面向DevSecOps的代码安全保障体系

董国伟

360企业安全集团 代码安全事业部

dongguowei@360.net

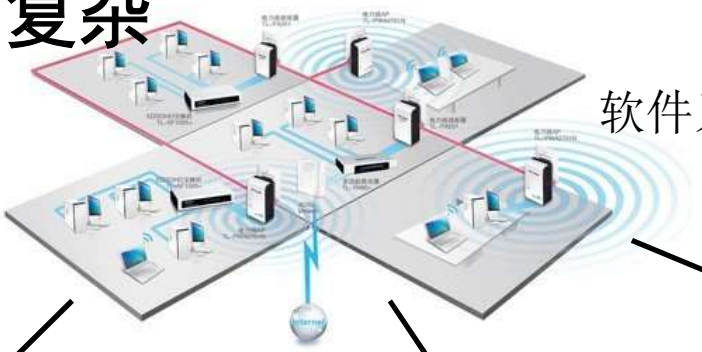
2018/11/27

CONTENTS

1、软件已成为核心基础设施

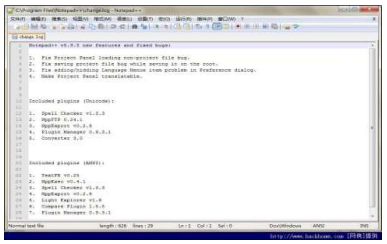
- 软件已应用于生产生活方方面面
  - 国家关键信息基础设施
  - 企业生产管理信息化系统
  - 软件安全直接关系国家、民生、经济、生产等安全
- 物联网车联网区块链等的应用
  - 万物互联
  - 软件用于设备控制
  - 软件安全关系到人身、财产、信息数据等的安全
- 代码是软件的原始基础形态
  - 设计时形态：架构、接口、图形等软件（代码）轮廓
  - 开发时形态：源代码、第三方组件（开源、其他渠道提供的代码等）
  - 运行时形态：可执行代码、配置脚本代码、集成代码等

