洞的组件	组件（例如：库、框架和其他软件模块）运行和应用程序相同的权限。如果使用含有已知漏洞的组件，这样的攻击可以造成严重的数据库丢失或服务接管。使用含有已知漏洞的组件的应用程序和 API，可能会破坏应用程序防御、造成各种攻击并产生严重影响。
A10:2017 不足的日志记录和监控	不足的日志记录和监控，以及事件响应集成的丢失或无效，使得攻击者能够进一步攻击系统、保持持续性或转向更多系统，以及篡改、提取或销毁数据。大多数缺陷研究显示，缺陷被检测出的时间超过200天，并且通常通过外部检测方检测，而不是通过内部进程或监控检测。

DevSecOps实现模型

DevSecOps实现模型

- **第一阶段 - 信息安全工具**

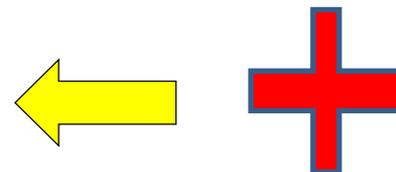
- 将DevSecOps工具嵌入到CICD流水线中实现自动化安全漏洞扫描
- 生成并公开信息安全漏洞报表

- **第二阶段 - 信心安全培训**

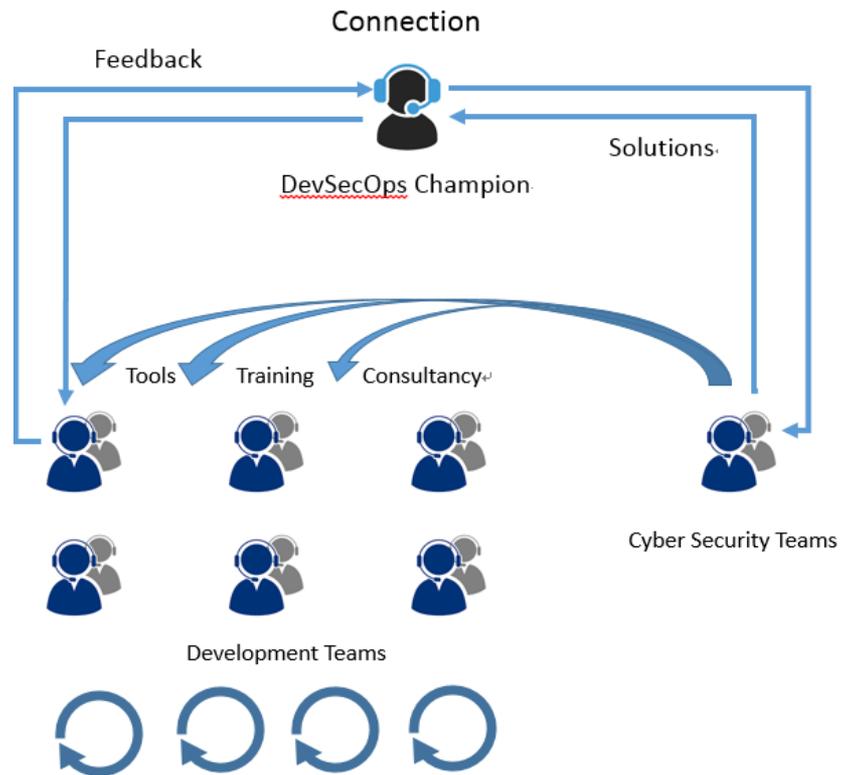
- 信息安全工具学习
- 信息安全知识培训

- **第三阶段 - 信息安全意识和”专家”**

- 建立信心安全意识和文化
- 培养开发团队中的”信息安全专家”



