漏洞管理流程

修订记录

编号	日期	修订内容
1	2019.3.15	初稿
A2	B2	C2

目录

1 概述

合适的漏洞响应可以尽快减少易受攻击的产品实例的数量，并减少针对易受攻击系统的攻击。

良好的漏洞管理流程的作用：

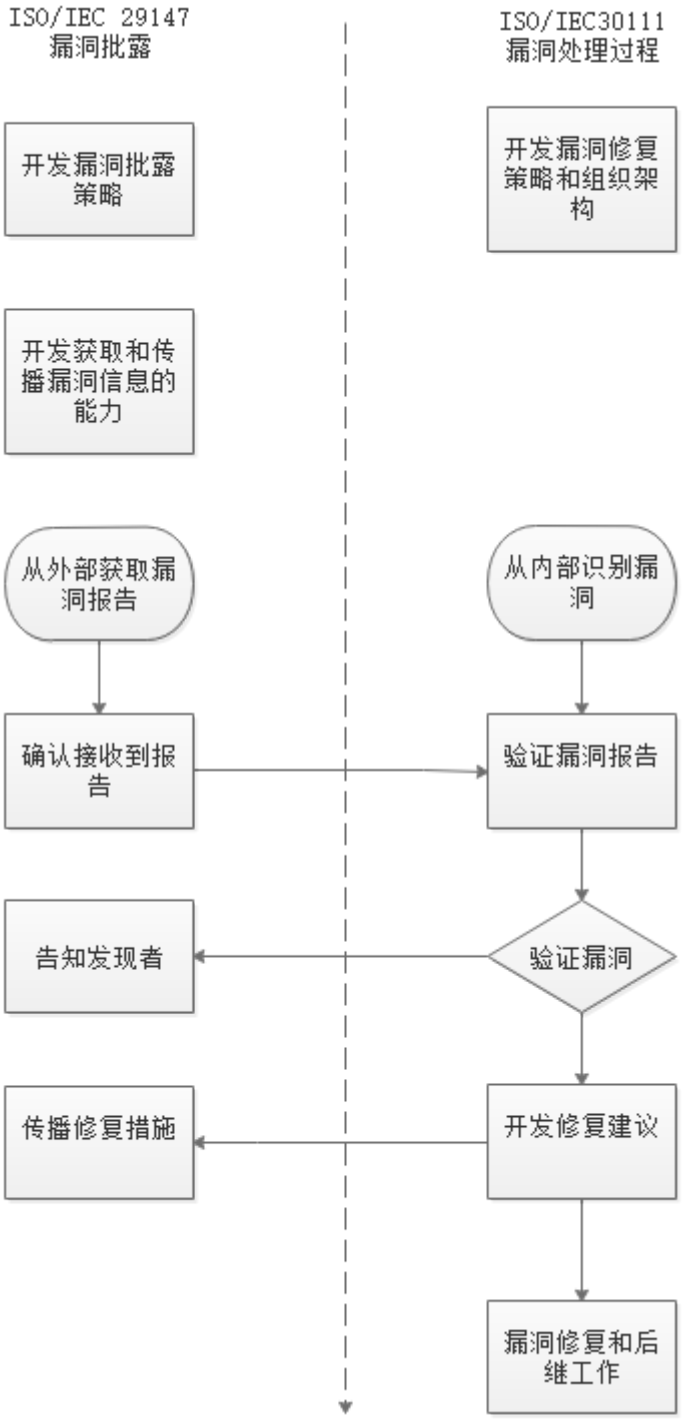
- 对组织
  - 提升漏洞修复效率
  - 降低漏洞再次发生的可能性

    - 漏洞修复建议知识库
    - 降低整体安全风险
- 对用户
    - 降低用户个人信息暴露的风险

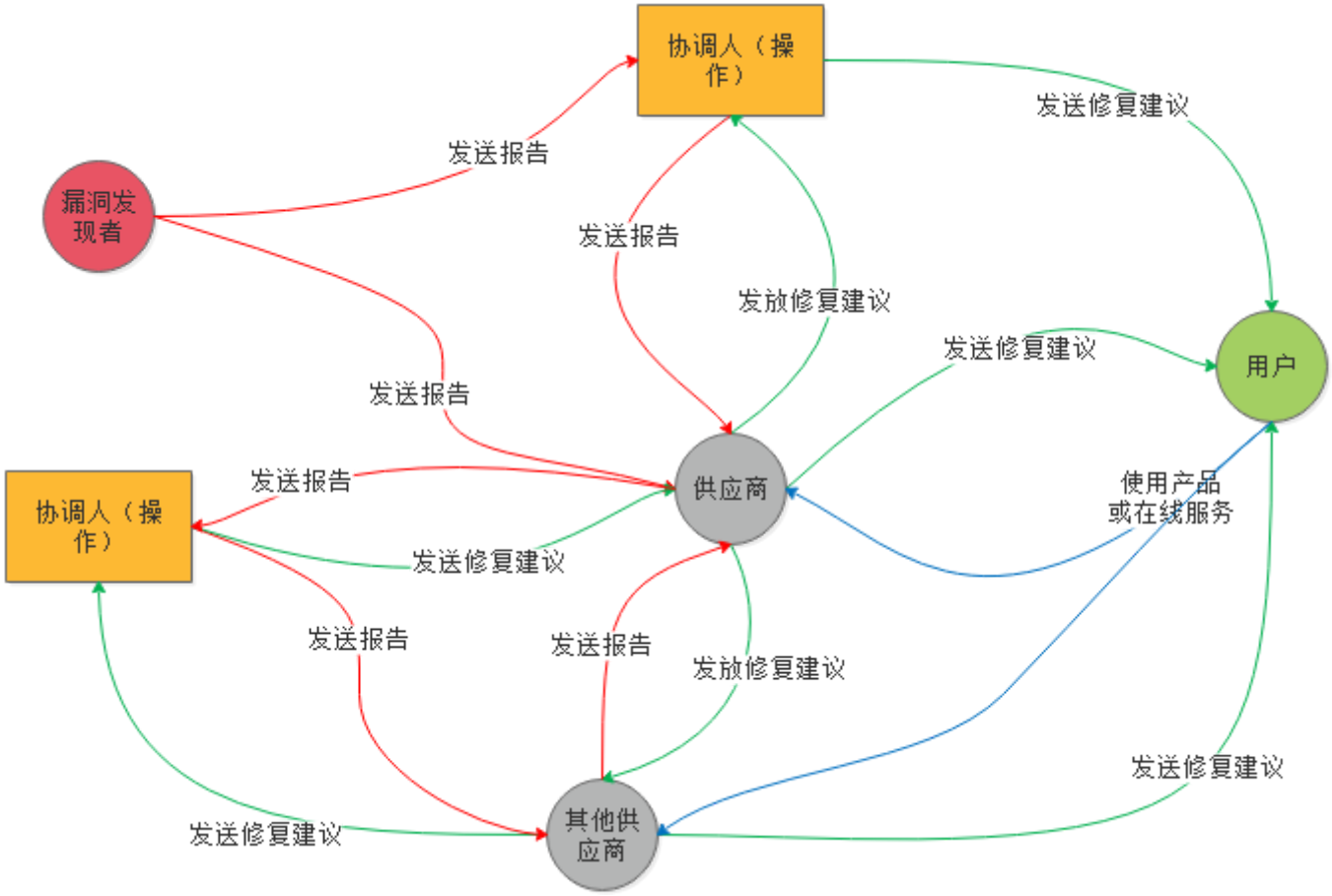
2 漏洞处理相关的标准和流程

2.1 ISO/IEC 29147 和 ISO/IEC 30111

漏洞批露和漏洞处理标准



漏洞信息交换



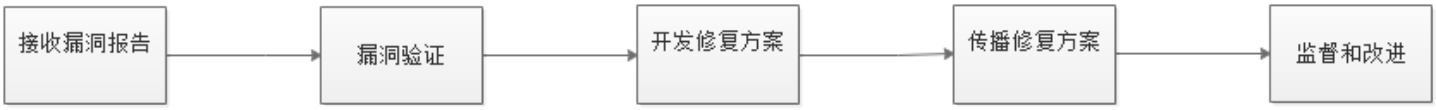
漏洞批露标准 ISO/IEC 29147

- 供应商应该有明确的方式来接收漏洞报告
- 供应商应在7个日历日内确认收到漏报报告
- 供应商应与发现者沟通协调（了解发现者的期望和详细的漏洞信息）
- 供应商应发布包含有用信息的建议，至少：
    - 漏洞的独特标识符
    - 受影响的产品
    - 如果利用漏洞，损害的影响/严重程度
    - 如何消除或缓解问题（指导或补丁说明）
- 如果发现者希望公开漏洞，建议给予发现者咨询相关的奖励。

漏洞处理流程 ISO/IEC 30111