2、软件供应链更加复杂

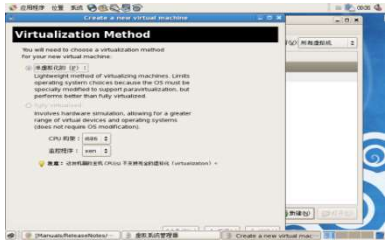


软件系统

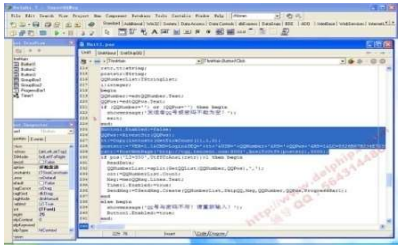
自主开发的



商业购买的



开源组件

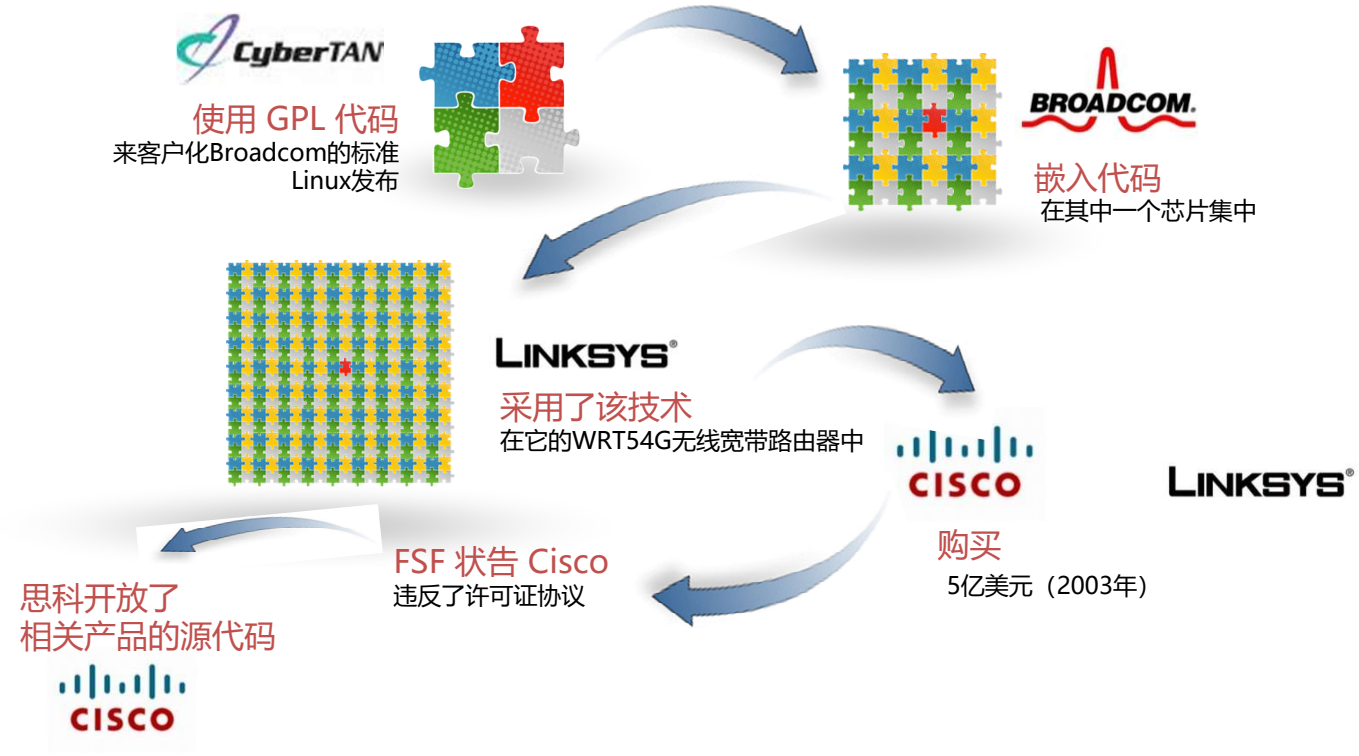


外包开发的

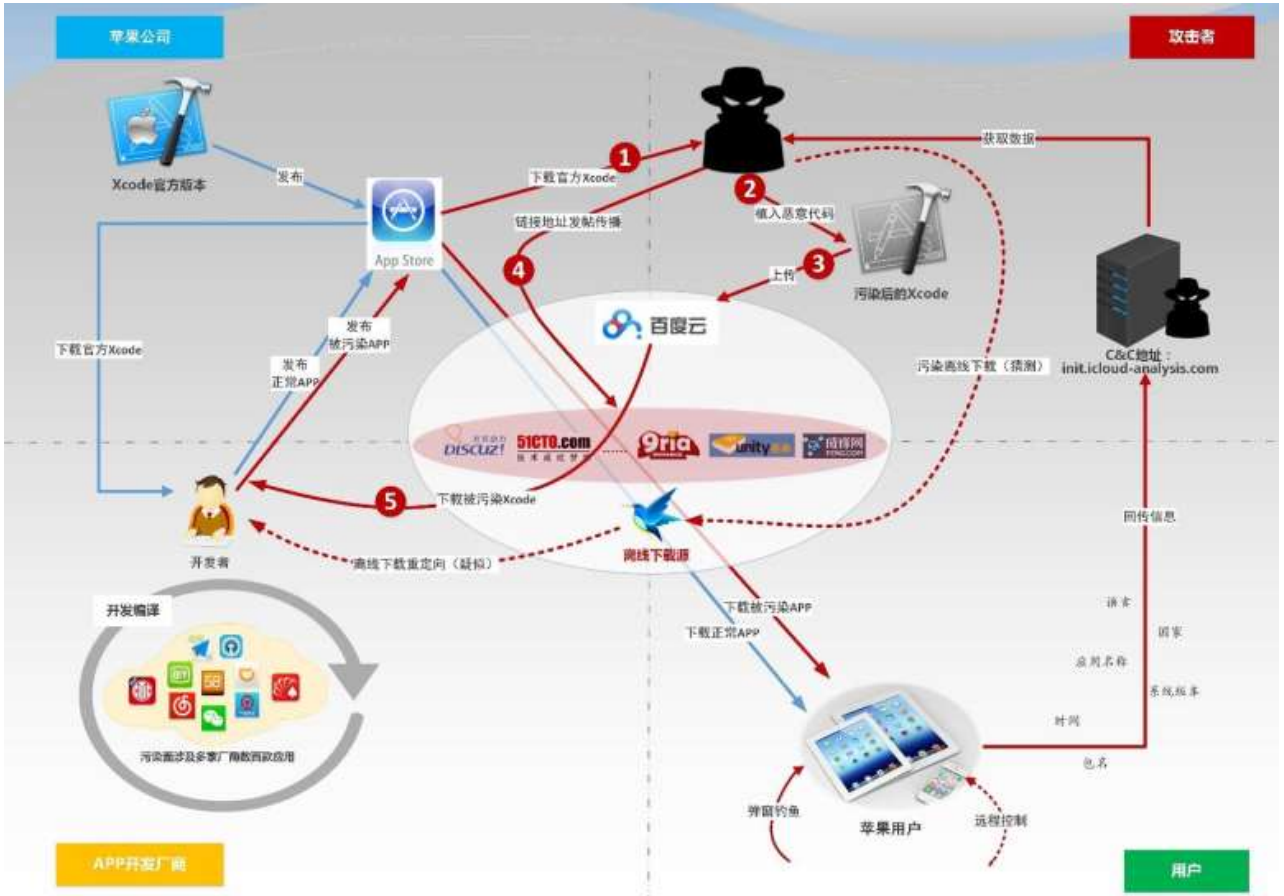
VeraCode：

- ✓ 30%-70%包含自主开发软件的代码也含有第三方代码
- ✓ 多以开源组件、商业或外包共享库/组件的形式存在

（1）思科违反GPL许可证问题



（2）苹果XCodeGhost隐患

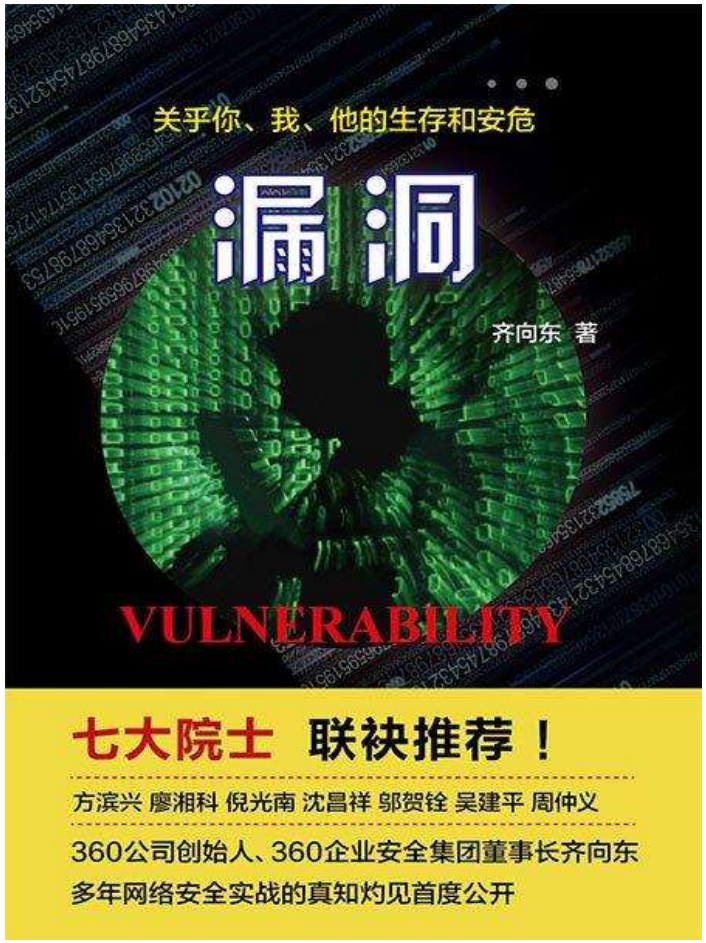


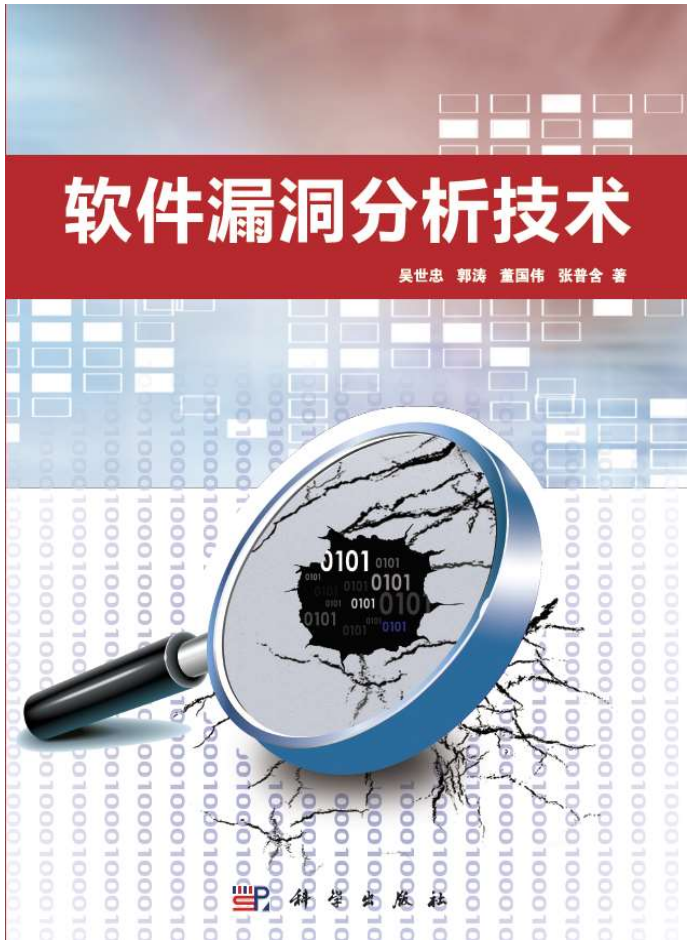
3、软件漏洞依然高发

数据来源：CNNVD

重大安全事件背后均有漏洞身影

事件类型	典型事例	漏洞
系统故障	航天器系统故障	代码错误
计算机病毒	WannaCry病毒	MS17-010 (NSA永恒之蓝EternalBlue漏洞)
关键基础设施安全问题	震网病毒	MS10-046/MS10-061/MS08-067/提权漏洞
	乌克兰电力系统攻击	CVE-2014-4114 (Windows任意代码执行漏洞)
	2017年6月数百万德国网民断网	CVE-2017-9765 (路由器劫持漏洞)
	2017年9月，十亿蓝牙设备可在用户未知情况下被打开并接管设备	信息泄露、代码远程执行、中间人攻击等类型的8个漏洞
重要数据信息泄露	CSDN六百万用户信息外泄	SQL注入漏洞 (获得数据库访问权限)
	美知名信用机构Equifax遭攻击，约1.43亿名用户数据遭泄露	CVE-2017-5638 (Apache Strusts2 远程代码执行漏洞，基于Struts2的网络服务器被提权)
	华住旗下酒店上亿条用户数据在暗网售卖	硬编码漏洞 (程序员将主数据库用户密码写到代码中，然后将代码上传到GitHub，并设置为公开访问权限)