DevSecOps运营模型



DevSecOps工具

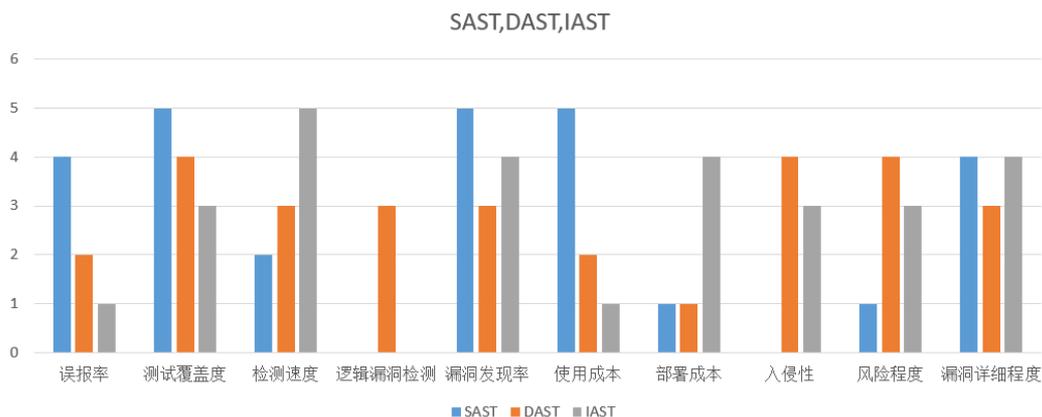
DevSecOps工具

➤ DevSecOps工具通过扫描开发代码和第三方插件，模拟攻击行为，从而帮助开发团队发现开发过程中潜在的信息安全漏洞

➤ 从信息安全角度来看的话，DevSecOps工具可以分为以下几类

- 静态应用安全工具 (SAST)
- 动态应用安全工具 (DAST)
- 交互式应用安全工具 (IAST)
- 开源软件安全工具(FOSS)
- 基础设施安全工具

工具特点和选择的比较



DevSecOps工具

➤ 静态应用安全工具 (SAST)

- Checkmarx
- Fortify
- IBM AppScan



➤ 动态应用安全工具 (DAST/IAST)

- Contrast
- OWASP ZAP



➤ 开源软件安全工具 (FOSS)

- Sonatype IQ Server
- Dependencies Check

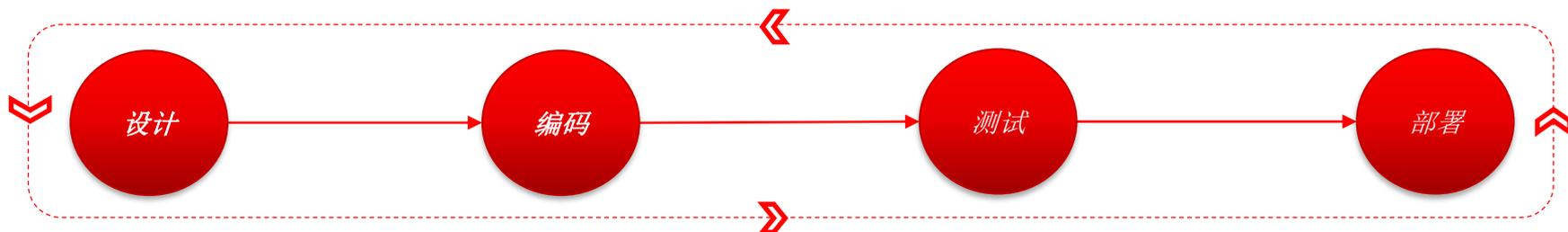


➤ 基础设施安全工具

- Nessus



SDLC 中的DevSecOps工具



工具				
输出	<ul style="list-style-type: none"> ❖ 风险评估 <ul style="list-style-type: none"> ▪ 范围评估 ▪ 安全设计评估 ▪ 威胁建模 ▪ 安全控制需求 	<ul style="list-style-type: none"> ❖ 同行审查 ❖ 工件签名 ❖ 静态代码检查 ❖ 第三方组件安全扫描 	<ul style="list-style-type: none"> ❖ 动态/交互式安全安全测 (DAST/IAST) ❖ UAT 测试用例 	<ul style="list-style-type: none"> ❖ 变更管理 <ul style="list-style-type: none"> ▪ 时间窗口 ▪ 发布审核流程 ▪ 工件比较 ❖ 发布记录 <ul style="list-style-type: none"> ▪ 变更证据 (审计)

手工

自动化

DevSecOps工具 -将CheckMarx集成到CICD流水线

The image shows two screenshots from the Jenkins interface. The left screenshot is the 'CxSAST Scan' configuration page. It includes fields for 'Checkmarx server URL', 'Credentials' (set to a service account), 'Checkmarx project name' (ta-service), 'Team' (CxServe), and 'Preset' (Checkmarx Default). The right screenshot shows the 'Build #17' report for 'Checkmarx Report'. It features a 'CxSAST Vulnerabilities Status' bar chart with a legend for 'Recent' (dark blue) and 'New' (light blue) vulnerabilities. Below the chart, it lists 'High - 0', 'Medium - 1', and 'Low - 18' vulnerabilities. A 'CxSAST Full Report' section is also visible, showing a table of vulnerabilities, including one 'Missing_HSTS_Header' with a severity of 'Low'.