- 供应商应该有一个流程和组织结构来支持漏洞调查和补救
- 供应商应该进行根本原因分析
- 供应商应权衡各种补救方案以适应现实世界的风险因素
    - 平衡速度和漏洞修复的彻底性
- 供应商应尝试与其他供应商进行适当的协调
    - 多供应商问题
    - 供应链问题

2.2 漏洞处理流程



流程处理流程

供应商漏洞验证相关的工作：

- 初步调查：供应商尝试确认潜在漏洞
- 根本原因分析：供应商尝试确定漏洞的根本原因
- 进一步调查：供应商试图在产品或服务中查找相同类型漏洞的其他实例， 或在其他产品中。
- 优先级：供应商将漏洞所构成的威胁视为受影响的产品或在线服务用户
- 对于每个受影响的产品或在线服务，可能存在相同基础问题的不同严重性

供应商处理漏洞可能的情况：

- 无法复现的漏洞
- 已知重复错误：问题是一个重复的漏洞，已经通过此过程解决或已经修复
- 过时的产品错误：该漏洞存在于供应商不再支持的产品中
- 非安全性错误：问题是一个没有安全隐患，或者目前已知技术无法利用的错误
- 第三方错误：该漏洞是由第三方代码，配置引起的，或者存在于供应商不直接负责的规范中

开发漏洞修复策略

- 解决方案决策：供应商确定如何全面解决漏洞，如何减少成功利用漏洞的影响，或如何减少暴露。
- 生成修复补丁：供应商生成修补程序，修复程序，升级程序或文档或配置更改以解决漏洞。
- 测试修复策略（补丁）：供应商开发并执行适当的测试，以确保在所有支持的平台上解决了漏洞问题。