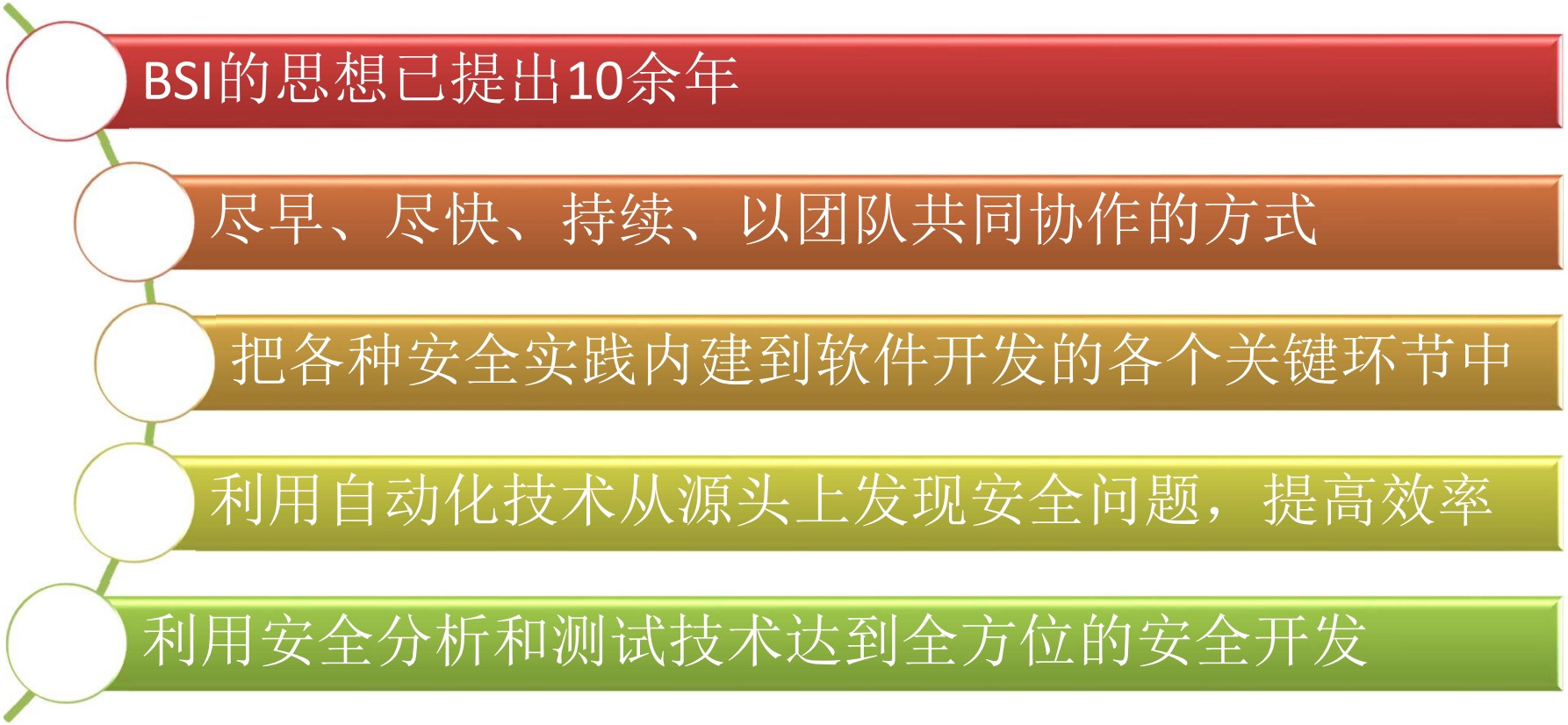

关乎你、我、他的生存和安危
漏洞
齐向东 著
VULNERABILITY
七大院士 联袂推荐！
方滨兴 廖湘科 倪光南 沈昌祥 邬贺铨 吴建平 周仲义
360公司创始人、360企业安全集团董事长齐向东
多年网络安全实战的真知灼见首度公开


软件漏洞分析技术
吴世忠 郭涛 董国伟 张普含 著
科学出版社

4、传统安全防护方法存在不足

- ✓ 交付运行后的被动防御手段
- ✓ 无法从源头上发现安全问题
- ✓ 未消减软件自身的安全问题
- ✓ 增加了软件安全的修复成本

1、内建安全BSI（Build Security In）



2、微软安全开发生命周期SDL

- BSI的成功实践，通过对软件工程的控制，从漏洞产生的根源上解决问题
- 主要针对瀑布型开发方式，多用于有较为成熟软件工程经验的开发商
- 安全部门或人员的角色定位相当于监管方，如项目须由安全部门审核完成后才能发布

3、应对开发和运维团队协同的DevOps

- DevOps通过自动化“软件交付”和“架构变更”的流程，使得构建、测试、发布软件更加快捷、频繁和可靠
  - 自动化、快速化、最小化、透明化、统一化

基础设施即代码

- 将重复的事情使用自动化脚本或软件来实现
- Docker（容器化）、Jenkins（持续集成）、Puppet（基础架构构建）、Vagrant（虚拟化平台）等

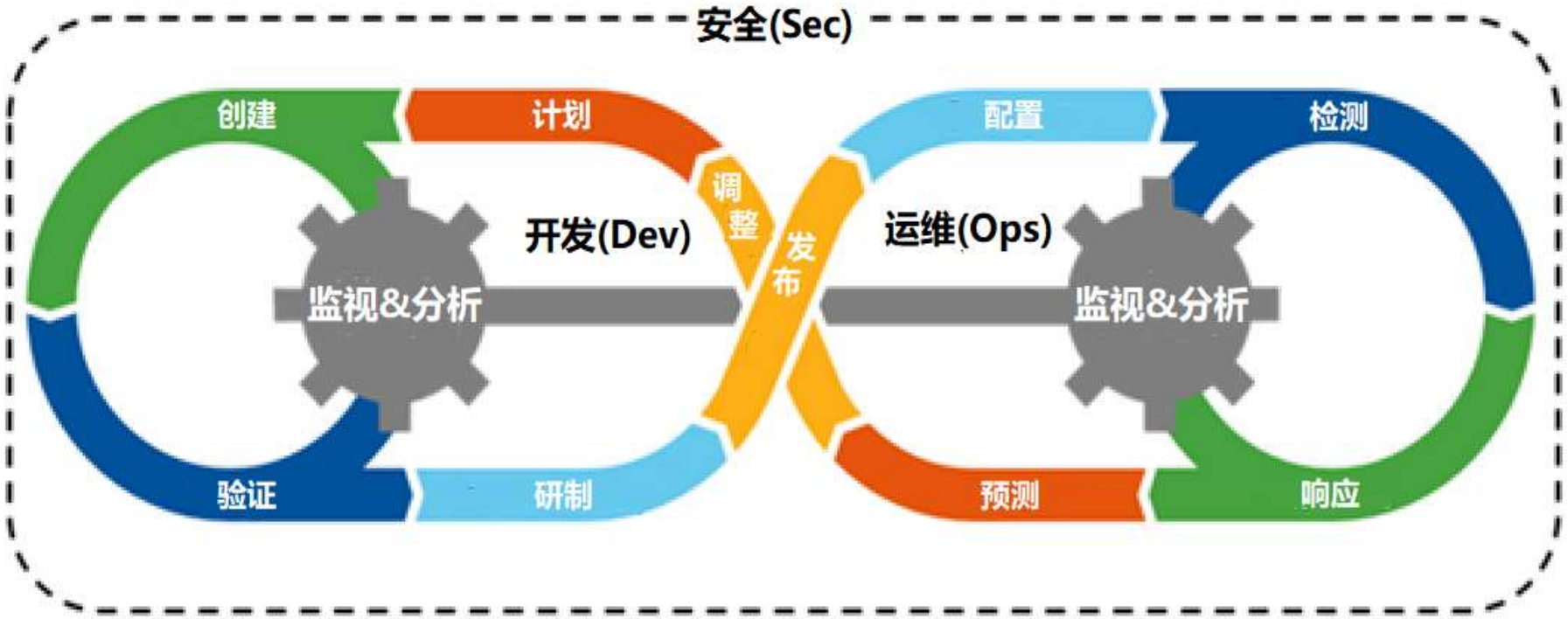
持续交付

- 持续交付在生产环境发布可靠软件并交付给用户，持续部署则不一定交付给用户
- 要做到高效交付可靠的软件，需要尽可能减少修复时间和产品上线时间
- 部署可以有多种方式，比如蓝绿部署、金丝雀部署等

