**渗透测试方案**

## 渗透测试概述

通过智能工具扫描与人工测试、分析的手段，以模拟黑客入侵的方式对服务目标系统进行模拟入侵测试，主要评估目标系统是否存在SQL注入、跨站脚本、跨站伪造请求、认证会话管理、弱口令、信息加密性、文件包含、目录浏览、不安全的跳转、溢出、上传、不安全的数据传输、未授权的访问等脆弱性问题,识别服务目标存在的安全风险，提交《渗透测试报告》。

* 由安全工程师模拟黑客的行为模式，采用黑客最可能采用的漏洞发现技术和尽可能多的攻击方法，对业务应用系统安全性进行深入分析，并提供相应的漏洞报告和解决建议；
* 参与渗透测试的人员需技术过硬、声誉良好。对于用户修复后的漏洞，由技术专家提供严格的复测，复测不通过则要督促信息系统开发人员进一步整改，直到漏洞被彻底修复；
* 渗透测试结束后，技术专家依据测试结果，提供整体性测试报告。

## 渗透测试测试评估

通过智能工具扫描与人工测试、分析的手段，以模拟黑客入侵的方式对服务目标系统进行模拟入侵测试，主要评估目标系统是否存在SQL注入、跨站脚本、跨站伪造请求、认证会话管理、弱口令、信息加密性、文件包含、目录浏览、不安全的跳转、溢出、上传、不安全的数据传输、未授权的访问等脆弱性问题,识别服务目标存在的安全风险，提交《渗透测试报告》。

由安全工程师模拟黑客的行为模式，采用黑客最可能采用的漏洞发现技术和尽可能多的攻击方法，对业务应用系统安全性进行深入分析，并提供相应的漏洞报告和解决建议；

参与渗透测试的人员需技术过硬、声誉良好。对于用户修复后的漏洞，由技术专家提供严格的复测，复测不通过则要督促信息系统开发人员进一步整改，直到漏洞被彻底修复；

渗透测试结束后，技术专家依据测试结果，提供整体性测试报告。

### 渗透测试概述

渗透测试是指安全工程师尽可能完整地模拟黑客使用的漏洞发现技术和攻击手段，对目标网络、系统、主机、应用的安全性作深入的探测，发现系统最脆弱的环节的过程。渗透测试能够直观的让管理人员知道自己网络所面临的问题。

由XXX渗透测试小组模拟黑客使用的漏洞发现工具，从攻击者的角度对目标系统的网络、主机系统、应用服务的安全性作深入的非破坏性检测，以发现系统的薄弱环节。渗透测试通常能以非常明显、直观的结果来反映出系统的安全现状。

### 渗透测试目的

渗透测试是站在实战角度对指定的目标系统进行的安全检测，可以让相关人员直观的了解到网络、系统、应用中隐含的漏洞和危害发生时可能导致的损失。通过渗透测试，可以获得如下增益：

* **安全缺陷**

从黑客的角度发现信息系统安全体系中的漏洞（隐含缺陷），协助明确目前降低风险的措施，为下一步的安全策略调整指明了方向。

* **测试报告**

能帮助用户方以实际案例的形式来说明目前安全现状，从而增加用户方对信息安全的认知度，提升用户方的风险危机意识，从而实现内部安全等级的整体提升。

* **交互式渗透测试**

渗透测试人员在用户方约定的范围、时间内实施测试，而用户方人员可以与此同时进行相关的检测监控工作，测试自己能否发现正在进行的渗透测试过程，从中真实的评估自己的检测预警能力。

### 渗透测试流程

**A、方案制定**

XXX获取到用户方的书面授权许可后，才进行渗透测试的实施。并且将实施范围、方法、时间、人员等具体的方案与用户方进行交流并得到用户方的认同。

在测试实施之前，XXX会做到让用户方对渗透测试过程和风险的知晓，使随后的正式测试流程都在单位的控制下。

**B、信息收集**

这包括：操作系统类型收集；网络拓扑结构分析；端口扫描和目标系统提供的服务识别等。可以采用一些商业安全评估系统（如：ISS、极光等）；免费的检测工具（NESSUS、Nmap等）进行收集。