发放漏洞修复方案：

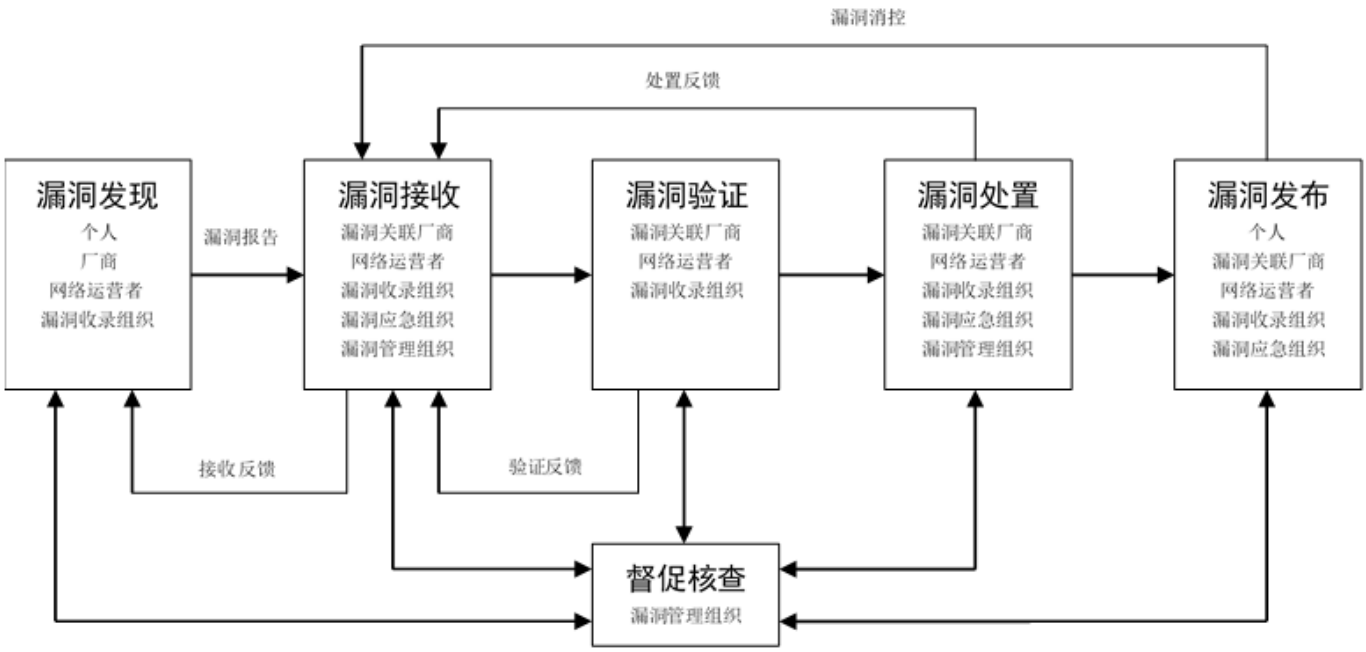
- 在线服务漏洞解决方案：遵循组织的生产系统更新部署或配置更改过程。
- 产品漏洞解决方案：对于受影响的用户必须采取某些措施来保护自己的产品中的漏洞（例如，安装补丁）

修复方案发布后工作

- 案例维护：解决方案发布后，可能会继续对解决方案进行进一步更新。
- 安全开发生命周期反馈：供应商使用在根本原因分析期间获得的信息更新开发生命周期，以防止新的或更新的产品或服务中出现类似的漏洞。
- 监控：
    - 对于在线服务漏洞，在供应商应用补救措施后，供应商应监控产品或服务的稳定性。
    - 用于开发的补丁后发布监控可以帮助将通信集中到大多数受影响的用户。

2.3 信息安全技术 网络安全漏洞管理规范

漏洞生命周期



漏洞生命周期的相关阶段
漏洞管理生命周期包含以下阶段：

- 漏洞发现：通过人工或者自动的方法对漏洞进行探测、分析，并证实漏洞存在的真实性的过程。
- 漏洞接收：通过相应途径接收漏洞报告者提交的漏洞信息的过程。
- 漏洞验证：收到漏洞报告后，进行漏洞信息的技术验证、确认和反馈的过程。
- 漏洞处置：通过升级版本、补丁、更改配置等方式，对漏洞进行修补的过程。
- 漏洞发布：通过公开渠道（如网站、邮件列表等）将漏洞信息向社会公布，或向限定范围的个人和组织公布的过程。
- 督促核查：督促并监督漏洞管理活动的实施情况。

漏洞报告应包含的内容

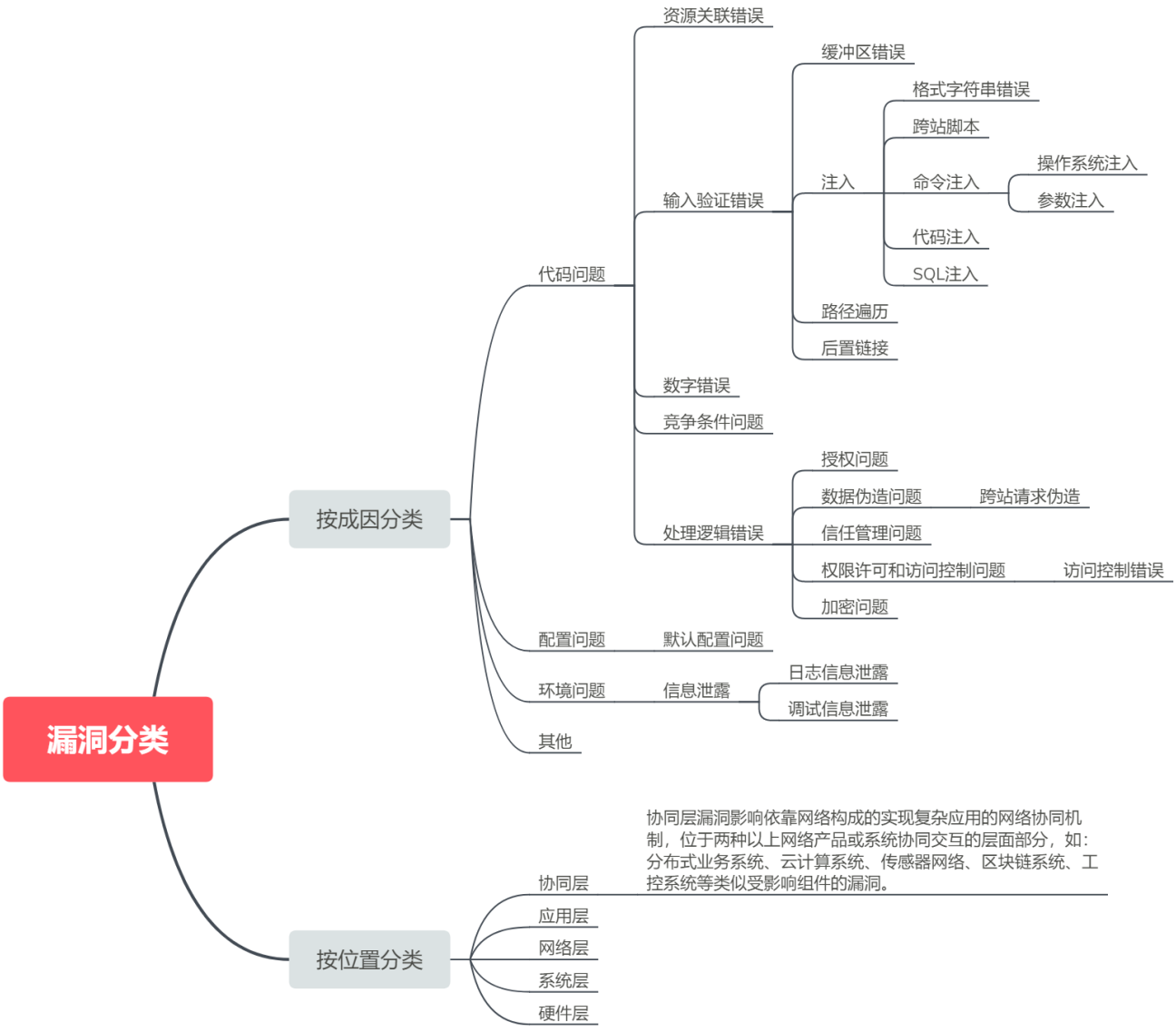
- 报告人信息
    - 姓名
    - 组织
    - 邮箱
    - 电话
    - 是否公开身份
- 漏洞信息
    - 漏洞名称
    - 漏洞位置（相关产品/服务名称、版本、URL地址或者目录）
    - 漏洞所属（关联厂商，信息系统管理者）
- 漏洞描述
- 漏洞复现方法
- 漏洞利用场景描述
- 漏洞预估等级