协同工作

- 开发应该把运维角色理解成软件的另一个用户群体
- 自动化(减少不必要的协作)、小范围(每次修改的内容不宜过多，减少发布的风险)、统一信息集散地(如wiki，让双方能够共享信息)、标准化协作工具(比如jenkins)

4、DevSecOps



（1）DevSecOps在代码安全方面要求归纳

- 总体和管理流程
  - 安全控制可编程和自动化（自动化）
  - 监控一切，为快速检测和响应构建架构（快速化）
  - 将安全整合到开发部署流水线中（透明化、统一化）
- 受攻击面方面
  - 为所有应用程序实现简单的风险和威胁建模（快速发现明显的安全问题）
- 源代码和可执行代码
  - 扫描自定义代码、应用程序和API（自动化发现代码缺陷）
  - 在开发过程中扫描开源软件问题（自动化尽早发现软件供应链安全问题）
  - 自定义或开源代码外的漏洞扫描并修正配置（自动化尽早发现其他安全问题）

（2）DevSecOps本质需求

- 针对不同开发阶段的软件对象形式
- 最快方式发现其中安全问题并修复
- 使用方法尽量是自动化和透明化的

一、软件安全新挑战

二、DevSecOps新要求

三、代码安全保障实践

四、未来发展趋势展望

1、安全技能培训

- ✓ **系统讲授代码安全保障体系的构建**
- ✓ **软件安全开发、典型漏洞机理分析、缺陷测试和漏洞挖掘等具体实践**
- ✓ **使用户全面了解代码安全保障基本理念和主要实现技术**

2、需求设计阶段-安全需求分析

依据

- 软件系统场景
- 国家、行业监管要求
- 系统常见的安全攻击手段
- ……

明确

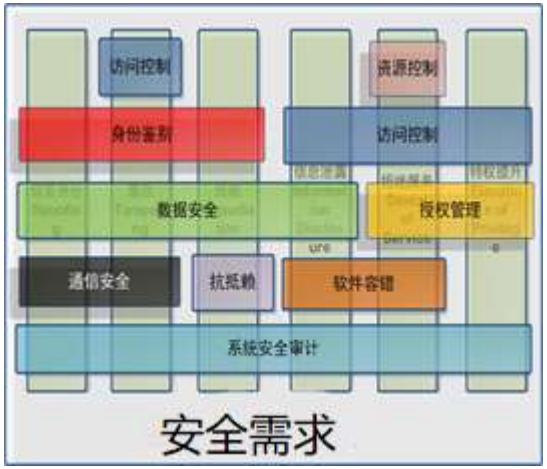
- 数据传输的安全性要求
- 数据传输的保密性要求
- 数据存储的安全性要求
- 接口安全
- 界面信息屏蔽的安全要求
- ……

2、需求设计阶段-需求威胁消减

- 分析安全需求涉及的通信、数据传输、数据交互、本地数据存储、安全合规要求等因素
- 找出安全需求所面临的威胁，并结合OWASP TOP10等安全问题，给出威胁解决手段

场景分类	安全威胁	安全威胁消减措施
登录认证类	（1） 密码长度过短，容易被猜测爆破 （2） 身份信息被仿冒 （3） 用户密码明文传输，容易被猜测	● 检查长度、复杂度、密码与当前密码是否重叠等措施 ● 增加用户的双因子增强型认证，例如 AD 域 ● 密码通用采用 Hash 码传输 ● 登录界面增加安全验证码
	（1） 接口调用前被未授权访问 （2） 接口可被任意访问，内容被人篡改	● 接口调用之前需要做身份认证，并且在调动过程中都要传输凭证码 ● 对于非终端用户访问接口，增加接口访问的黑白名单 ● 特定系统可做 IP 与用户的绑定
数据传输类	（1） 数据传输中数据明文被人截获，造成信息泄露	● 采用 SSL VPN 支持外部的远程访问走安全通道 ● 系统自身支持 HTTPS 协议
	（1） 数据未做完整性校验，被人非法篡改	● 采用数据完整性校验机制，防范数据被篡改的风险
访问控制类	（1） 用户数据被越权访问 （2） 数据未控制访问范围	● 应设置安全策略控制用户访问指定资源，遵守最小授权原则，建立生产系统关键账户与权限的关系，对重要资源设置敏感标记同时严格控制用户访问
安全审计类	（1） 业务操作没有记录，相应的操作无法溯源 （2） 破坏者作案后，将操作记录删除，导致无法追踪	● 应提供覆盖到每个用户的安全审计功能 ● 定期备份审计记录，保存时间不少于半年 ● 不提供删除、修改或覆盖审计记录的功能
交易安全类	（1） 交易发生后，客户抵赖未发起交易 （2） 交易被篡改导致资金损失	● 采用 PKI 体系的数字签名技术对交易原文进行签名，防范用户抵赖以及被篡改的风险

