

企业软件开发安全 (SDLC) 实践

2017. 3. 8



错过乱

泽西岛



扫一扫上面的二维码图案，加我微信

国内信息安全事件



2014年3月，乌云平台连续披露了两个携程网安全漏洞，由于携程开启了用户支付服务接口的调试功能，导致携程安全支付日志可被任意读取，日志泄露包括持卡人姓名、身份证、银行卡类别、银行卡号、CVV码等。

漏洞一：业务权限问题

问题原因：处理用户支付服务接口开启调试功能，使所有向银行验证持卡所有者接口传输的数据包均直接保存在本地服务器

漏洞二：系统权限问题

问题原因：支付日志存储服务存在安全漏洞，导致越权遍历目录，获得日志信息。

漏洞三：审计监控问题

问题原因：违反银联规定将用户信用卡信息留存于本地，并写入日志文件，导致黑产通过日志获得用户信用卡信息。



2015年10月，补天漏洞响应平台再次爆出中国电信某系统的重大漏洞。通过该漏洞可以查询上亿用户信息，涉及姓名、证件号、余额，并可以进行任意金额充值、销户、换卡等操作。10月29日上午10点，该漏洞已得到中国电信厂商确认。

漏洞一：弱口令问题

问题原因：账号口令未进行强度限制，导致黑客轻易猜出口令进入系统。

漏洞二：越权访问问题

问题原因：登录系统后，可越权访问。黑客借此可以收现更多高危漏洞，访问敏感数据。。

漏洞三：信息泄露问题

问题原因：敏感信息存储时未进行严格访问控制，导致越权访问后直接获得用户隐私数据。

国内信息安全事件



云南机场OA客户端应用被破解，泄漏大量机场员工的隐私数据和企业内部的邮件数据，甚至包括重要领导人的航班信息。

漏洞一：应用破解问题

问题表现：通过调试手段获知到应用执行目录

问题原因：未采用应用加固技术，应用代码未加密保护。

漏洞二：数据未加密存储问题

问题表现：本地敏感数据未加密

问题原因：客户端存储的企业员工的电话、邮箱、住址等信息和邮件信息未做加密处理，能够被非法读走。



2013-2015年国内知名连锁酒店（锦江之星、速八..）及高端酒店（万豪、喜来登、洲际..）等网站存在高危漏洞，大量客户开房信息泄露姓名、身份证、手机号、开房时间、退房时间、房型、家庭住址、信用卡后四位、信用卡截止日期及邮件等

漏洞一：外包商问题

问题原因：都采用同一个外包商开发的酒店管理系统，该系统存在众多高危漏洞被黑客利用。

漏洞二：数据明文传输

问题原因：用户敏感数据未做有效保护，用户认证、传输采用明文方式实现

信息安全事件造成的影响



品牌或公
司形象受
到影响



合规性要
求不符合

公司财产
的损失



用户敏
感数据
泄露



安全漏洞
导致法律
风险



公司经营
风险

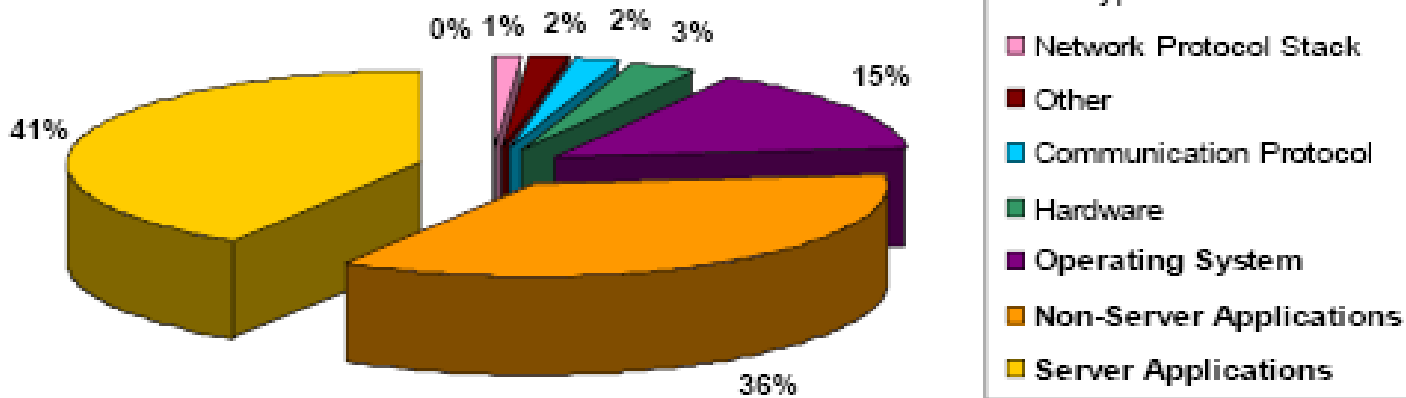
客户的流
失

为什么软件安全重要？



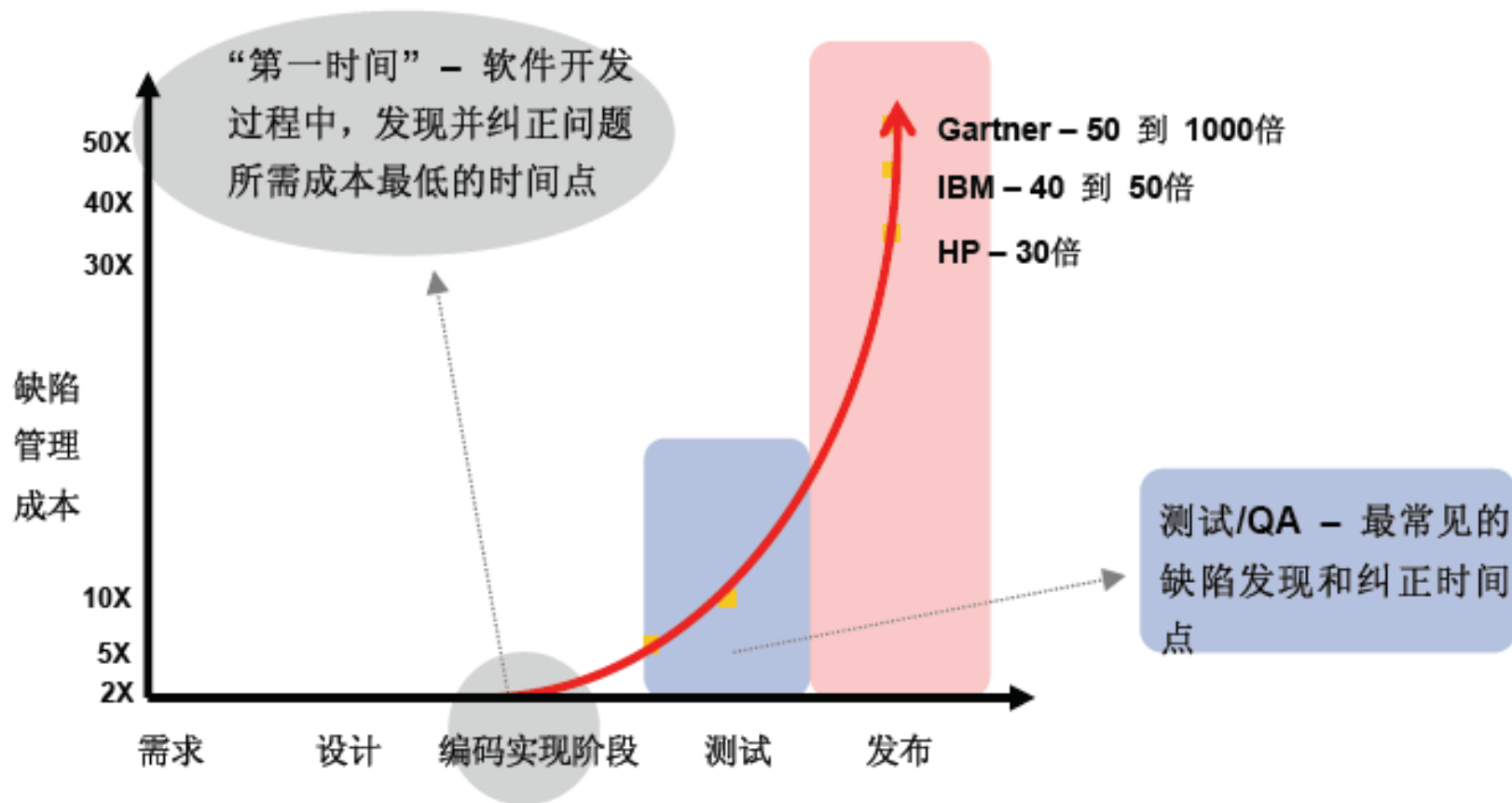
- 根据NIST等权威机构的报告，超过90%的黑客安全事故都发生在软件应用本身，而不是在网络。