**C、测试实施**

在规避防火墙、入侵检测、防毒软件等安全产品监控的条件下进行：操作系统可检测到的漏洞测试、应用系统检测到的漏洞测试（如：Web应用），此阶段如果成功的话，可能获得普通权限。

渗透测试人员可能用到的测试手段有：扫描分析、溢出测试、口令爆破、社会工程学、客户端攻击、中间人攻击等，用于测试人员顺利完成工程。在获取到普通权限后，尝试由普通权限提升为管理员权限，获得对系统的完全控制权。一旦成功控制一台或多台服务器后，测试人员将利用这些被控制的服务器作为跳板，绕过防火墙或其他安全设备的防护，从而对内网其他服务器和客户端进行进一步的渗透。此过程将循环进行，直到测试完成。最后由渗透测试人员清除中间数据。

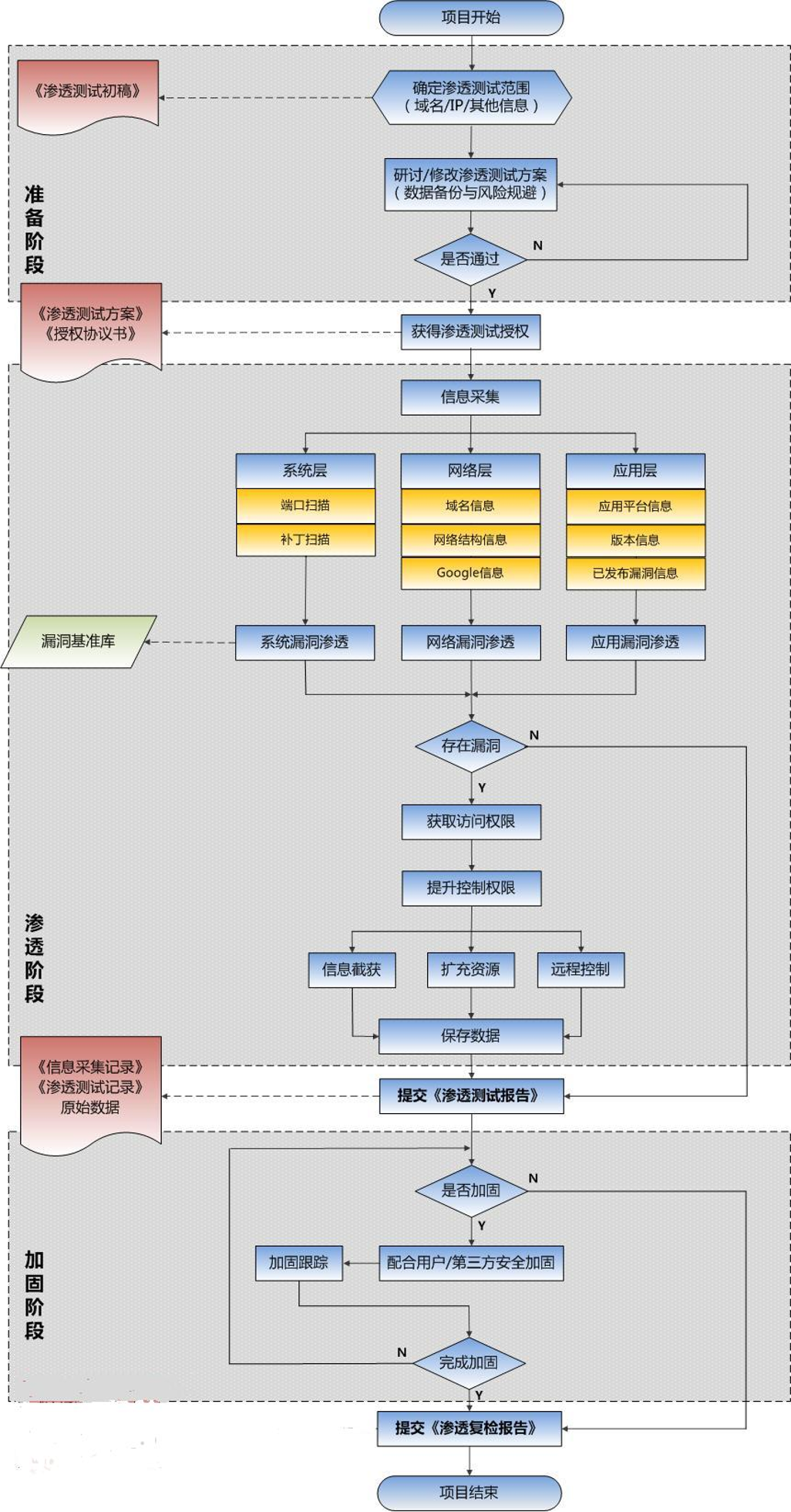
**D、报告输出**

渗透测试人员根据测试的过程结果编写直观的渗透测试服务报告。内容包括：具体的操作步骤描述；响应分析以及最后的安全修复建议。

**E、安全复查**

渗透测试完成后，XXX协助用户方对已发现的安全隐患进行修复。修复完成后，XXX渗透测试工程师对修复的成果再次进行远程测试复查，对修复的结果进行检验，确保修复结果的有效性。

下图是更为详细的步骤拆分示意图：



### 渗透测试内容

渗透测试内容包括但不限于：

| **渗透内容** | **内容描述** |
| --- | --- |
| 网络系统渗透 | 对网络拓扑进行发现，对网络设备（如防火墙、路由器、交换机）等进行漏洞检测，寻找网络构架的风险薄弱点 |
| 应用服务渗透 | 对应用服务（包括操作系统），等进行漏洞检测，获取或篡改关键信息，最大程度提升权限，据此形成专业的风险渗透报告 |