2、需求设计阶段-安全功能设计



1.身份鉴别	1.1密码支持
	1.2账户策略
	1.3辅助安全设备
2.授权管理	2.1 功能授权
3.访问控制	3.1 系统内访问控制
	3.2 系统外访问控制
4.系统安全审计	4.1 WEB应用访问日志完备性
	4.2 用户认证日志完备性
	4.3 应用操作日志完备性
	4.4 后台日志完备性
	4.5 日志信息安全存储
5.通信安全	5.1 通讯协议
	5.2 通讯安全认证
6.数据安全	6.1 用户数据的输入与输出
	6.2 用户数据保密性
	6.3 用户数据完整性鉴别
	6.4 用户数据的存储
7.抗抵赖	7.1 原发抗抵赖
	7.2 接收抗抵赖
	7.3 数字证书
8.软件容错	8.1 降级容错
	8.2 受限容错
9.资源控制	9.1 内部资源控制
	9.2 外部资源控制

规则编号	基本规则设计说明
B9010001	系统的软件配置文件中，认证信息需采用加密方式存储，如认证用户名、密码等信息
B9010002	系统的软件配置文件中，应删除所有的信息注释、配置选项说明、解释信息
B9010003	系统的软件代码中应移除所有测试代码
B9010004	系统的软件源代码应存储于安全的介质中，通过安全策略保证源代码安全
B9010005	系统的软件源代码版本控制过程，应保证采用最新的稳定版本
B9010006	系统使用过程中的资源建立，应通过身份鉴别后方可创建
B9010007	系统使用过程中的资源建立，应避免使用有规律的、简单的规则，如资源命名的顺序排列

规则编号	增强规则设计说明
E1010001	软键盘被点击过程，每个被点击键无明显状态变化
E1010002	密码安全控件程序应能够进行自身的完整性校验
E1010003	密码安全控件程序应能够防范调试、跟踪等黑客行为
E1010004	密码复杂度应大写字母、小写字母、数字、特殊字符这四类中至少两类
E1010005	密码不能是等差数字
E1010006	修改密码时，新密码不能是旧密码的回文
E1010007	修改密码时，新密码不能是旧密码改了大小写
E1010008	修改密码时，新密码不能是旧密码的一个循环
E1010009	不容许重复使用相同密码，对密码历史进行检测

2、需求设计阶段-安全需求设计自动化

- 以威胁资源为基础
- 输入应用场景和需求数据
  - 场景问答
  - 功能勾选
  - 人机交互
- 自动输出
  - 安全需求
  - 安全设计
  - 安全开发建议
  - ……

3、供应链分析阶段

- 代码基准库构建及特征提取
- 组件成分分析
- 历史漏洞分析
- 知识产权风险分析

3、供应链分析阶段-开源项目检测计划

- 来自NVD的数据统计：已公开的开源软件漏洞超过2.8万个。
- 美国：自2006年开始，美国国土安全部资助Coverity公司开展“开源项目检测计划”，目前已检测7000多款开源软件。
- 2015年初，360代码卫士基于自身产品能力发起了国内最大的“开源项目检测计划”，该计划目前已检测2230个开源项目，测试代码2.6亿行，积累了大量的源代码安全缺陷检测基础数据。