92% of reported vulnerabilities are in applications, not networks

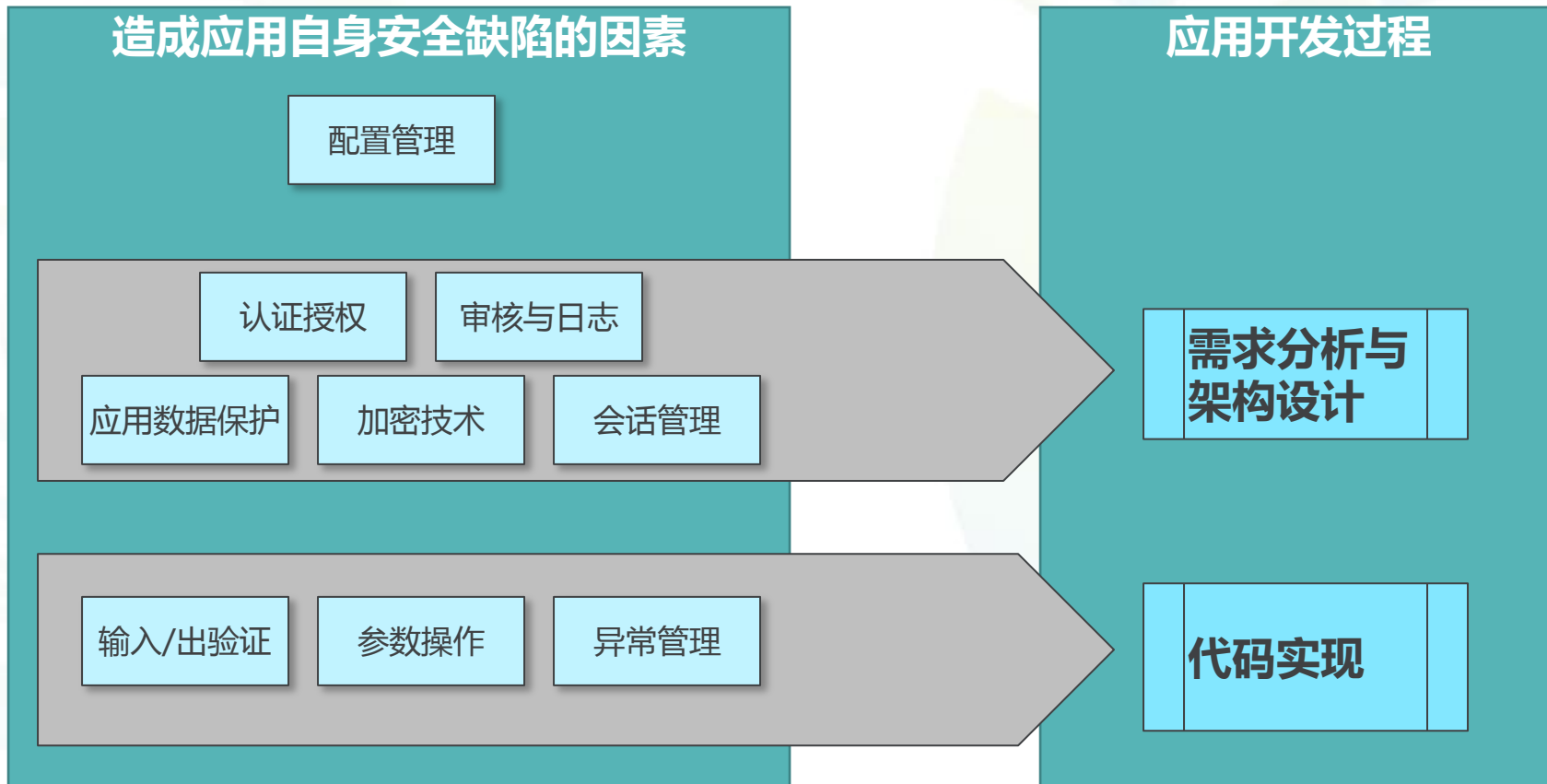


Source: NIST

安全修复成本



软件自身安全问题在于应用开发



Web应用开发安全涉及内容



web应用程序安全

身份
鉴别

访问
控制

安全
审计

剩余信
息保护

完
整
性

通
信
保
密
性

抗
抵
赖

软件
容错

资源
控制

标识
管理

鉴别
管理

会话
管理

权限
管理

标记
管理

后台
访问

审计
范围

审计
内容

审计
保护

鉴别
信息

应用
数据

版权
注释

数据
完整

系统
完整

鉴别
信息

应用
数据

原发
证据

接收
证据

输入
验证

输出
验证

状态
管理

操作
回退

系统
会话

帐户
进程


系统
服务

信息系统安全现状



IT系统生命周期与信息安全

- 看得不够远，是因为站得不够高
- 做得不够好，是因为做得不够早



看起来做的很多
却始终逃不出这么
个小小的圈子

需求

设计

开发

上线

运维

响应

© 2006 R.L. McNish. All Rights Reserved.

其实信息安全可以做的更多

安全发开业内最佳实践



- **Microsoft SDL (方案)**
- **IBM软件开发生命周期和Web应用安全解决方案 (产品)**
- **FORTIFY SOFTWARE SECURITY CENTER (产品)**
- **Cisco CSDL Process (官方文档)**
- **Oracle Secure Development (官方文档)**
- **(ISC)十个安全开发软件最佳实践 (公开文档)**
- **CLASP — Comprehensive, Lightweight Application Security Process**
- **OWASP Top 10 2013 Web应用安全风险**
- **WASC (Web application security consortium) Threat Classification**
- **OWASP软件保证成熟度模型(SAMM)**
- **Common Weakness Enumeration (CWE)/SANS Top 25**
- **Common Vulnerabilities and Exposures (CVE)**
- **美国国家安全局(NAS)发布的SSE-CMM V2.0模型**
- **FIRST通用弱点评价体系(CVSS) V2.10**
- **PMI项目管理知识体系(PMBOK)**

微软软件安全开发实践—SDL



Inception

- Security advisor assigned
- Ensure security milestones understood
- Identify security requirements

项目启动阶段

- 任命安全顾问
- 设置清晰的安全里程碑
- 识别安全需求

Design & Threat Modeling

- Design guidelines documented
- Threat models produced
- Security architecture documented
- Threat model and design review completed
- Ship criteria agreed to

设计和威胁模型

- 文档化的设计指南
- 威胁模型分析
- 文档化安全架构
- 审核威胁模型和安全设计

Guidelines & Best Practices

- Coding and test standards followed
- Test plans developed and executed (including fuzz testing)
- Tools used (Code analysis)
- 指南和最佳实践
- 遵循标准编码和测试
- 测试计划开发和执行
- 代码分析工具的使用

Final Security Review (FSR)

- Threat models reviewed
- Unfixed bugs reviewed
- New bugs reviewed (postmortem)
- Penetration testing completed
- Documentation archived
- 最终安全审核
- 重新审核威胁模型
- 审核未修正的BUG
- 审核新BUG
- 完成渗透测试
- 完成相关文档

RTM & Deployment

- Signoff by security team
- 生产和部署
- 安全团队签名确认

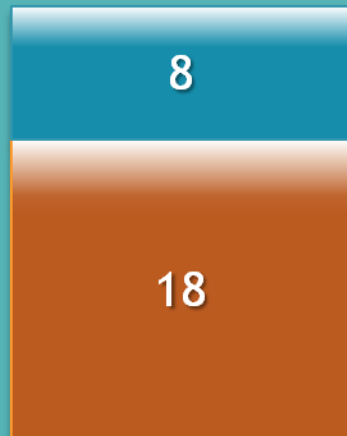
Security Response Feedback

- Tools/processes evaluated
- Postmortems completed

IE采用SDL后的效果

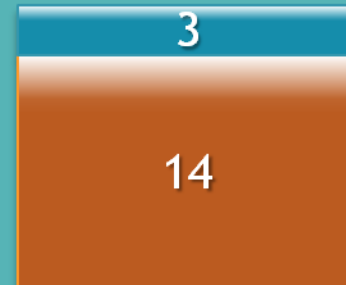
正式发布后12个月内修复的漏洞总数

■ Medium ■ High



Internet Explorer 6

采用SDL之前



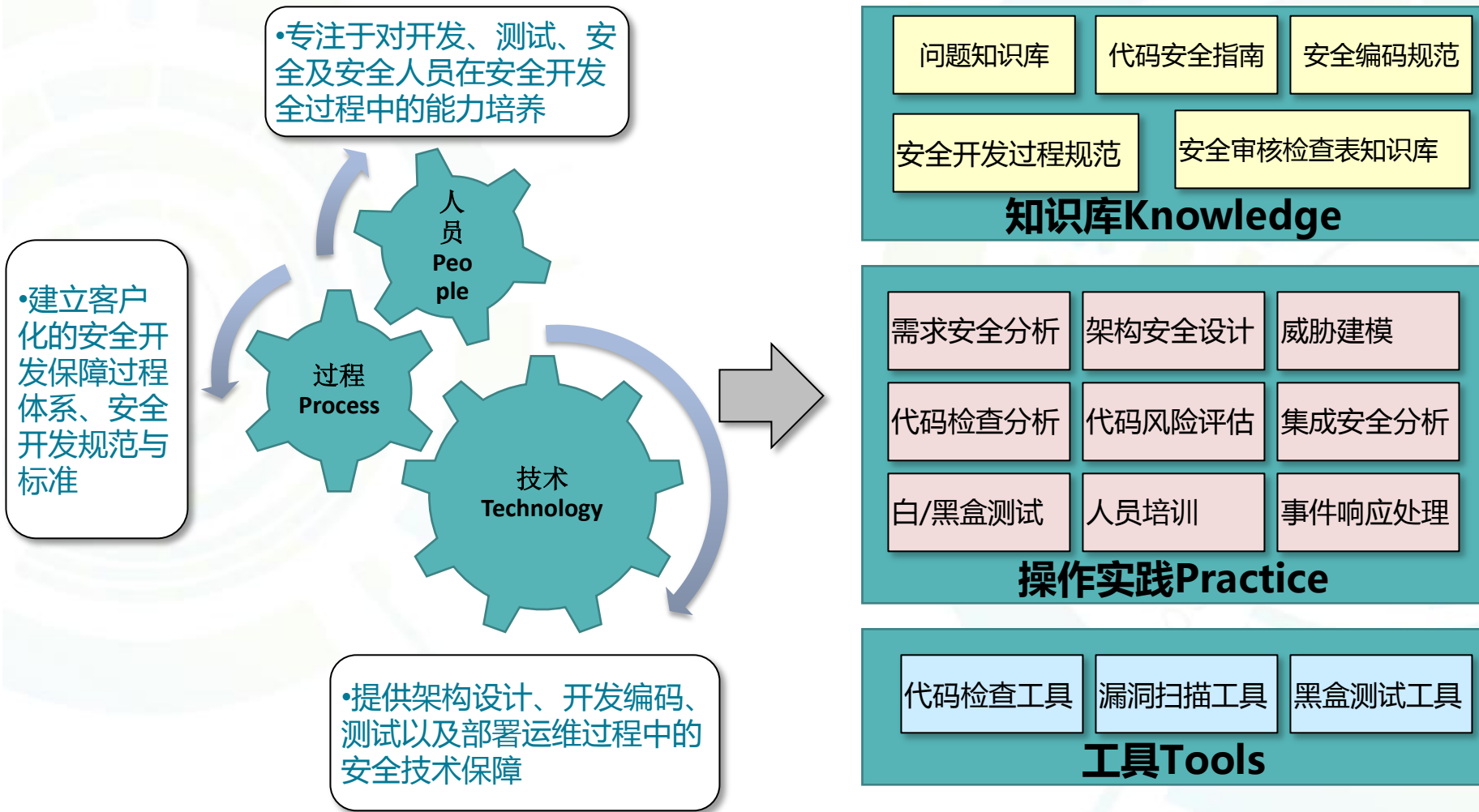
Internet Explorer 7

采用SDL之后

漏洞总数降低了35%

高危漏洞数降低了63%

应用开发安全解决方案



软件应用安全开发生命周期-SDLC



安全培训

需求分析

安全设计

安全开发

安全测试

安全上线

应急响应

- 安全意识&流程
- 业务安全需求分析
- **威胁建模**
- **安全开发规范**
- 白盒测试
- **上线安全检查**
- 漏洞预警
- **安全开发技能**
- 确认安全需求
- 攻击面分析
- Java开发规范
- 黑盒测试
- 配置安全
- **漏洞发现及响应**
- 行业安全实践
- 安全设计规范
- PHP开发规范
- 社工
- 上线Checklist
- 防护手段（设备）
- HTML5开发规范
- 安全Checklist
- 应急服务