- 漏洞修复建议

2.4 信息安全技术 网络安全漏洞分类分级指南

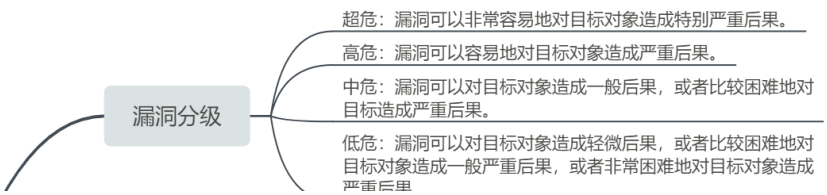
2.4.1 漏洞分类

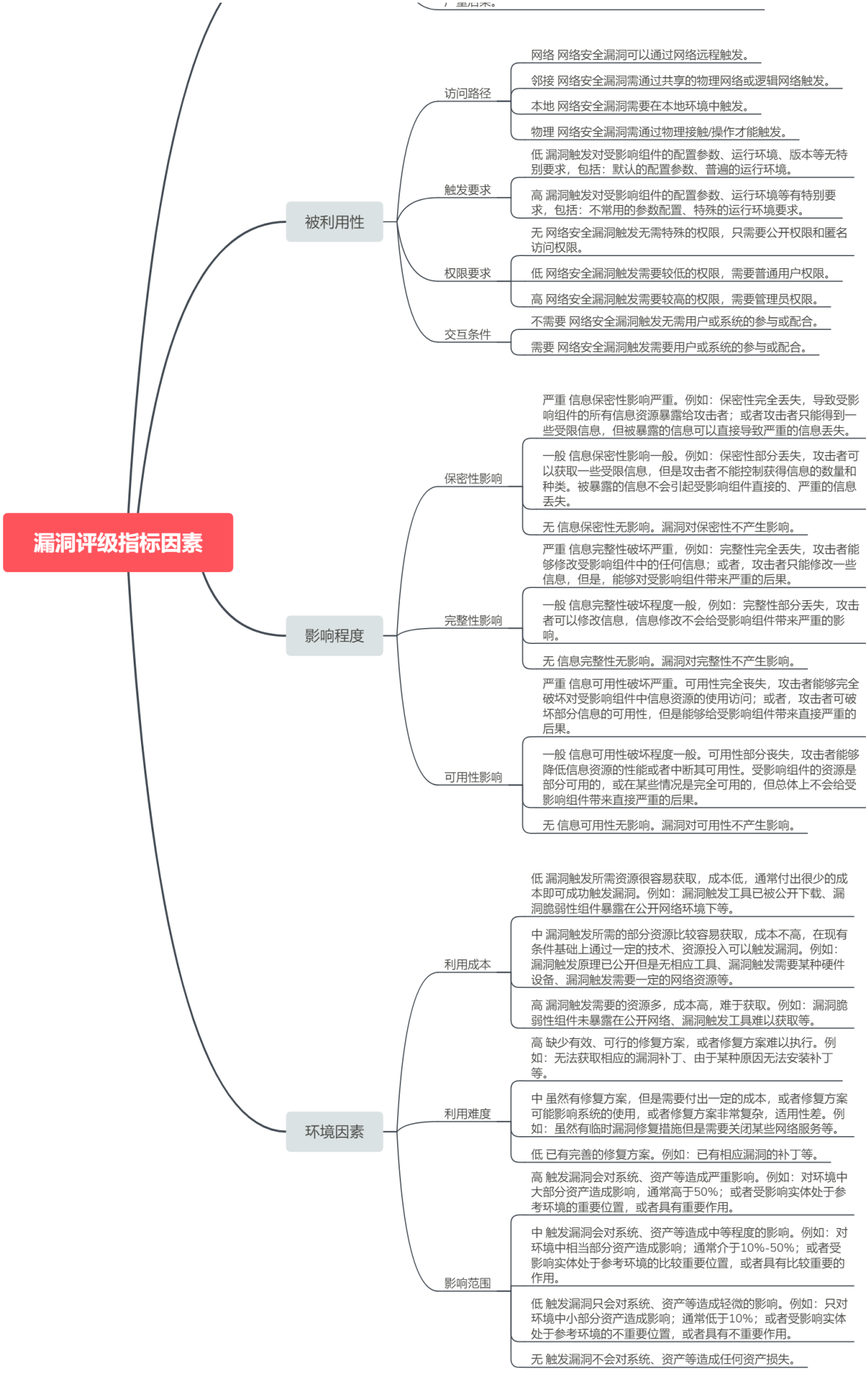
按漏洞成因和漏洞位置分类



2.4.2 漏洞分级

漏洞评级的相关因素




漏洞评级指标因素
被利用性
访问路径
网络 网络安全漏洞可以通过网络远程触发。
邻接 网络安全漏洞需通过共享的物理网络或逻辑网络触发。
本地 网络安全漏洞需要在本地环境中触发。
物理 网络安全漏洞需通过物理接触/操作才能触发。
触发要求
低 漏洞触发对受影响组件的配置参数、运行环境、版本等无特别要求，包括：默认的配置参数、普遍的运行环境。
高 漏洞触发对受影响组件的配置参数、运行环境等有特别要求，包括：不常用的参数配置、特殊的运行环境要求。
权限要求
无 网络安全漏洞触发无需特殊的权限，只需要公开权限和匿名访问权限。
低 网络安全漏洞触发需要较低的权限，需要普通用户权限。
高 网络安全漏洞触发需要较高的权限，需要管理员权限。
交互条件
不需要 网络安全漏洞触发无需用户或系统的参与或配合。
需要 网络安全漏洞触发需要用户或系统的参与或配合。
影响程度
保密性影响
严重 信息保密性影响严重。例如：保密性完全丢失，导致受影响组件的所有信息资源暴露给攻击者；或者攻击者只能得到一些受限信息，但被暴露的信息可以直接导致严重的信息丢失。
一般 信息保密性影响一般。例如：保密性部分丢失，攻击者可以获取一些受限信息，但是攻击者不能控制获得信息的数量和种类。被暴露的信息不会引起受影响组件直接的、严重的信息丢失。
无 信息保密性无影响。漏洞对保密性不产生影响。
完整性影响
严重 信息完整性破坏严重，例如：完整性完全丢失，攻击者能够修改受影响组件中的任何信息；或者，攻击者只能修改一些信息，但是，能够对受影响组件带来严重的后果。
一般 信息完整性破坏程度一般，例如：完整性部分丢失，攻击者可以修改信息，信息修改不会给受影响组件带来严重的影响。
无 信息完整性无影响。漏洞对完整性不产生影响。
可用性影响
严重 信息可用性破坏严重。可用性完全丧失，攻击者能够完全破坏对受影响组件中信息资源的使用访问；或者，攻击者可破坏部分信息的可用性，但是能够给受影响组件带来直接严重的后果。
一般 信息可用性破坏程度一般。可用性部分丧失，攻击者能够降低信息资源的性能或者中断其可用性。受影响组件的资源是部分可用的，或在某些情况是完全可用的，但总体上不会给受影响组件带来直接严重的后果。
无 信息可用性无影响。漏洞对可用性不产生影响。
环境因素
利用成本
低 漏洞触发所需资源很容易获取，成本低，通常付出很少的成本即可成功触发漏洞。例如：漏洞触发工具已被公开下载、漏洞脆弱性组件暴露在公开网络环境下等。
中 漏洞触发所需的部分资源比较容易获取，成本不高，在现有条件基础上通过一定的技术、资源投入可以触发漏洞。例如：漏洞触发原理已公开但是无相应工具、漏洞触发需要某种硬件设备、漏洞触发需要一定的网络资源等。