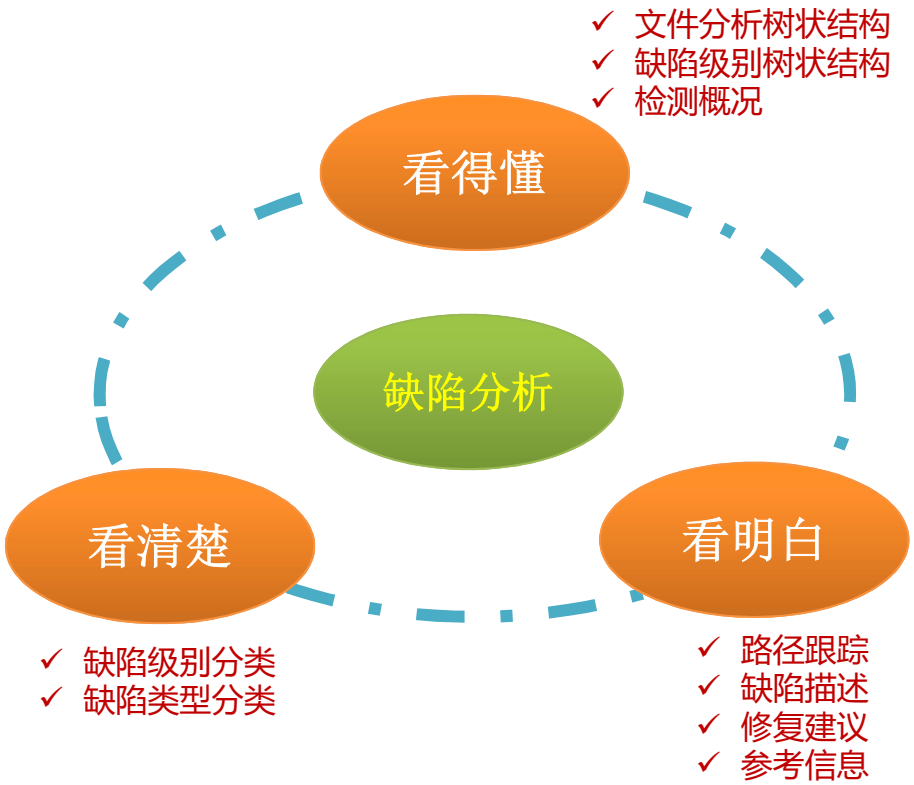
4、自主开发阶段-编程规范定制

- CERT C/C++/Java
- MISRA C/C++

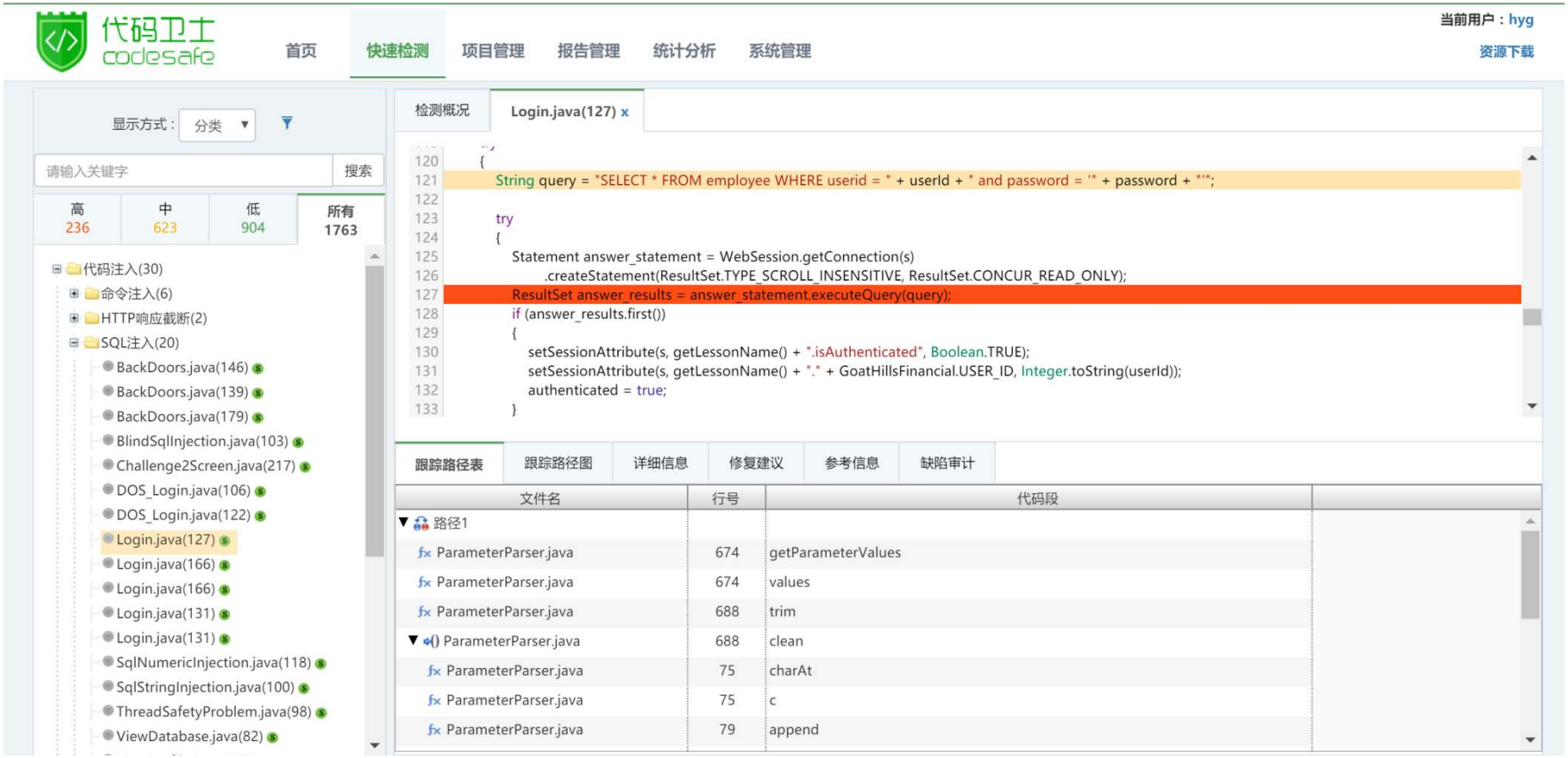
- 功能需求
- 行业标准
- 公司策略
- 数据敏感级别

4、自主开发阶段-源代码合规性和缺陷检测（360代码卫士能力）

4、自主开发阶段-可视化缺陷管控



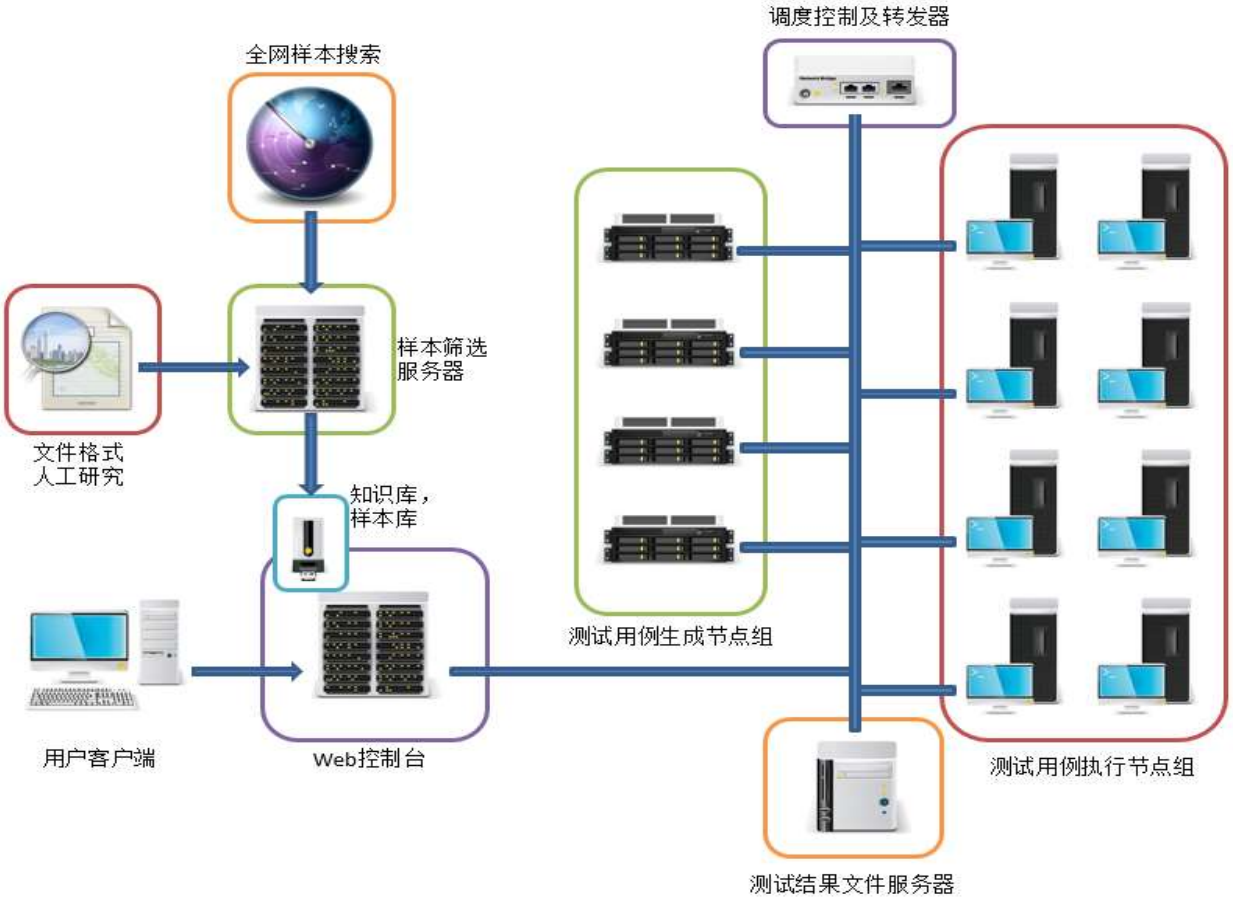

代码卫士
codesafe
首页
快速检测
项目管理
报告管理
统计分析
系统管理
当前用户：hyg
资源下载
源代码类型信息
.java
3.77 万行
.jsp
3558 行
.xml
1293 行
.js
662 行
.html
3.14 万行
.properties
115 行
缺陷总数
1763 个
缺陷等级统计
高
236 个
中
623 个
低
904 个
高
中
低
缺陷分类统计
代码质量
异常处理
跨站脚本
资源管理
密码管理
输入验证
代码注入
292
221
163
110
54
30
代码质量
数量：871


代码卫士
codesafe
首页
快速检测
项目管理
报告管理
统计分析
系统管理
当前用户：hyg
资源下载
显示方式：
分类
请输入关键字
搜索
高 236
中 623
低 904
所有 1763
代码注入(30)
命令注入(6)
HTTP响应截断(2)
SQL注入(20)
BackDoors.java(146)
BackDoors.java(139)
BackDoors.java(179)
BlindSqlInjection.java(103)
Challenge2Screen.java(217)
DOS_Login.java(106)
DOS_Login.java(122)
Login.java(127)
Login.java(166)
Login.java(166)
Login.java(131)
Login.java(131)
SqlNumericInjection.java(118)
SqlStringInjection.java(100)
ThreadSafetyProblem.java(98)
ViewDatabase.java(82)
检测概况
Login.java(127) x
120 {
121 String query = "SELECT * FROM employee WHERE userid = " + userId + " and password = '" + password + "'";
122
123 try
124 {
125 Statement answer_statement = WebSession.getConnection(s)
126 .createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
127 ResultSet answer_results = answer_statement.executeQuery(query);