项目安全评审&需求分析

安全培训&安全设计

上线前

安全测试

上线后

线上防御

响应

运营

SDLC-安全培训



安全培训



需求阶段



设计阶段



开发阶段



测试阶段



上线阶段



响应阶段

- 安全开发培训阶段主要是通过安全开发技能培训，帮助其了解安全攻防现状、安全编程基本技能及软件安全最新趋势，为后续软件安全开发生命周期管理落地实施奠定基础：

安全设计：减小攻击面、最小授权、威胁建模等；

安全编码：验证输入、避免缓存溢出、程序内部安全、安全调用组件、程序编写编译、使用安全、统一的编码或转义方式等；

安全测试：黑盒测试、白盒测试、灰盒测试等；

安全开发最佳实践：（OWASP）Top10漏洞风险分析及防范措施；

安全培训-培训大纲



培训大纲	培训课程
软件开发简介	安全开发历史发展；
	安全开发的问题及其必要性；
	安全开发的好处以及成效。
软件安全开发模型及安全开发框架	安全软件开发生命周期；
	安全软件开发模型（微软SDL、BSI系列模型、CLASP和SAMM等）；
	安全开发框架。
软件安全开发设计	软件安全设计的重要性及基本原则；
	隐私开发设计；
	理解受攻击面概念和常用减少受攻击面的保护措施；
	威胁建模的概念和目的；
	威胁建模的关键因素及作用。
安全编码	掌握通用安全编程准则，包括验证输入、避免缓存溢出、程序内部安全、安全调用组件、程序编写编译、使用安全、统一的编码或转义方式等；
	理解编码时禁止使用的函数；
	常见代码安全问题及处置办法举例；
	源代码静态分析工具；
	源代码审核（工具使用及人工分析）。
开发安全编码实践	输入数据验证过程、以及转换和传输过程中的安全注意事项；
	对象声明与初始化过程中安全陷阱与防范方法；
	数字类型变量声明和操作中的各种安全问题及其防范方法；
	异常处理、共享变量访问和复合操作原子性、线程安全当中的问题及防范方法；
	Java运行平台和运行环境当中的安全问题及防范措施。

软件安全开发测试	白盒测试；
	黑盒测试；
	灰盒测试；
	模糊测试。
安全开发最佳实践（OWASP）Top10漏洞风险分析及防范措施）	Injection(注入)；
	Broken Authentication and session Manager（失效的认证和会话管理）；
	XSS（跨站脚本）；
	Insecure Direct Object References（不安全的直接对象引用）；
	Security Misconfiguration（安全配置错误）；
	Sensitive Data Exposure（敏感信息泄露）；
	Missing Function Level Access Control（功能级访问控制缺失）；
	CSRF（跨站请求伪造）；
	Using Known Vulnerable Components（使用含有已知漏洞的组件）；
	Unvalated Redirects and Forwards（未验证的重定向和转发）。

SDLC-需求阶段



- 需求分析阶段主要是分析系统对外提供的功能，因此，在需求分析阶段的安全功能主要关注身份认证、访问控制、数据保护、抗抵赖等需求：

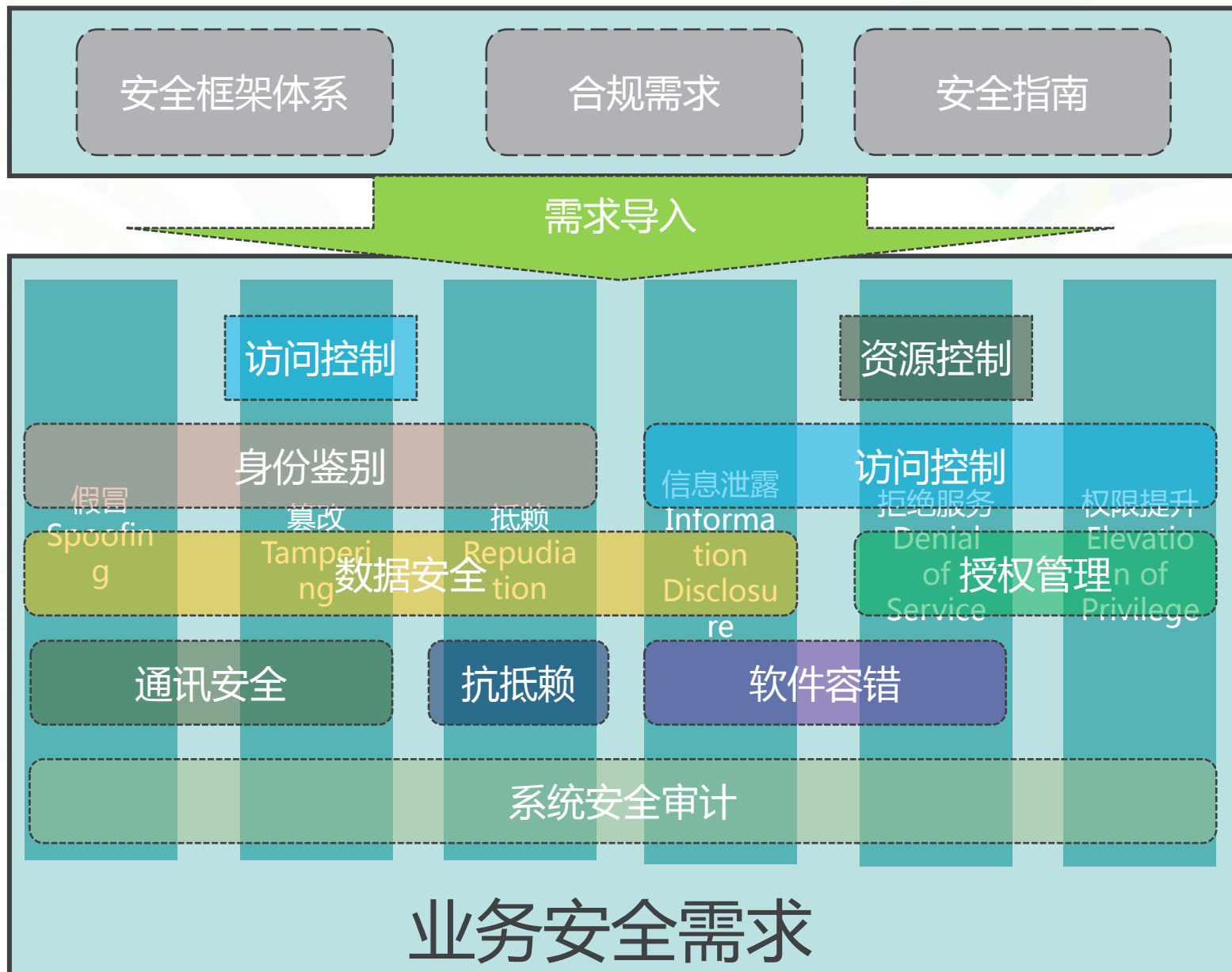
身份认证：用户分类、用户认证等；

访问控制：用户功能权限、用户数据访问权限等；

数据保护：重点保护数据的识别、重点保护数据的保护要求等；

抗抵赖：抗抵赖的用户、操作、数据分析等；

SDLC-需求阶段



SDLC-需求阶段实践

3.1 身份鉴别

- **标识和鉴别：**应支持用户标识和用户鉴别；在每一个用户注册到系统时，采用用户名和用户标识符标识用户身份，并确保在系统整个生存周期用户标识的唯一性；

安全需求	评审内容	涉及范围	自评请选择	自评备注	自评填写
安全需求项	需求子项				若有豁免请求，请在备注
登录/注册需求	注册页面有验证码，登录失败3后，需要出现验证码	项目的应用中有登陆、注册环节时，确认以下安全需求			如果有未满足的需求，请
	登录过程使用https安全通道				请在备注中指出在方案文
	检查登录成功后跳转的URL，是否在白名单内				请在备注中指出在方案文
	登录的账号认证只能由内部登录服务器认证，不能经过第3方转发验证				请在备注中指出在方案文
	对外项目：统一使用UIC提供的登录验证				请在备注中指出在方案文
	登录需要记录日志，日志记录以下信息: 登录时间，登录IP（内部应用记录内部IP）				请在备注中指出在方案文
用户信息使用需求	身份证，银行帐号，信用卡号，密码不能明文在页面显示	用户隐私数据的使用			请在备注中指出在方案文
	密码不能明文保存，其他建议加密保存	用户隐私数据的分类：			请在备注中指出在方案文
	收集信息时，需要明确告知用户获取用户数据的方式/内容	a)真实姓名，身份证，生日，电话号码，联系地址，			请在备注中指出在方案文
	收集信息时，需要明确告知用户获取数据的用途	密码			请在备注中指出在方案文
	收集信息时，用户对自己的隐私数据用可操作权限（如是否用户决定是否输入/是否显示等）	银行帐号，信用卡号，			请在备注中指出在方案文

- **自主访问控制：**应在安全策略控制范围内，使用户/对象创建的具体资源相应的访问操作权限，并能将这些权限部分或全部授予其他用户；
- **控制粒度：**应确定自主访问控制主体和客体的粒度，如主体的粒度可以为用户级，客体的粒度为文件或数据库表级和（或）记录或字段级；
- **特权管理：**各种访问操作应尽可能使用执行该过程所需的最小用户权限。

SDLC-设计阶段



- 基于前期的安全需求分析，通过威胁分析对业务的进行数据流层的呈现和分析，利用STRIDE模型将数据流以及其要素转化成威胁在应用系统的立足点，最终在此基础上形成明确的基于业务流程的安全设案。设计阶段关注的重点：