高 漏洞触发需要的资源多，成本高，难于获取。例如：漏洞脆弱性组件未暴露在公开网络、漏洞触发工具难以获取等。
利用难度
高 缺少有效、可行的修复方案，或者修复方案难以执行。例如：无法获取相应的漏洞补丁、由于某种原因无法安装补丁等。
中 虽然有修复方案，但是需要付出一定的成本，或者修复方案可能影响系统的使用，或者修复方案非常复杂，适用性差。例如：虽然有临时漏洞修复措施但是需要关闭某些网络服务等。
低 已有完善的修复方案。例如：已有相应漏洞的补丁等。
影响范围
高 触发漏洞会对系统、资产等造成严重影响。例如：对环境中大部分资产造成影响，通常高于50%；或者受影响实体处于参考环境的重要位置，或者具有重要作用。
中 触发漏洞会对系统、资产等造成中等程度的影响。例如：对环境中相当部分资产造成影响；通常介于10%-50%；或者受影响实体处于参考环境的比较重要位置，或者具有比较重要的作用。
低 触发漏洞只会对系统、资产等造成轻微的影响。例如：只对环境中小部分资产造成影响；通常低于10%；或者受影响实体处于参考环境的不重要位置，或者具有不重要作用。
无 触发漏洞不会对系统、资产等造成任何资产损失。

2.5 互联网公司的漏洞和威胁情报的分级分类

2.5.1 漏洞处理

1. **预报告阶段**：漏洞报告者前往指定反馈平台注册帐号。
2. **报告阶段**：报告者登录指定安全反馈平台，提交相关信息（状态：未审核）。
3. **处理阶段**：一个工作日内，工作人员会确认收到的报告并跟进开始评估问题（状态：审核中），三个工作日内工作人员处理问题、给出结论并计分（状态：已确认/已忽略）。必要时与报告者沟通确认，请报告者予以协助。
4. **修复阶段**：针对安全漏洞，业务部门修复漏洞并安排更新上线，修复时间根据问题点严重程度及修复难度而定，一般来说，严重漏洞24小时内，高危三个工作日内，中风险七个工作日内。客户端漏洞受版本发布限制，修复时间根据实际情况确定。由于情报分析调查的时间较长，因此确认周期相比漏洞的时长较长
5. **完成阶段**：完成处理后，更新处理状态，报告者可见更新状态。

2.5.1.1 漏洞分级

严重

漏洞容易直接或间接利用，利用后会对核心业务/核心服务器、生产环境用户数据造成严重的安全事故。包括但不限于：

1. 直接获取核心服务器权限的漏洞，包括但不限于任意代码执行、远程命令执行、上传WebShell并可执行、SQL注入获取系统权限、缓冲区溢出（包括可利用的ActiveX缓冲区溢出）等。
2. 生产业务系统严重的逻辑设计缺陷，包括但不限于账户、支付方面的安全问题，如：任意账户登录、任意账户密码修改、任意账户资金消费、支付交易方面的严重漏洞。
3. 严重的敏感信息泄漏，包括但不限于核心DB（资金、用户、交易相关）的SQL注入，可获取大量核心用户的身份信息、订单信息、银行卡信息等接口问题引起的敏感信息泄露。