- 首先, 在Jenkins服务器上安装Checkmarx Jenkins插件
- 有两种方法可以在Jenkins上配置Checkmarx扫描
 - Freestyle job:
 - 在"Build"部分选择"Execute Checkmarx Scan"项
 - 配置Checkmarx 服务器URL, 权限和源代码路径
 - Pipeline job
- Checkmarx 扫描结果报表可以在Jenkins界面上显示出来

The image shows the 'Generate Pipeline Script' tool in Jenkins. It displays a code block with a pipeline script snippet for the CxScanBuilder step. The script includes configuration for scan parameters such as 'fullScanCycle', 'generatePdReport', 'includeOpenSourceFolders', 'osaEnabled', 'password', 'serverUrl', and 'username'.

DevSecOps工具 - CheckMarx

\\apache-tomcat-7.0.67\webapps\examples\jsp\plugin\applet\Clock2.java

```
33 private static final long serialVersionUID = 1L;
34 Thread timer; // The thread that displays clock
35 int lastxs, lastys, lastxm,
36 lastym, lastxh, lastyh; // Dimensions used to draw hands
37 SimpleDateFormat formatter; // Formats the date displayed
38 String lastdate; // String to hold date displayed
39 Font clockFaceFont; // Font for number display on clock
40 Date currentDate; // Used to get date to display
41 Color handColor; // Color of main hands and dial
42 Color numberColor; // Color of second hand and numbers
43
44 @Override
45 public void init() {
46 lastxs = lastys = lastxm = lastym = lastxh = lastyh = 0;
47 formatter = new SimpleDateFormat ("EEE MMM dd hh:mm:ss yyyy", Loc
48 currentDate = new Date();
49 lastdate = formatter.format(currentDate);
50 clockFaceFont = new Font("Serif", Font.PLAIN, 14);
51 handColor = Color.blue;
52 numberColor = Color.darkGray;
53
54 try {
55 setBackground(new Color(Integer.parseInt(getParameter("bgcolor"),16)));
56 } catch (Exception E) { }
```

The dashboard displays the following information:

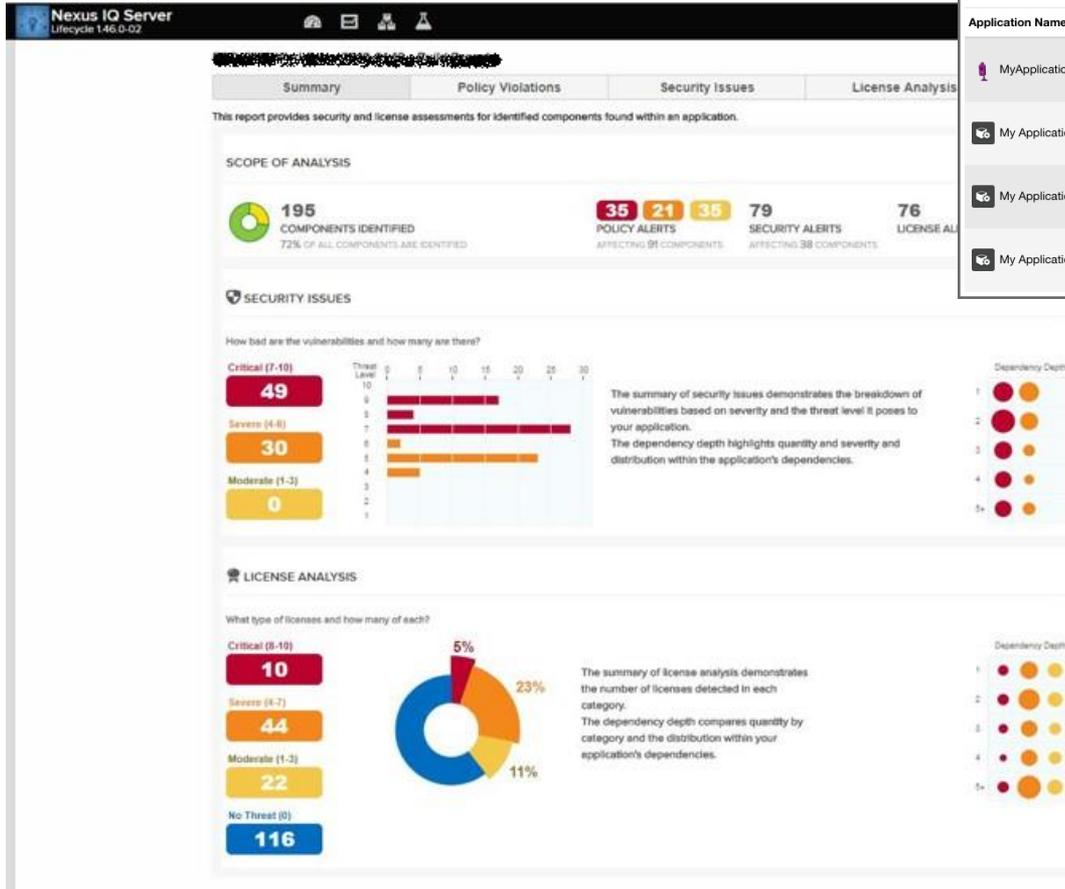
- Projects State / DocsProject4**: Navigation menu with Dashboard, Projects & Scans, Management, Users & Teams, Data Analysis, My Profile, AppSec Coach, and admin admin.
- Summary / OSA / Scans History**: Tabs for navigation.
- Current status (Public Scan on 3/16/2016 4:05:09 PM)**: Overview of scan results.
- SAST Vulnerabilities Status**: Summary of scan results showing 197 High, 275 Med, and 229 Low vulnerabilities. All are recurrent.
- SAST progress status**: Bar chart showing progress for High, Med, and Low severity levels. High: 197 Previous, 197 Solved, 197 Recurrent. Med: 275 Previous, 275 Solved, 275 Recurrent. Low: 229 Previous, 229 Solved, 229 Recurrent.
- Open Source Analysis (OSA)**: Last Scan on 6/1/2016 4:07:10 PM. 310 Libraries were analyzed. Vulnerabilities Score: High. 306 No Known Vulnerabilities, 4 Vulnerable & Outdated.