攻击面分析

威胁建模

基础安全功能的设计

业务安全功能设计

软件基础架构设计安全

攻击面分析



❖ 什么是受攻击面

对一个软件系统可以采取的攻击方法集合，软件的功能、API、接口、资源、数据存储等都是受攻击面

❖ 为什么要降低受攻击面

一个软件的攻击面越大安全风险就越大

较高受攻击面	较低受攻击面
默认执行	默认关闭
打开网络连接	关闭网络连接
同时侦听UDP和TCP流量	仅侦听TCP流量
匿名访问	鉴别用户访问
弱ACLs	强ACLs
管理员访问	普通用户访问
因特网访问	本地子网访问
代码以管理员或root权限运行	代码以Network Services、Local Services或自定义的低权限账户运行
统一缺省配置	用户可选的配置
ActiveX控件	.NET代码
标记有脚本安全的ActiveX控件	未标记有脚本安全的ActiveX控件
非SiteLocked ActiveX控件	SiteLocked ActiveX控件



SDLC-威胁分析模型



威胁分析模型

分析界面	安全纬度	威胁分析	威胁统计		现状	影响程度 (来自：威胁量化表)
			威胁行为	威胁源		
Xxxx系统	鉴权与身份认证	攻击绕过鉴权与身份认证，可访问服务器被攻击	扫描 任意代码执行 密码破解 未授权访问 目录遍历	竞争对手 恶意用户 职业黑客 内部员工 病毒	无安全测试，渗透测试记录 攻击防护设备目前只有天融信防火墙，无针对应用层的防护措施	高
	权限控制	1. https连接验证服务器类访问控制措施缺陷 2. 无用服务被攻击导致权限控制失效 3. 后门，恶意代码程序攻击	任意代码执行 权限绕过 伪造与欺骗 拷贝与修改 特权滥用	竞争对手 恶意用户 职业黑客	服务器加固措施不确定，需要进行手工检查 服务器开启的各类应用的权限控制措施无整体监控 可见度中，攻击难度中，可信度高	中
	日志与审计	访问，登陆，操作日志被绕过	伪造与欺骗 日志清除	竞争对手 恶意用户 职业黑客 内部员工 病毒	部分日志收集，暂无日志关联分析措施，日志收集状况需要 可见度低，攻击难度低，可信度高	中
	机密性	源代码信息泄漏，系统配置信息泄漏	网络指纹识别 源代码偷取	竞争对手 恶意用户 职业黑客 内部员工	系统本身没有进行操作系统加固 可见度中，攻击难度高，可信度高	中
	通信安全	1. ftp 密码被劫持 2. 报文重放攻击 3. 用户登陆信息被劫持	网络监听	国外敌对势力 职业黑客 内部员工	ftp负责web系统文件传输，未使用sftp等加密传输 和内部网络相连，访问控制策略需确认，对内部网络的影响需测试后得出结论 可见度中，系统要求可信度中，攻击难度低	中
	完整性	安装程序完整性确认，整站程序完整性确认	伪造与欺骗 拷贝与修改 网站挂马	竞争对手 恶意用户 职业黑客	服务器应用程序没有白名单，程序本身安全性需要确认。 可见度中，攻击难度中，可信度中	中
	可用性	1. DoS攻击攻击：单用户发起IP层Dos攻击，导致系统过载而无法处理正常业务。 2. DDOS攻击：分布式拒绝服务攻击导致服务不可用	拒绝服务攻击 用户误操作	竞争对手 恶意用户 职业黑客	有负载均衡，但是无针对DOS攻击防护措施 可见度高，攻击难度低，可信度中	高
	隐私保护	用户注册信息泄漏，用户视频信息泄漏	拖库攻击 密码破解	竞争对手 恶意用户 职业黑客	用户敏感信息保护措施未经过测试验证 可见度低，攻击难度中，可信度高	中

SDLC-威胁量化



威胁量化表

分析界面	安全纬度	可见度	可信任度	攻击难度	分值	威胁级别
Xxx系统 (价值说明：未来将会做为重要战略方向)	鉴权与身份认证	3	3	2	7	高
	权限控制	2	3	2	4	中
	日志与审计	1	3	1	2	中
	机密性	2	3	3	3	中
	通信安全	2	2	1	3	中
	完整性	2	2	2	2	中
	可用性	3	2	1	5	高
	隐私保护	1	3	2	1	中

威胁量化表 说明				威胁级别定义	
等级说明	高3	中2	低1		可见度×可信任度-攻击难度=威胁级别
可见度	任意用户可随意访问	部分用户可以访问	访问条件非常苛刻		低：-3-0分
可信任度	业务对其信任需求很高	业务或系统需要获取部分信息	业务对信任要求可有可无		中：1-4分
攻击难度	漏洞利用条件非常苛刻	熟练的攻击者才能完成这次攻击	初学者可以短期内掌握攻击方法		高：5-8分

STRIDE 数据流威胁分析



威胁	安全属性	定义	举例
Spoofing (假冒)	可鉴别性	模仿其他人或实体	伪装成microsoft.com或ntdll.dll。
Tampering (篡改)	完整性	修改数据或代码	修改硬盘、DVD或网络数据包中的DLL
Repudiation (抵赖)	不可抵赖性	声称没有执行某个动作	“我没有发送过那封电子邮件”，“我没有修改过那个文件”，“亲爱的，我确实没有访问过那个网站！”
Information Disclosure (信息泄露)	机密性	把信息披露给那些无权知道的人	允许某人阅读Windows源代码；公布某个Web网站的用户清单。
Denial of Service (拒绝服务)	可用性	拒绝为用户提供服务	使得Windows或Web网站崩溃，发送数据包并耗尽CPU时间，将数据包路由到某黑洞中。
Elevation of Privilege (权限提升)	授权	获得非授权访问权	允许远程因特网用户执行命令，让受限用户获得管理员权限。

STRIDE 威胁削减



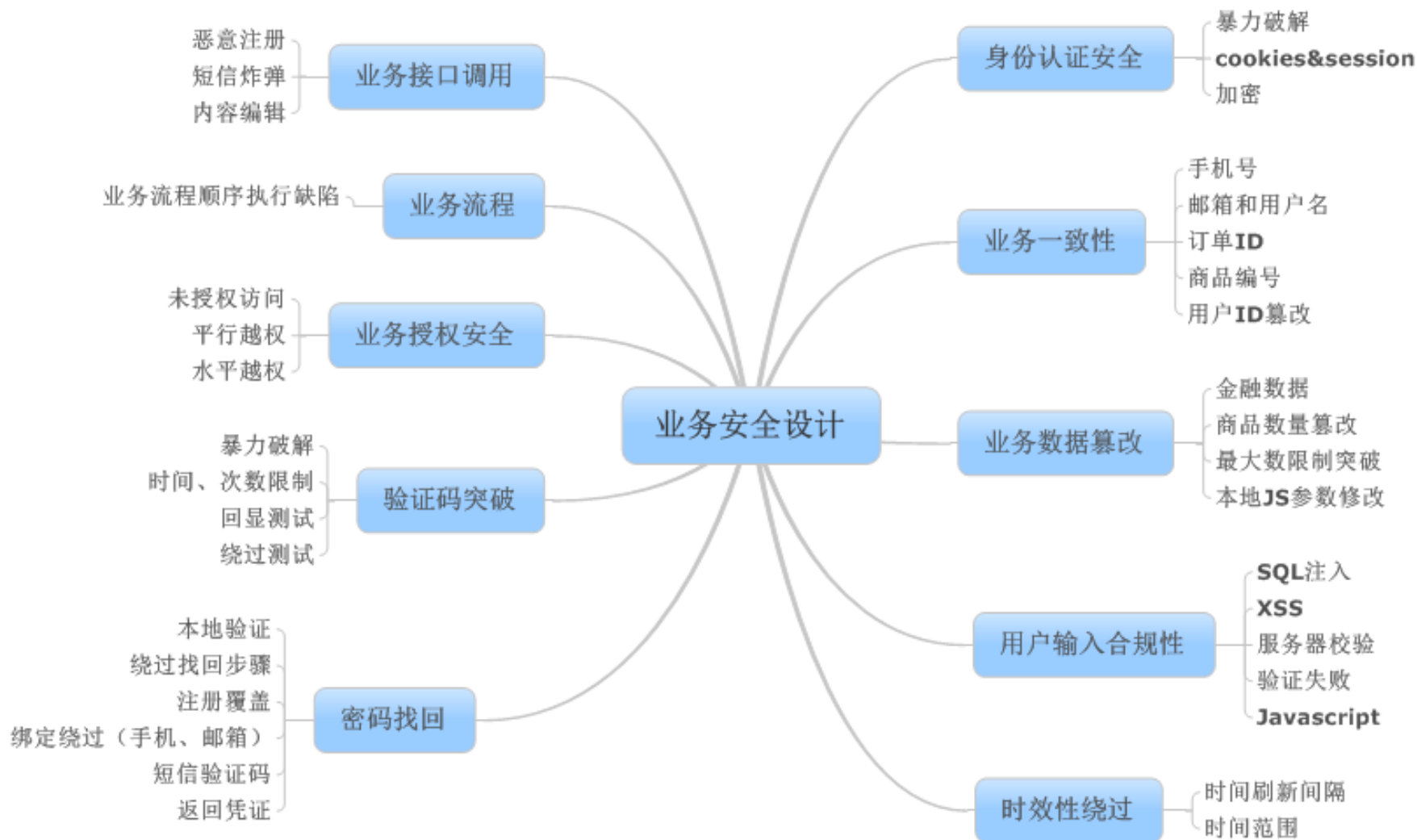
威胁类型	消减机制举例	消减技术举例
假冒	认证	认证方式： Cookie认证、Kerberos认证、PKI
篡改	完整性	哈希函数、消息认证码、数字签名、防篡改协议
抵赖	非抵赖性服务	强认证、安全审计、数字签名、时间戳
信息泄露	保密性	加密、保护秘密、访问控制、不保存秘密、隐私保护协议
拒绝服务	可用性	认证、访问控制、过滤、流量控制、授权
特权提升	授权	访问控制列表、最小权限运行