高危

漏洞一旦被利用会导致业务系统或服务器被直接控制，存在批量数据泄漏、服务器权限被控制等风险。包括但不限于：

1. 重要敏感数据信息泄露，包括但不仅限于非核心DB SQL注入、源代码压缩包泄漏、服务器应用加密可逆或明文、移动API访问摘要、硬编码等问题引起的敏感信息泄露。
2. 敏感信息越权访问。包括但不仅限于绕过认证直接访问管理后台、后台弱密码、获取大量内网敏感信息的漏洞。
3. 越权敏感操作。包括但不仅限于账号越权修改重要信息、进行订单普通操作、重要业务配置修改等较为重要的越权行为。
4. 影响应用正常运转，造成不良影响的漏洞，包括但不限于应用层拒绝服务等。
5. 直接获取非核心业务系统权限的漏洞，包括但不限于可以利用的远程代码执行漏洞等。

中危

漏洞被利用后产生的影响在承受的范围内，且不会造成批量数据泄漏，受其他机制有效保护的且较难利用的高危漏洞。包括但不限于：

1. 普通信息泄露，包括但不仅限于客户端明文存储密码以及web路径遍历、系统路径遍历。
2. 普通越权操作，包括但不仅限于不正确的直接对象引用。
3. 需交互方可对用户产生危害的漏洞，包括但不仅限于一般页面的存储型XSS、反射型 XSS（包括反射型 DOM-XSS）、重要敏感操作CSRF。
4. 拒绝服务漏洞。包括但不限于导致网站应用拒绝服务等造成影响的远程拒绝服务漏洞。
5. 本地保存的敏感认证密钥信息泄漏且能做出有效利用。

低危

漏洞不会直接造成影响，以普通安全bug的形式存在，漏洞被利用后不会有用户或服务受到明显的影响。包括但不限于：

1. 轻微信息泄露，包括但不限于路径信息泄露、SVN信息泄露、PHPinfo、异常信息泄漏，以及客户端应用本地SQL注入（仅泄漏数据库名称、字段名、cache内容）、日志打印、配置信息、异常信息等。
2. 本地拒绝服务，包括但不限于客户端本地拒绝服务（解析文件格式、网络协议产生的崩溃），由Android组件权限暴露、普通应用程序权限引起的问题等。
3. 难以利用但存在安全隐患的漏洞。包括但不仅限于难以利用的SQL注入点、可引起传播和利用的Self-XSS、有一定影响的CSRF、URL跳转漏洞。

提示

1. 不涉及安全问题的 Bug，包括但不仅限于产品功能缺陷、网页乱码、样式混乱、静态文件目录遍历、应用兼容性等问题。
2. 无法利用的漏洞，包括但不仅限于Self-XSS、无敏感操作的 CSRF、无意义的异常信息泄漏、内网IP 地址/域名泄漏。
3. 不能直接反映漏洞存在的其他问题，包括但不仅限于纯属用户猜测的问题。

2.5.1.2 影响范围说明

关于业务规模

- 【大】业务中涉及用户、商户及资金等敏感数据的**核心应用业务**。
- 【中】业务中不涉及用户、商户、骑手及资金等敏感数据的**一般应用业务**。
- 【小】定义为**第三方供应系统**提供的系统。

2.5.1.3 漏洞描述

项目	说明
漏洞定义	攻击者通过操纵某些数据，使得程序偏离设计者的逻辑，进而引发的安全问题。
漏洞名称	PHPTEST v1.0.0前台无限制Getshell
漏洞分类	包括：XSS跨站；SQL注入；XXE；命令执行；文件包含；任意文件操作；权限绕过；存在后门；文件上传；逻辑漏洞；栈溢出；堆溢出；内存破坏；整数溢出；释放后重用；类型混淆；沙盒绕过；本地提权；拒绝服务；CRLF注入；SSRF；点击劫持；时间竞争漏洞；敏感信息泄露等
风险评价	严重/高/中/低/提示
利用方式	远程/本地/物理

项目	说明
用户交互	不需要登录/需登录/需登录(开放注册)
权限要求	访客/普通用户/功能管理员/系统管理员
利用接口	http://victim.com/show.php?id=100&cat=news
漏洞参数	id
漏洞证明	测试步骤及截图
风险描述	漏洞利用方法
修复建议	

关于漏洞风险级别
snipaste20190109_135535

可以参考：
https://www.first.org/cvss/specification-document

2.5.2 威胁情报处理

2.5.2.1 安全情报威胁系数

严重

1. 核心业务系统、生产及办公网络的入侵情报。如：内网漫游、核心生产服务器入侵、核心数据库的拖库等。
2. 核心业务造成重大影响的威胁组织活动情报。如：大规模刷单活动等。
3. 大规模敏感信息泄漏并验证真实有效的情报。如：用户信息、商户信息、订单信息、内部信息等。
4. 业务系统存在的未公开的0day漏洞情报。

高危

1. 非核心业务系统的入侵线索。
2. 新型可利用的工具、平台并提供完整可用的工具。如：黑产刷单工具等。
3. 内部机密泄漏情报。如：尚未公开的活动计划或者方案等。
4. 金融逻辑漏洞线索。如：支付相关产品的逻辑缺陷，商家恶意套现牟利手法等。

中危

1. 一般风险的业务安全问题。如：营销活动作弊、业务规则绕过等。
2. 新型可利用的工具、平台。如：扫号工具等。
3. 新型的攻击技术或攻击方法。

低危

1. 威胁组织基础信息。包括但不限于威胁组织相关人员、架构、规模、地域、活动情况等信息、交流及销售渠道、使用的工具和平台、造成的相关影响、行业动态等。
2. 低风险的业务安全问题。如：批量注册饿了么账户等。

2.5.2.2 情报完整性说明

由于情报的完整性对情报的价值有着重要的影响，因此上报情报的价值会进行情报完整性考量。情报完整性的评价会综合情报的多个方面进行考虑。

情报线索关键点包括：

1. 攻击者个人或者组织的信息，比如身份信息、联系方式、交流渠道等。
2. 攻击者的场景信息，比如产品或业务入口，页面地址等。
3. 攻击过程还原，比如绕过安全校验手法，新型刷单工具原理等。