Scan Results	Severity	Results	Graph	AppSec Coach										
Java	High	1	Unchecke...	Reccurre...	\\apache-t...	Clock2.ja...	56	catch	\\apache-t...	Clock2.ja...	56	catch	To Verify	Info
Medium	2	Unchecke...	Reccurre...	\\apache-t...	Clock2.ja...	59	catch	\\apache-t...	Clock2.ja...	59	catch	To Verify	Info	
Low	3	Unchecke...	Reccurre...	\\apache-t...	Clock2.ja...	62	catch	\\apache-t...	Clock2.ja...	62	catch	To Verify	Info	
Info	4	Unchecke...	Reccurre...	\\apache-t...	Clock2.ja...	199	catch	\\apache-t...	Clock2.ja...	199	catch	To Verify	Info	

DevSecOps工具 – Sonatype Nexus IQ Server

- Nexus IQ Server 可以很容易地被集成到CICD流水线中去自动化安全漏洞扫描和报表流程
- Nexus IQ Server 产生的报表, 用于展示开源代码和插件中已经存在的信息安全和执照问题
- Nexus IQ Server 有一个可以把所有系统里的相关安全漏洞都能展示的全方位的中央报表
- Nexus IQ Server 将安全漏洞分类为三个等级 – 严重, 中等, 没有威胁

DevSecOps工具 – Sonatype Nexus IQ Server



Filter Applications

Application Name	Build Violations	Stage Release Violations	Release Violations	Contact	Organization
MyApplication	28 159 3	28 159 3			My Organization
My Application 4			6 5		My Organization 3
My Application 3	6 11 1				My Organization 7
My Application 2	6 11 1	6 11 1	6 11 1	John Smith	My Organization 4



信息安全编程大赛



- 为了增强开发团队信息安全意识, 一系列信息安全编程竞赛在公司各个地域的办公室里举办
 - 第一届信息安全编程大赛于2018年10月在印度区办公室成功举办
 - 第二届信息安全编程竞赛于2019年2月份在中国区办公室成功举办
- 这些活动通过让程序员参与富有竞争和协作的安全编程竞赛, 让程序员感受到编写安全代码的乐趣和提高其编写安全代码的技能
- 这些活动也给开发团队提供了一个可以评估自身信息安全成熟度水平的机会

DevSecOps成熟度模型

DecSecOps成熟度模型

等级	要求
六级	基本拥有专业信息安全专家的能力，可以指导和培训其他程序员进行安全开发
五级	在开发过程中可以一直应用信息安全技能和可以在系统上执行动态安全测试
四级	通过超过20个小时培训，在等级三的基础上能力进一步提高
三级	通过超过15个小时培训，在开发过程中具备超过OWASP前10大安全漏洞的能力
二级	通过大约10个小时培训，可以将OWASP前10大安全漏洞结合实际开发过程
一级	通过两个小时的培训，了解基本软件开发信息安全概念

- 作为一支成熟的DevSecOps团队，必须满足以下条件：
 - 工具扫描出来的高危漏洞在开发阶段需要被及时解决
 - 团队所有开发人员必须达到一级水平
 - 团队中的Security Champion至少达到三级水平，推荐达到五级水平
- 成熟的DevSecOps团队，会根据成熟度简化相应的传统信息安全扫描和审核流程

Gdevops

全球敏捷运维峰会

THANK YOU!