128 if (answer_results.first())
129 {
130 setSessionAttribute(s, getLessonName() + ".isAuthenticated", Boolean.TRUE);
131 setSessionAttribute(s, getLessonName() + "." + GoatHillsFinancial.USER_ID, Integer.toString(userId));
132 authenticated = true;
133 }
跟踪路径表
跟踪路径图
详细信息
修复建议
参考信息
缺陷审计
文件名
行号
代码段
路径1
ParameterParser.java 674 getParameterValues
ParameterParser.java 674 values
ParameterParser.java 688 trim
ParameterParser.java 688 clean
ParameterParser.java 75 charAt
ParameterParser.java 75 c
ParameterParser.java 79 append

4、自主开发阶段-平台技术优势

- 360企业安全集团旗下，专注于软件源代码与可执行码的漏洞分析技术研究和产品开发
- 国家发改委“大数据协同安全技术国家工程实验室-代码安全实验室”的承担单位，负责国家工程实验室在代码安全、漏洞分析方面的研究工作规划和组织实施

5、软件测试阶段-模糊测试



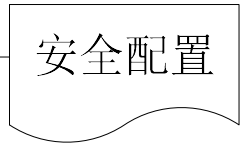
5、软件测试阶段-渗透测试

渗透测试指导原则：

- OWASP TEST GUIDE
- WASC威胁分类

渗透测试工具包：

- BT3/4/5
- Metasploit
- CANVAS



6、开发流程无缝对接功能

- ✓ 支持从SVN、Git等代码管理系统中获取源代码进行自动化周期检测
- ✓ 支持检测结果与Bugzilla、Jira等缺陷管理系统进行整合

- 场景需求及描述
  - 以最小代价与企业原有开发流程进行无缝整合
    - IDE整合（插件）
    - 代码库整合
    - 构建工具整合
    - 缺陷管理系统整合等
  - 场景描述
    - 开发人员在管理平台中配置好任务计划（例如晚上12点以后执行检测任务）
    - 平台会根据任务计划的设定，自动从SVN/Git中获取代码进行检测
    - 开发人员第二天上班时查看和审计检测结果

一、软件安全新挑战

二、DevSecOps新要求

三、代码安全保障实践

四、未来发展趋势展望

几点不成熟的想法

- 基于代码片段的安全缺陷检测
- 基于组件依赖关系的安全影响分析
- 开源代码库的缺陷和漏洞分析
- 基于云的代码缺陷检测服务
- 基于机器学习的缺陷和漏洞分析
- ……

谢 谢!

360企业安全集团 | 安全第一 就用360

b.360.cn