| 数据库渗透 | 对专用数据服务（包括操作系统），如SQL、Oracle等进行漏洞检测，获取或篡改关键信息，最大程度提升权限，据此形成专业的风险渗透报告 |
| 黑盒测试 | 黑盒测试渗透者完全处于对系统一无所知的状态，通常这类型测试，最初的信息获取来自于DNS、Web、Email及各种公开对外的服务器。 |
| 隐秘测试 | 隐秘测试是对被测单位而言的，通常情况下，接受渗透测试的单位网络管理部门会收到通知：在某些时段进行测试。因此能够监测网络中出现的变化。但隐秘测试则被测单位也仅有极少数人知晓测试的存在，因此能够有效地检验单位中的信息安全事件监控、响应、恢复做得是否到位。 |
| 攻击路径 | 测试目标不同，涉及需要采用的技术也会有一定差异，因此下面简单说明在不同位置、攻击路径不同时可能采用的技术。 |
| 内网测试 | 内网测试指的是渗透测试人员由内部网络发起测试，这类测试能够模拟企业内部违规操作者的行为。最主要的“优势”是绕过了防火墙的保护。内部主要可能采用的渗透方式：  1)远程缓冲区溢出；  2)口令猜测；  3)B/S或C/S应用程序测试（如果涉及C/S程序测试，需要提前准备相关客户端软件供测试使用）； |
| 外网测试 | 外网测试指的是渗透测试人员完全处于外部网络（例如拨号、ADSL或外部光纤），模拟对内部状态一无所知的外部攻击者的行为。  1)对网络设备的远程攻击；  2)口令管理安全性测试；  3)防火墙规则试探、规避；  4)Web及其它开放应用服务的安全性测试； |
| 不同网段/Vlan之间的渗透 | 这种渗透方式是从某内/外部网段，尝试对另一网段/Vlan进行渗透。这类测试通常可能用到的技术包括：  1)对网络设备的远程攻击；  2)对防火墙的远程攻击或规则探测、规避尝试； |