安全设计-功能安全设计



1.身份鉴别	1.1密码支持	1.1.1 密码输入
		1.1.2 密码强度
		1.1.3 密码有效期
	1.2账户策略	
	1.3辅助安全设备	1.3.1 数字证书
		1.3.2 OTP令牌
		1.3.3 手机短信动态密码
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	3.1.1 会话管理
		3.2.1 数据库和数据文件
		3.2.2 系统之间访问
	3.2 系统外访问控制	3.2.3 客户端非认证访问
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	8.1.1 认证类容错
		8.2.1 命令类容错
		8.2.2 逻辑攻击类容错
9.资源控制		8.2.3 信息泄露类容错
		9.1.1 软件资源配置
	9.1 内部资源控制	9.1.2 资源建立与释放
		9.1.3 敏感资源的信息保护
		9.2.1 支撑资源配置
	9.2 外部资源控制	9.2.2 数据库资源
		9.2.3 输入到系统的资源

安全设计-业务安全设计



安全设计-基础架构设计



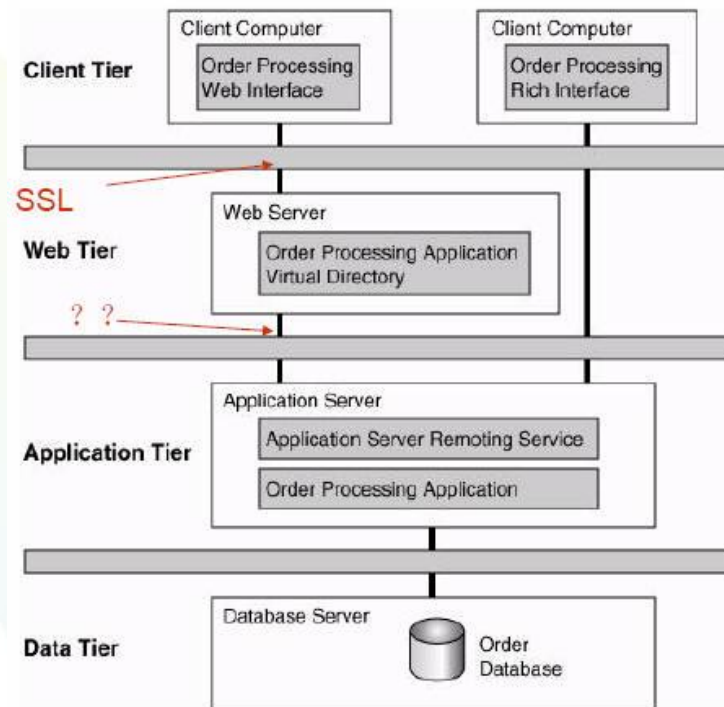
软件部署结构安全：三层部署、纵深防御等

- 软件开发语言环境安全

- 软件开发语言的安全性：如cgi、asp安全性较差，delphi语言缺少安全检测技术，难以验证其安全性等。
- 所需开发包的安全性
- 依赖第三方中间件的安全性
- 第三方应用系统接口安全性

- 所选新技术的安全性

- Strust 2\Open ssl



SDLC-开发阶段



安全培训



需求阶段



设计阶段



开发阶段



测试阶段



上线阶段



响应阶段

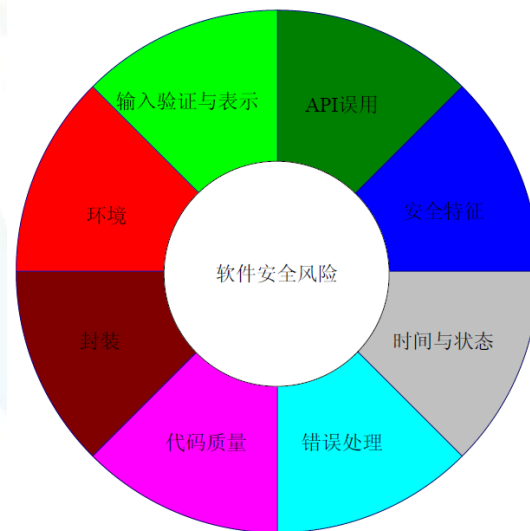


在软件的编码阶段，主要软件安全问题来源于如下几个方面：

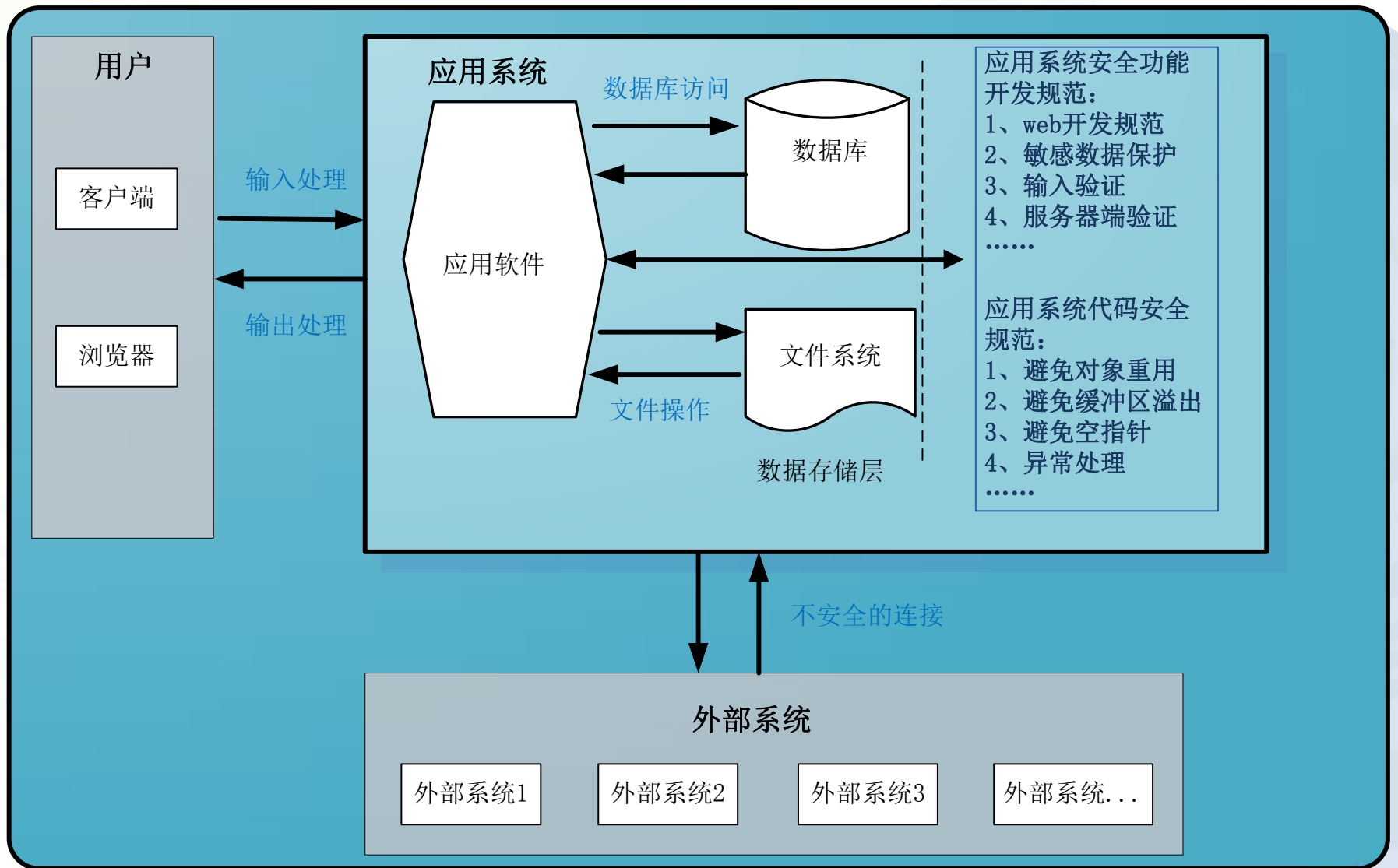
- 软件自身的代码缺陷
- 用户恶意输入
- 不期望的连接

在整个软件系统中主要体现在如下几个方面带来安全隐患：

- 输入验证与表示
- API误用
- 安全特征
- 时间与状态
- 错误处理
- 代码质量
- 封装和环境等安全漏洞



开发阶段-系统架构



开发阶段-通用编码规则



- 统一的安全规范：包括命名规则、**API引用**、错误处理、避免使用全局变量等
- 模块划分：应按安全性划分模块，审计和访问控制等模块为安全可信模块，其它模块为不可信任模块。只有安全可信模块才可以执行安全控制功能。
- 最小功能性：根据“没有明确允许的就默认禁止”的原则，应只包含那些为达到某个目标而确实需要的功能
- 多任务、多进程：应尽量使用单任务，如果使用多任务多进程时同步所有的进程和任务以避免冲突
- 界面输出最小化：用户界面只提供必须的功能和信息
- 避免高危的服务、协议
- 数据和代码分离：应该把数据与程序放置在不同的目录中
- 禁止赋予用户进程特权
- 使用适当的数据类型
- 使用经过验证的安全代码
- 设计错误、异常处理机制等

开发阶段-安全框架



安全框架主要实现安全特性增强和缓解安全漏洞，其中功能主要包括：会话管理、访问控制、输入验证、输出编码、密码、日志记录等，**实践证明，通过使用成熟的安全框架比开发人员自实现安全特性或黑名单等方式过滤更加高效可靠**，企业统一维护一套安全框架已是安全开发生命周期流程的基本要求。

Log4j	http://logging.apache.org/log4j/
Spring Security	https://projects.spring.io/spring-security/
ESAPI	https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API
Apache Commons Validator	http://commons.apache.org/proper/commons-validator/
Hibernate Validator	http://hibernate.org/validator/
Apache Santuario	http://santuario.apache.org/
Jasypt	http://www.jasypt.org/
Apache Shiro	https://shiro.apache.org/
BouncyCastle	https://www.bouncycastle.org/
AntiSamy	https://www.owasp.org/index.php/Category:OWASP_AntiSamy_Project
HDIV	https://hdivsecurity.com/