2.5.2.3 无效情报

无效威胁情报是指：错误、无意义或根据提供信息无法调查利用的威胁情报，例如：

1. 上报虚假捏造或者无法还原的情报信息。
2. 只上报可能刷单、扫号的QQ群号或微信群，但未提供其他有效信息。
3. 上报单个或少量商铺的非业务规则问题导致的刷单行为。
4. 上报已过期、已失效的威胁情报。

3 漏洞批露注意的问题

供应商接受漏洞报告时应注意收集以下信息

- 受影响的产品/版本/URL
- 系统详细信息（操作系统等）
- 技术说明和复现步骤
- PoC
- 其他参与方/涉及的产品
- 披露计划/日期

漏洞响应能力

- 政策
    - 为什么要回应?
- 组织能力
    - 谁负责响应?
- 工程能力
    - 我们如何快速，有效，彻底地做出回应?
- 沟通能力
    - 我们的指导是否清晰及时?
- 分析能力
    - 我们如何从中吸取教训以防止相同的情况? 我们能否预测出有助于资源投资的趋势?

漏洞修复流程的沟通工作

- 与漏洞发现者沟通
    - 使用PGP等安全方法来传达技术细节
    - 传达修复时间表和计划
- 与产品和业务部门沟通
    - 为紧急情况和非紧急情况的内部团队制定SLA
    - 响应小组应更新漏洞的风险和影响范围
- 与协调员或其他供应商的沟通
    - 了解其他供应商的对应人员
    - 与受影响用户的沟通
- 建立可验证的沟通渠道，提醒用户注意漏洞通告

漏洞批露应该注意的问题

- 供应商应注意保持敏感漏洞信息的机密性。：
    - 与漏洞报告相关的任何个人信息（例如，被窃取的SSN或发现者的信息，如果他们希望保持匿名）
    - 尚未发布或广为人知的漏洞信息，目前还没有防御，例如技术细节使攻击者受益的信息
- 过早披露敏感漏洞信息会增加与供应商和用户披露相关的成本和风险。
    - 供应商应采取合理措施保护漏洞信息。

供应链问题

- 如果漏洞是另一个供应商供应链（上游或下游）的一部分，或者是多供应商问题
    - 协调：供应商应尽可能包括其他受影响的供应商讨论潜在解决方案
- 供应链/多供应商场景：
    - 由于底层操作系统或CPU等的露会影响特定平台
    - 有缺陷的标准功能规范或已发布的算法;
    - 常用库中的漏洞
    - 缺少当前维护者的软件组件中的漏洞。
- 灵活性至关重要!
    - 重点应放在最小化风险上

在线服务应注意的问题

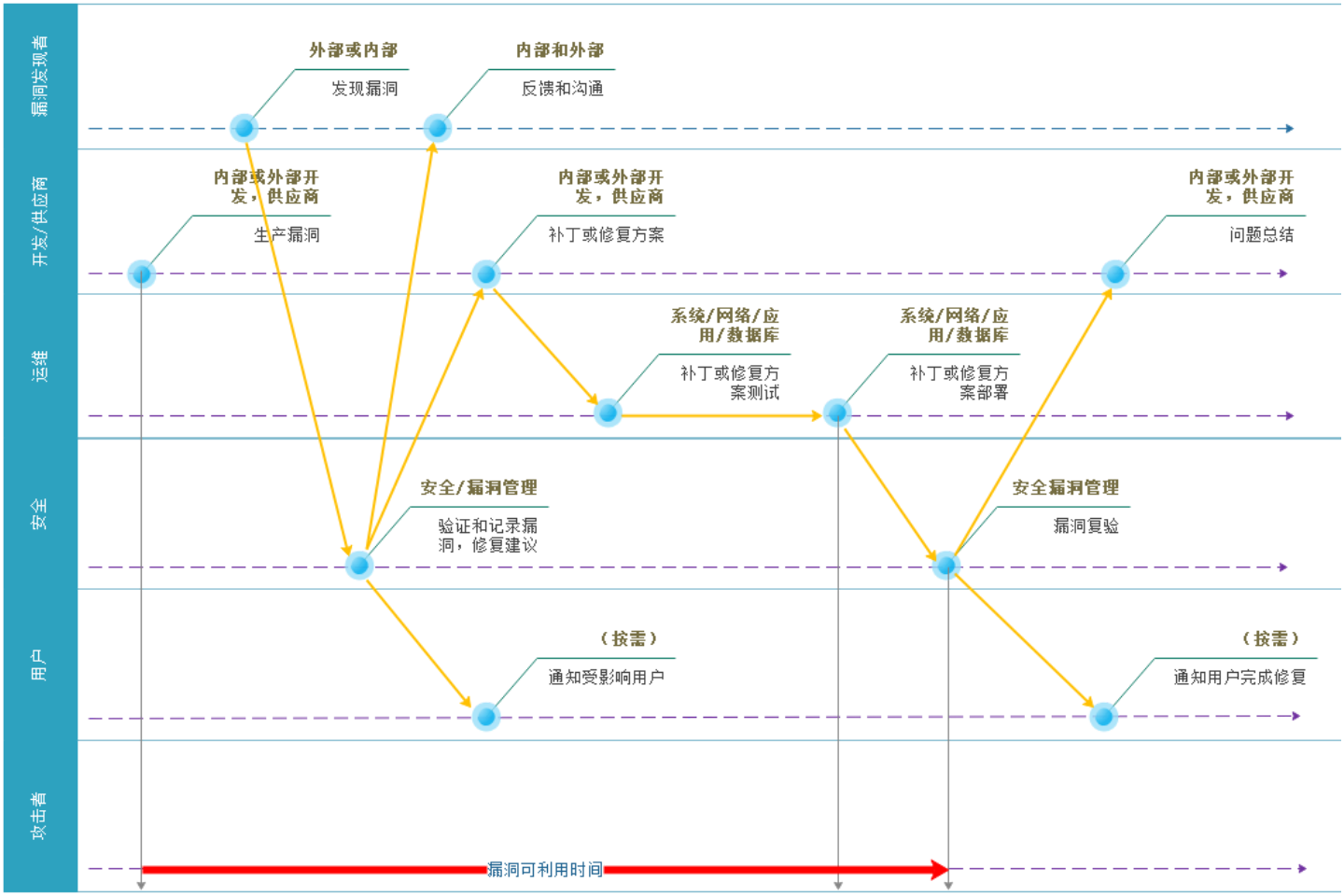
- 尽量避免漏洞的复现影响在线用户的个人信息
- 可以使用 2 个在线服务的测试账号
- 如果漏洞泄露用户个人信息，应该向通知到相关用户采取措施