#### 渗透测试范围

OWASP（开放式Web应用程序安全项目）是一个开放的社区，由非营利组织 OWASP基金会支持的项目。对所有致力于改进应用程序安全的人士开放，旨在提高对应用程序安全性的认识。

其最具权威的就是“10项最严重的Web 应用程序安全风险列表” ，总结并更新Web应用程序中最可能、最常见、最危险的十大漏洞。

测试范围包含OWASP TOP 10漏洞列表如下：

| 序号 | 漏洞名称 | 攻击方式 | 漏洞原因 | 漏洞影响 |
| --- | --- | --- | --- | --- |
| 1 | 注入漏洞 | 在 2019、2020 的版本中都是第一名，可见此漏洞的引入是多么的容易，同时也证明此漏洞的危害有多么严重。 | 未审计的数据输入框  使用网址直接传递变量  未过滤的特殊字符  SQL 错误回显 | 获取敏感数据或进一步在服务器执行命令接管服务器 |
| 2 | 失效的身份认证 | 攻击者利用网站应用程序中的身份认证缺陷获取高权限并进行攻击应用服务 | 应用程序身份认证系统认证缺陷 | 盗用账号与身份  常见设计缺陷  修改利用网络协议数据包获取使用者账号密码  网站设计不良，可直接绕过验证页面  使用者忘记注销，而让攻击者有可趁之机 |
| 3 | 敏感数据泄露 | 常见的攻击方式主要是扫描应用程序获取到敏感数据 | 应用维护或者开发人员无意间上传敏感数据，如 github 文件泄露  敏感数据文件的权限设置错误，如网站目录下的数据库备份文件泄露  网络协议、算法本身的弱点，如 telent、ftp、md5 等 | 应用程序、网站被修改  个人资料、公司资料泄露，被用于售卖获利 |
| 4 | XML 外部实体漏洞 | 当应用程序解析 XML文件时包含了对外部实体的引用，攻击者传递恶意包含 XML 代码的文件，读取指定的服务器资源。 | XML 协议文档本身的设计特性，可以引入外部的资源；定义 XML 文件时使用的外部实体引入功能 | 读取服务器敏感资料，如、/etc/password  读取应用程序源码 |
| 5 | 无效的访问控制 | 未通过身份验证的用户始是恰当的访问控制，攻击者可以利用这些缺陷访问未经授权的功能或者数据 | 缺陷BUG | 绕过路径，如未读取的参数做检查，导致路径绕过读取到敏感文件  权限提升，如未对权限做检查，导致攻击者变更权限  垂直越权，攻击者可以从普通的用户权限提升到管理员的权限访问应用程序  水平越权，攻击者可以从普通用户A的权限提升到普通用户B的权限访问应用程序 |
| 6 | 安全配置错误 | 攻击者利用错误配置攻击，获取敏感数据或者提升权限 | 开发或者维护人员设置了错误的配置，如 python 开发中对于 Django 框架在生产环境启用了 Debug 模式 | 可让攻击者获取到敏感数据  可让攻击者提升权限，如未修改应用程序配置的默认密码，未删除应用程序安装程序目录文件等目录遍历 |
| 7 | 跨站脚本攻击 | 攻击者使用恶意字符嵌入应用程序代码中并运行，盗取应用程序数据 | 应用程序未对应用输入做过滤与检查，导致用户数据被当作代码执行。 | 欺骗使用者点击嵌入了恶意网站的正常网站，获取使用得的敏感数据  盗取使用者 cookie，冒用使用者身份 |
| 8 | 不安全的反序列化漏洞 | 攻击者利用应用程序反序列化功能，反序列化恶意对象攻击应用程序。 | 应用程序在反序列化数据对象时，执行了攻击者传递的恶意数据对象 | 最严重情况下，可导致远程代码执行 RCE注入攻击越权 |
| 9 | 使用含有已知漏洞的组件 | 利用应用程序技术栈中的框架、库、工具等的已知漏洞进行攻击，获取高权限或者敏感数据 | 应用程序技术栈中使用的框架、库、工具爆出了漏洞，应用程序未能及时更新与修复 | 敏感数据泄露  提升权限  远程代码执行 |
| 10 | 日志记录和监控不足导致的风险 | / | 对于日志记录的监控不足，造成攻击者攻击系统、应用、盗取数据等操作无法被发现和追查。 | 无法判断安全事件的发生  无法判断和修复漏洞  导致再次被入侵 |