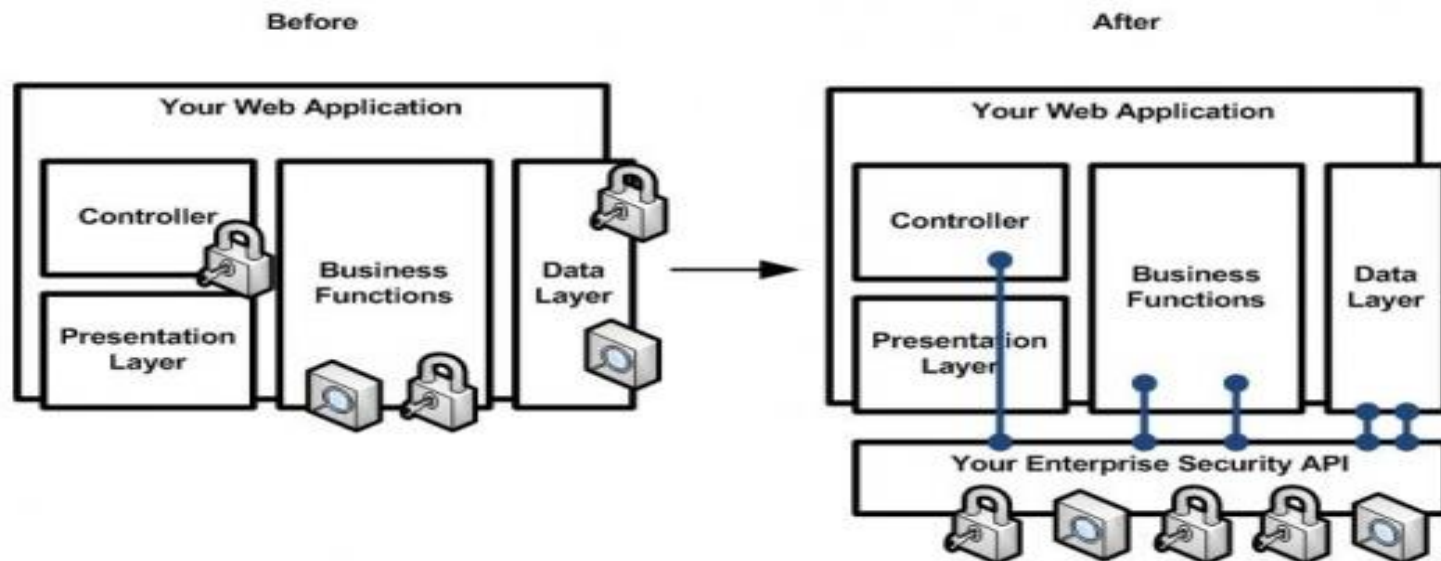
开发阶段-安全框架ESAPI

ESAPI (OWASP企业安全应用程序接口)是一个免费、开源的、网页应用程序安全控件库，它使程序员能够更容易写出更低风险的程序。ESAPI接口库被设计来使程序员能够更容易的在现有的程序中引入安全因素。ESAPI包含安全控件的示例：

- 身份认证
- 访问控制
- 输入验证

- 通信安全
- HTTP 安全
- 安全配置

- 输出编码/转义
- 密码
- 错误处理和日志



开发阶段-编码规范



Java安全	.NET安全	JavaScript安全	PHP安全	
6.1. 安全	8.1. .NET	7.1. 避	9.1. PH	 软件安全开发生命周期安全编码环境配置知识库 (Java) .doc
6.2. 输入	8.2. 输入	7.2. 注	9.2. XS	 软件安全开发生命周期安全编码环境配置知识库 (NET) .doc
6.3. 输出	8.3. 输出	7.3. D	9.3. HT	 软件安全开发生命周期安全编码知识库 (Java) .doc
6.4. 跨站	8.4. 跨站	7.4. JC	9.4. 代	 软件安全开发生命周期安全编码知识库 (Javascript) .doc
6.5. 路径	8.6. 路径	7.5. 不	9.5. 命	 软件安全开发生命周期安全功能编码知识库 (NET) .doc
6.6. 命令	8.7. 命令	7.6. 包	9.6. SQ	 软件安全开发生命周期安全功能编码知识库 (Objective-C) .doc
6.7. JSC	8.8. JSC	7.7. IF	9.7. 文	 软件安全开发生命周期安全编码环境配置知识库 (PHP) .doc
6.8. Me	8.9. Me	7.8. 事	9.8. 文	 软件安全开发生命周期安全编码知识库 (PHP) .doc
6.9. 变	8.10. 变	7.9. 子	9.9. 变	 软件安全开发生命周期安全功能编码知识库 (PHP) .doc
6.10. X	8.11. X	7.10. 子	9.10. 文	
6.11. X	8.12. X	7.10. 子	9.11. 变	
6.11. X	8.13. X	7.10. 子	9.12. 文	



SQL注入编码规范
列 (包含框架引入

SDLC-测试阶段



安全培训



需求阶段



设计阶段



开发阶段



测试阶段



上线阶段



响应阶段



- 测试阶段是指业务代码已具备所有功能并进入用户试用版测试的阶段。在对业务代码进行试用版测试时，包括进行安全代码审核和集中式安全测试（配置管理测试、认证测试、会话管理测试、授权测试、业务逻辑测试、数据验证测试、阻断服务测试、WEB服务测试等）。测试阶段关注的重点：

安全功能测试

黑盒测试

白盒测试

模糊测试

安全测试VS功能测试



- **传统的功能测试不同的安全测试，功能测度不能替代安全测试！**
- **功能测试:**
 - ✓ 功能测试验证应用程序是否做它应该做的。
 - ✓ 包括提交输入去验证正确的输出。
 - ✓ 功能测试者会问 “什么是软件应该做的？”
- **安全测试:**
 - ✓ 安全测试验证应用程序不做它不应该做的。
 - ✓ 包括提交输入去验证不出现异常。
 - ✓ 安全测试者会问 “什么是软件不应该做的？”

安全测试



测试分类	测试方式	测试条件	特点	测试重点
位置分类	外部测试	用户的办公网络甚至业务网络。	绕过防火墙、入侵保护等安全设备。	检测内部威胁源和路径。
	内部测试	直接从互联网访问用户的某个接入到互联网的系统并进行测试。	可以测试外部防护中存在的风险。	检测外部威胁源和路径。
方法分类	白盒测试	获得应用源码	模拟内部恶意用户。	模拟并检测内部的恶意用户可能为系统带来的威胁。
	黑盒测试	目标系统的IP或域名	模拟黑客行为。	模拟外部恶意用户可能为系统带来的威胁
	灰盒测试	获得部分测试信息，例如账号，流程等	模拟内外部攻击行为	模拟内外部攻击行为

安全测试规范



2. Web 安全漏洞

- 2.1. 十大 Web 应用风险
- 2.2. 十大业务逻辑漏洞

3. Web 安全测试规范

3.1. 十大 Web 应用风险

- 3.1.1. 注入
- 3.1.2. 跨站脚本
- 3.1.3. 失效的身份认证和会话管理
- 3.1.4. 不安全的直接对象引用
- 3.1.5. 跨站请求伪造
- 3.1.6. 安全配置错误
- 3.1.7. 功能及访问控制缺失
- 3.1.8. 使用含有已知漏洞的组件
- 3.1.9. 未认证的重定向和转发
- 3.1.10. 传输层保护不足