4 如何设计组织内部的漏洞处理流程

4.1 漏洞修复过程责任分配

明确的责任分配是漏洞响应能力的基础，首先需要分析组织中涉及漏洞修复的部门和第三方有哪些，各个业务的主管是谁，谁在开发，谁在运维。由组织的漏洞修复策略来说明各个责任部门的相关责任和漏洞修复的时间要求等。其次需要统一漏洞定义语言，所有漏洞生命周期涉及到的人员对漏洞相关的概念理解一致。

通常漏洞生命周期内涉及到的部门有：



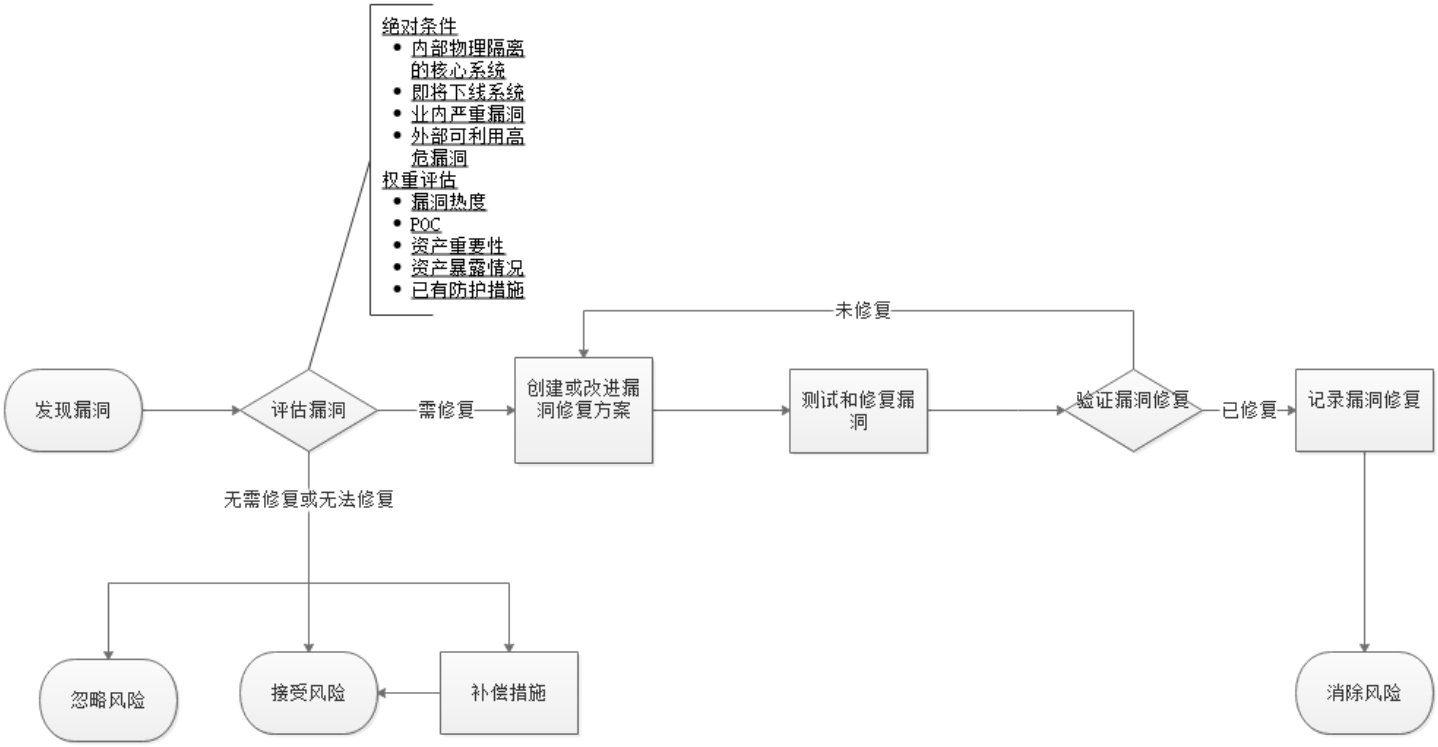
各部门的主要责任：

- 安全部门
  - 提供漏洞管理相关培训
  - 收集、发现和验证漏洞
  - 漏洞风险和影响性评估
  - 与外部漏洞发现者沟通
  - 提供漏洞修复建议
  - 提供漏洞修复补偿措施
  - 跟踪漏洞修复结果
- 开发/供应商
  - 提供漏洞修复方案或修复补丁
  - 漏洞总结
- 运维部门
  - 测试漏洞修复方案
  - 开发漏洞修复（变更）计划（业务、运维、安全）
  - 实施漏洞修复

漏洞的修复可能涉及第三方开发和运维，需要在合同中或补充条款中明确定义安全相关的责任，包括漏洞修复要求、保密要求等。

4.2 漏洞修复流程

所以常见的漏洞修复流程



可能漏洞的修复部署要走组织变更流程，组织可以根据自己的实际情况调整。

4.3 漏洞管理支持系统

主要是支持漏洞发现和导入汇总、展示、告警和跟踪漏洞修复记录的系统，可以结合资产管理一起来做，另外常见漏洞的修复建议和修复步骤可以做成知识库共享给所有相关部门。最好结合自己组织的工单系统实现漏洞的自动化跟踪。

一些开源的漏洞管理系统：

- SeMF
- 洞察
- ThreadFix
- DefectDojo

组织内有自己的开发能力建议开发合适自己组织的漏洞管理系统。