#### Web测试内容

针对web应用类系统渗透测试包括但不限于以下内容：

| **类别** | **序号** | **测试项目** | **漏洞描述** |
| --- | --- | --- | --- |
| 信息收集 | 1.1 | 测试:蜘蛛，机器人和爬虫(OWASP-IG-001) |  |
| 1.2 | 搜索引擎发现/侦查(OWASP-IG-002) |  |
| 1.3 | 应用入口识别(OWASP-IG-003) |  |
| 1.4 | WEB 应用指纹测试(OWASP-IG-004) |  |
| 1.5 | 应用发现(OWASP-IG-005) |  |
| 1.6 | 错误代码分析(OWASP-IG-006) |  |
| 配置管理测试 | 2.1 | SSL/TLS 测试(OWASP-CM-001) |  |
| 2.2 | 数据库监听测试(OWASP-CM-002) |  |
| 2.3 | 基础结构配置管理测试(OWASP-CM-003) |  |
| 2.4 | 应用配置管理测试(OWASP-CM-004) |  |
| 2.5 | 文件扩展名处理测试(OWASP-CM-005) |  |
| 2.6 | 过时的、用于备份的以及未被引用的文件(OWASP-CM-006) |  |
| 2.7 | 基础结构和应用管理界面(OWASP-CM-007) |  |
| 2.8 | HTTP 方法和XST 测试(OWASP-CM-008) |  |
| 认证测试 | 3.1 | 加密信道证书传输(OWASP-AT-001) |  |
| 3.2 | 用户枚举测试(OWASP-AT-002) |  |
| 3.3 | 默认或可猜解(遍历)用户帐户(OWASP-AT-003) |  |
| 3.4 | 暴力破解测试(OWASP-AT-004) |  |
| 3.5 | 认证模式绕过测试(OWASP-AT-005) |  |
| 3.6 | 记住密码和密码重置弱点测试(OWASP-AT-006) |  |
| 3.7 | 注销和浏览器缓存管理测试(OWASP-AT-007) |  |
| 3.8 | CAPTCHA 测试(OWASP-AT-008) |  |
| 3.9 | 多因素认证测试(OWASP-AT-009) |  |
| 3.10 | 竞争条件测试(OWASP-AT-010) |  |
| 会话管理测试 | 4.1 | 会话管理模式测试(OWASP-SM-001) |  |
| 4.2 | COOKIES 属性测试(OWASP-SM-002) |  |
| 4.3 | 会话固定测试(OWASP-SM\_003) |  |
| 4.4 | 会话变量泄漏测试(OWASP-SM-004) |  |
| 4.5 | CSRF 测试(OWASP-SM-005) |  |
| 授权测试 | 5.1 | 路径遍历测试(OWASP-AZ-001) |  |
| 5.2 | 绕过授权模式测试(OWASP-AZ-002) |  |
| 5.3 | 提权测试(OWASP-AZ-003) |  |
| 业务逻辑测试 | 6.1 | 业务逻辑测试(OWASP-BL-001) |  |
| 数据验证测试 | 7.1 | 反射式跨站脚本测试(OWASP-DV-001) |  |
| 7.2 | 存储式跨站脚本测试(OWASP-DV-002) |  |
| 7.3 | 基于DOM 的跨站脚本检测(OWASP-DV-003) |  |
| 7.4 | FLASH 跨站脚本测试(OWASP-DV-004) |  |
| 7.5 | SQL 注入(OWASP-DV-005) |  |
| 7.6 | LDAP 注入(OWASP-DV-006) |  |
| 7.7 | ORM 注入(OWASP-DV-007) |  |
| 7.8 | XML 注入(OWASP-DV-008) |  |
| 7.9 | SSI 注入(OWASP-DV-009) |  |
| 7.10 | XPATH 注入(OWASP-DV-010) |  |
| 7.11 | IMAP/SMTP 注入(OWASP-DV-011) |  |
| 7.12 | 代码注入(OWASP-DV-012) |  |
| 7.13 | OS 指令执行(OWASP-DV-013) |  |
| 7.14 | 缓冲区溢出检测(OWASP-DV-014) |  |
| 7.15 | 潜伏式漏洞检测(OWASP-DV-015) |  |
| 7.16 | HTTP Splitting/Smuggling 测试(OWASP-DV-016) |  |
| 阻断服务测试 | 8.1 | SQL 通配符攻击测试(OWASP-DS-001) |  |
| 8.2 | 锁定用户账户(OWASP-DS-002) |  |
| 8.3 | 缓冲溢出(OWASP-DS-003) |  |
| 8.4 | 用户指定型对象分配(OWASP-DS-004) |  |
| 8.5 | 将用户输入作为循环计数器(OWASP-DS-005) |  |
| 8.6 | 将用户写入的数据写到磁盘(OWASP-DS-006) |  |
| 8.7 | 释放资源失败(OWASP-DS-007) |  |
| 8.8 | 存储过多会话数据(OWASP-DS-008) |  |
| WEB 服务测试 | 9.1 | WS 信息收集(OWASP-WS-001) |  |
| 9.2 | WSDL 测试(OWASP-WS-002) |  |
| 9.3 | XML 结构测试(OWASP-WS-003) |  |
| 9.4 | XML 内容级别测试(OWASP-WS-004) |  |
| 9.5 | HTTPGET 参数/REST 测试(OWASP-WS-005) |  |
| 9.6 | 调皮的SOAP 附件(OWASP-WS-006) |  |
| 9.7 | 重现测试(OWASP-WS-007) |  |
| AJAX 测试 | 10.1 | AJAX 漏洞(OWASP-AJ-001) |  |
| 10.2 | 检测AJAX (OWASP-AJ-002) |  |