3.2. 十大业务逻辑漏洞

- 3.2.1. 身份认证安全
- 3.2.2. 业务一致性安全
- 3.2.3. 业务数据篡改
- 3.2.4. 用户输入合规性

3.2.5. 密码找回漏洞

3.2.6. 验证码突破

3.2.7. 业务授权安全

3.2.8. 业务流程乱序

3.2.9. 业务接口调用

3.2.10. 时效性绕过

3.3. 其他严重安全漏洞

3.3.1. 文件上传漏洞

3.3.2. 敏感信息泄漏

3.4. 自动化安全扫描工具测试

3.4.1. 黑盒安全测试

3.4.2. 白盒安全测试

4. 附件

4.1. burpsuite 工具使用指南

4.2. SQLMAP 工具使用指南

4.3. WEB 安全扫描平台使用指南

4.4. 源代码安全扫描平台试用指南

4. 附件

4.1.burpsuite 工具使用指南

4.2.SQLMAP 工具使用指南

4.3.WEB 安全扫描平台使用指南

4.4.源代码安全扫描平台试用指南

information classification: Internal
信息分类: 内部

安全功能测试

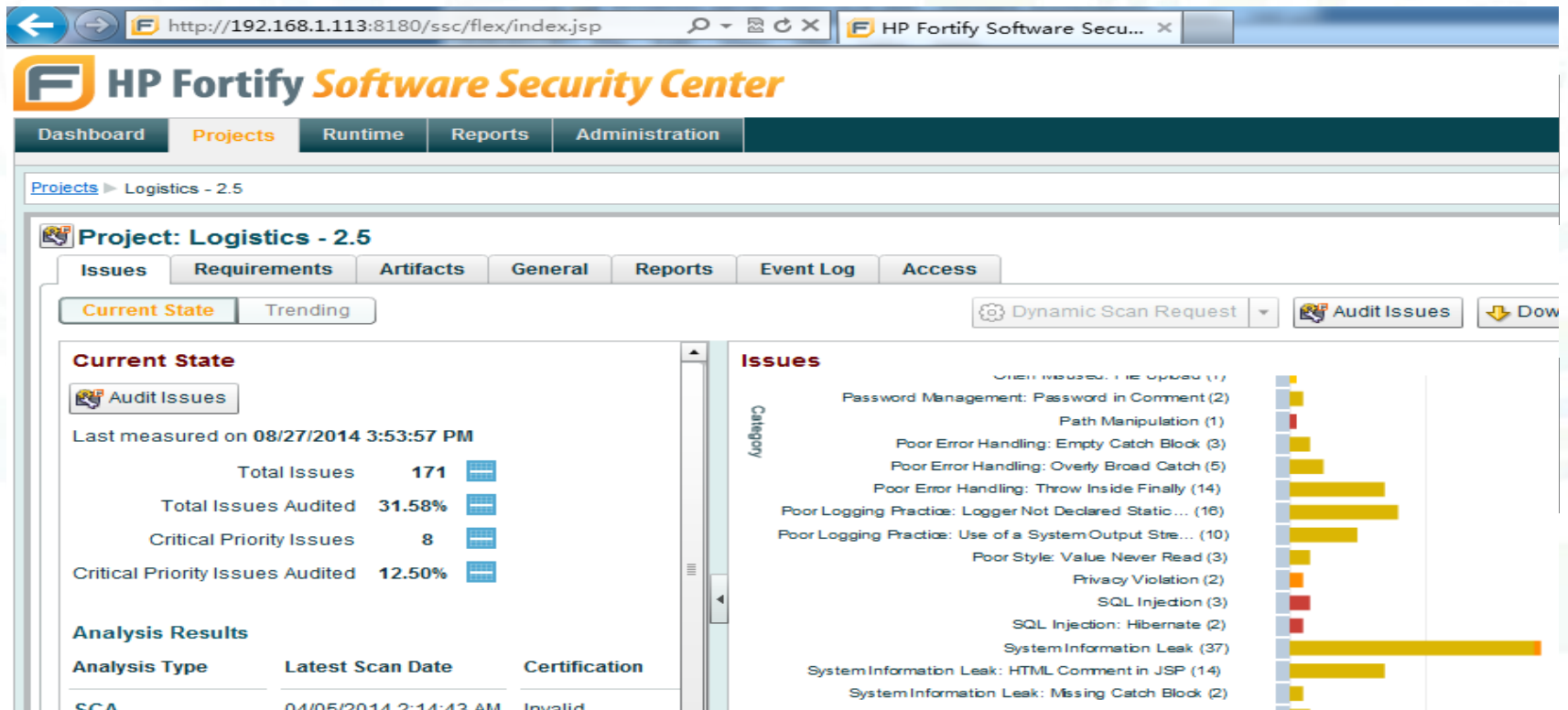


小类	序号	检查内容	必测/可选	测试操作方法	安全性评价标准	测试结果	主要存在的问题
配置口令策略	1	设置用户口令允许的最短长度	可选	系统管理员设置用户口令允许的最短长度为0	用户口令长度的取值范围为[1,32]，不允许设置为空	无	页面无此功能，需求也没有提到
	2	设置用户口令允许的最大长度	可选	系统管理员设置用户口令允许的最大长度为33	用户口令长度的取值范围为[1,32]，不允许设置最大长度为33	无	页面无此功能，需求也没有提到
	3	设置用户口令允许的最小长度大于最大长度	可选	系统管理员设置用户口令允许的最短长度为20，最大长度为8	用户口令允许的最小长度不能大于最大长度	无	页面无此功能，需求也没有提到
	4	设置用户口令允许的最小长度、最大长度符合安全基线要求	可选	系统管理员设置用户口令允许的最短长度为8，最大长度为20	用户口令长度的取值范围为[1,32]，且最小长度小于或等于最大长度	无	页面无此功能，需求也没有提到
	5	设置口令中必须包含特殊字符且指定特殊字符集合	必测	系统管理员设置口令中必须包含特殊字符且指定特殊字符集合『! "\$ % & ' () * + , - . / : ; < = > ? @ [\ ^ _ { } ~ 』及空格	系统管理员可以设置口令中是否必须包含特殊字符	无	页面无此功能，需求也没有提到
	6	设置口令中可以不包含特殊字符	可选	系统管理员设置口令中可以不包含特殊字符	系统管理员可以设置口令中是否必须包含特殊字符	无	页面无此功能，需求也没有提到
	7	设置口令中可重复字符数	必测	分别设置口令中可重复字符数为0,1,3,6,7	最多可重复字符数要求范围为[1,6]，因此设置为0或7均不能成功	无	页面无此功能，需求也没有提到
	8	设置口令弱词典	可选	系统管理员分别进行添加、修改、删除弱口令操作	系统管理员可以及时维护弱口令词典，尤其是在口令被泄密（或被猜到）时	无	页面无此功能，需求也没有提到
	9	设置历史口令记录数	可选	系统管理员分别设置历史口令记录数为-1,0,3,30,31等值	历史口令记录数范围为[0,30]，在此范围之外设置不成功	无	页面无此功能，需求也没有提到
	10	设置最短口令保存期	可选	系统管理员分别设置最短口令保存期为-1,0,5,1440,1441（单位为分钟时）	最短口令保存期为[0,1440]分钟，在此范围之外设置不成功	无	页面无此功能，需求也没有提到
	11	设置最长口令保存期	可选	系统管理员分别设置最长口令保存期为0,1,90,1096,1097（单位为天）	最长口令保存期为[1,1096]天，在此范围之外设置不成功	无	页面无此功能，需求也没有提到
	12	设置口令到期前提示天数	可选	系统管理员分别设置口令到期前提示天数为0,1,7,99,100（单位为天）	口令到期前提示天数范围[1,99]天，在此范围之外设置不成功	无	页面无此功能，需求也没有提到

白盒测试



- 自动评估程序代码功能
- 分析所有可能出错的输入点
- 开发中立即指出程序弱点所在
- 开发中立即提出可行的安全程序建议方案
- 快速、有效率且无副作用的安全程序设计



黑盒测试



- + JSP 文件包含 (13)
- + SQL 盲注 (3)
- + SQL 注入 (8)
- + 存储的跨站点脚本编制 (17)
- + 基于 DOM 的跨站点脚本编制 (1)
- + 跨站点脚本编制 (9)
- + 通过 URL 重定向钓鱼 (9)
- + 已解密的登录请求 (8)
- + Windows 文件参数变更 (2)
- + 不充分帐户封锁 (1)
- + 会话标识未更新 (1)
- + 跨站点请求伪造 (2)
- + 链接注入 (便于跨站请求伪造) (5)
- + 启用了不安全的 HTTP 方法 (1)
- + 通过框架钓鱼 (5)
- + Apache Tomcat Cookie 处理会话标识泄露 (1)
- + 多供应商 Java Servlet 源代码泄露 (1)
- + 会话 cookie 中缺少 HttpOnly 属性 (1)
- + 直接访问管理页面 (2)
- + Poison Null Byte Windows 文件检索 (1)
- + SQL 盲注 (7)
- + SQL 注入 (13)
- + 格式字符串远程命令执行 (1)
- + 基于 DOM 的跨站点脚本编制 (3)
- + 可预测的登录凭证 (1)
- + 跨站点脚本编制 (14)
- + 使用 SQL 注入的认证旁路 (2)
- + 通过 URL 重定向钓鱼 (1)
- + 已解密的登录请求 (5)
- + XPath 注入 (1)
- + 不充分帐户封锁 (1)
- + 会话标识未更新 (1)
- + 开放式重定向 (2)
- + 跨站点请求伪造 (7)
- + 链接注入 (便于跨站请求伪造) (6)
- + 路径遍历 (1)
- + 目录列表 (1)
- + 通过框架钓鱼 (6)

灰盒测试



american fuzzy lop 0.47b (readpng)

process timing

run time : 0 day
last new path : 0 day
last uniq crash : none
last uniq hang : 0 day
cycle progress
now processing : 38 (0)
paths timed out : 0 (0)

stage progress

now trying : interest
stage execs : 0/9990 (0)
total execs : 654k
exec speed : 2306/sec

fuzzing strategy yield

bit flips : 88/14.4k
byte flips : 0/1804,
arithmetics : 31/126k,
known ints : 1/15.8k,
havoc : 34/254k,
trim : 2876 B/9

Burp Suite Professional

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Results Scan queue Live scanning Options

Issues

- SQL injection [7]
- Cross-site scripting (stored)
- HTTP header injection
- Cross-site scripting (reflected)**
- Cleartext submission of password [2]
- OS command injection
- LDAP injection
- Open redirection
- Password field with autocomplete enabled [2]
- Cross-domain Referer leakage [2]

Advisory Request Response

Cross-site scripting (reflected)

Issue: Cross-site scripting (reflected)
Severity: High
Confidence: Certain
Host: http://0b7bd624bab7.mdseclabs.net
Path: /search/11/Default.aspx

Issue detail

The value of the **SearchTerm** request parameter is copied into the HTML document as plain text between tags. The payload `1d329<script>alert(1)</script>27a3a1b60c71d9423` was submitted in the **SearchTerm** parameter. This input was echoed unmodified in the application's response.

SDLC-上线阶段



- 上线（发布）阶段主要是对软件进行最终安全审核，在完成上述工作流程后安全转交给运维人员进行系统上线。对最初标识为安全漏洞，但后来经过深入分析确定为对安全性没有影响的缺陷进行审核，以确保分析的正确性；包括审核业务代码是否能抵御最新报告的影响类似业务代码的漏洞。可能还需要利用外面的安全审核承包商来协助进行安全审核。
- 安全配置
- 支持文档
- 支撑组件清单

清理调试信息

上线部署前必须将代码中的调试信息进行清理。不能将带有调试选项的代码部署到生产系统中。

清理WEB源代码注释

上线部署前必须清理html 等web程序源代码中出现的与软件设计、Web服务器环境、文件系统结构相关的所有的参考和注释；这些信息包括但不限于：

- （1）目录结构；
- （2）Web 根目录的位置；
- （3）调试信息；
- （4）Cookie结构；
- （5）开发中涉及到的问题；
- （6）开发者的姓名、email地址、电话号码等；

清理不需要的代码

上线部署前必须清理软件程序代码中不需要的代码和那些不能完成任何功能的代码。

网络服务管理

服务器必须对提供的服务端口进行控制。要求在需求分析中明确说明本系统必须开放的网络服务。在实际运行环境中必须严格按照需求中的要求实施、部署。

代码及文档安全



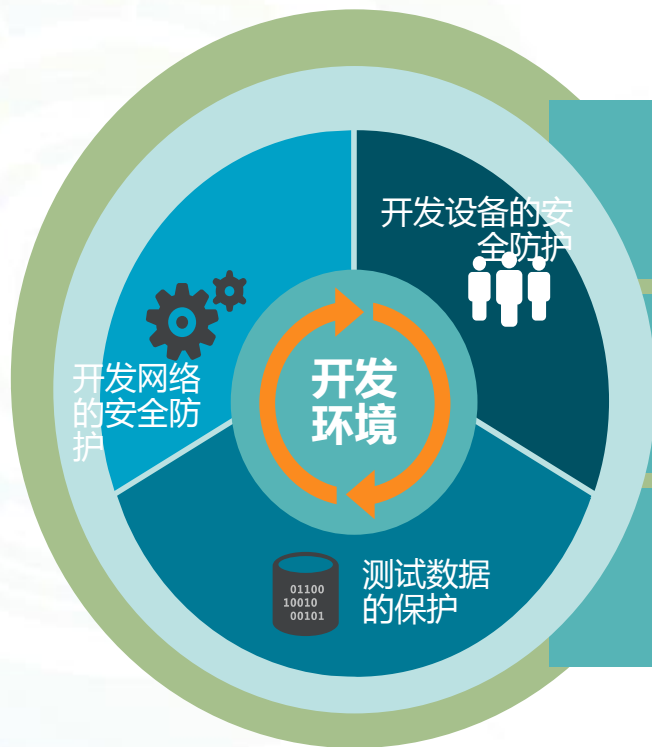
按“最小必需”的原则设定源代码的访问权限，严格控制外包人员对源代码的访问权限；严禁源代码带离开发环境；

应采取合同制约、源代码抽查、代码中恶意代码检测等方式对源代码中可能存在的后门、木马、隐蔽信道等进行管理

软件开发所使用的操作系统、通信软件、数据库等必须是正式版本软件

与应用软件开发相关的核心技术、设计文档的安全保护

开发环境



开发网络与测试网的有效隔离；严格控制对源代码服务器和文档服务器的访问控制；严格控制设备的接入

禁止在开发环境使用超级用户或者其他特权用户进行软件开发；开发用机必须安装的相关安全管理软件如防病毒软件、补丁管理软件、数据安全软件等

测试数据的脱敏处理；测试数据的及时销毁；测试数据使用范围的严格控制等

安全交付



系统安全交付清单模板：

1. 系统各组件安装盘
2. 系统各组件安装指南文档
3. 组件安装到各平台的文件清单，注册表清单（增加，删除，更改），以及其他安装动作清单（如打开或关闭防火墙端口，打开关闭操作系统相关服务）
4. 系统各组件的安全操作指南
5. 系统各组件的数据备份指南
6. 系统各组件的日志审计指南
7. 系统各组件的操作系统平台配置清单
8. 系统各组件的网络配置指南，包括外部防火墙设置，IP地址配置等

9.



XXX系统安全交付标准：

1. 系统各组件安装盘
2. 系统各组件安装指南文档
3. 系统各组件配置文件清单及其配置
4. 系统各组件的安全操作指南
5. 系统各组件的操作系统平台配置清单
6. 系统各组件的网络配置指南，包括外部防火墙设置，IP地址配置等
7.

SDLC-响应阶段



安全培训



需求阶段



设计阶段



开发阶段



测试阶段



上线阶段

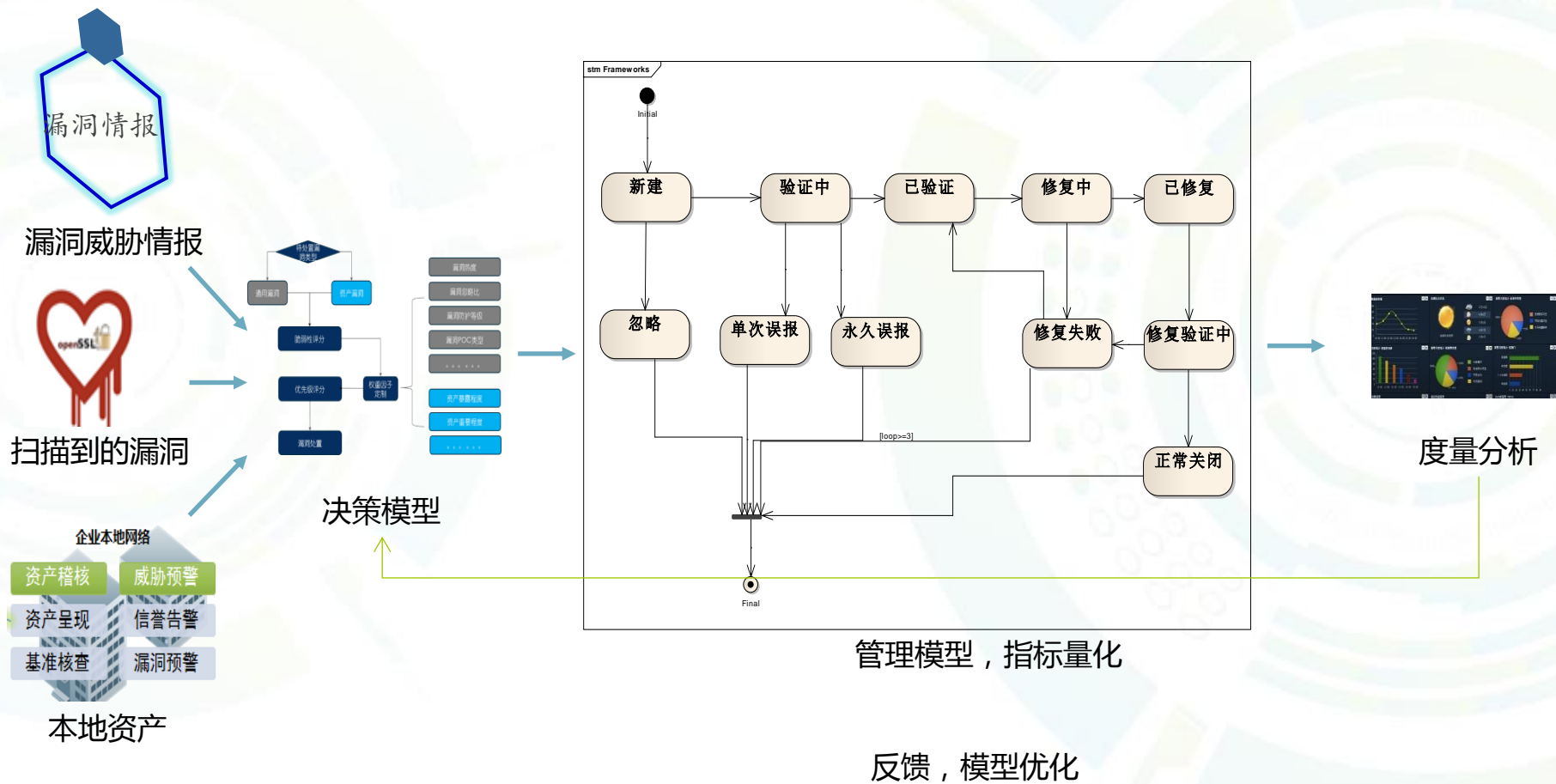


响应阶段

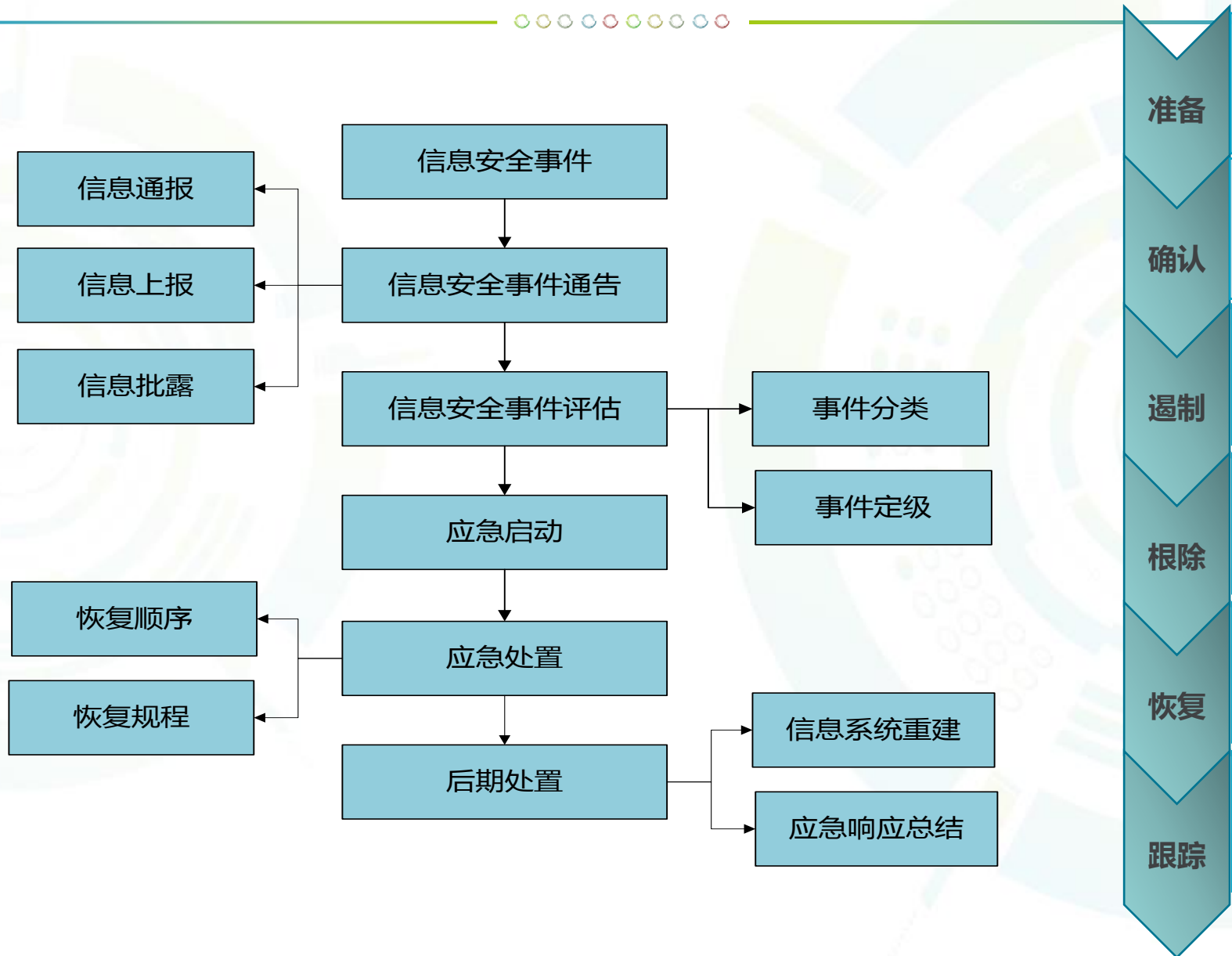
- 响应阶段的目标是从错误中吸取教训，并使用漏洞报告中提供的信息帮助在业务代码投入使用前检测和消除深层漏洞，以免这些漏洞给用户带来危害。帮助业务系统人员分析并在合适的情况下对流程进行改造，以免将来犯类似错误。主要内容如下：
- 漏洞的获取及验证
- 补丁的测试及加固
- 应急响应



漏洞发现及修复



应急响应





谢谢