#### 移动客户端测试内容

针对移动客户端类系统渗透测试包括但不限于以下内容：

| **序号** | **检测类别** | **检测项** | **风险等级** |
| --- | --- | --- | --- |
| 1.1 | 基本信息检测 | 应用包信息检测 | **--** |
| 1.2 | 应用签名信息检测 | **--** |
| 1.3 | 应用行为信息检测 | **--** |
| 2.1 | 客户端安全检测 | 反编译保护检测 |  |
| 2.2 | 完整性保护检测 |  |
| 2.3 | Activity组件检测 | 中危 |
| 2.4 | Service组件检测 | 安全 |
| 2.5 | Content Provider检测 | 中危 |
| 2.6 | Broadcast Receiver检测 | 安全 |
| 2.7 | 数据文件检测 | 安全 |
| 2.8 | Logcat日志检测 | 安全 |
| 2.9 | ROOT环境检测 | 低危 |
| 2.10 | 敏感权限检测 | 中危 |
| 2.11 | 恶意行为检测 | 安全 |
| 2.12 | 非法内容检测 | 安全 |
| 3.1 | 账号安全检测 | 密码复杂度检测 | 安全 |
| 3.2 | 密码修改验证检测 | 安全 |
| 3.3 | 双因子认证检测 | 不适用 |
| 3.4 | 账号锁定策略检测 | 安全 |
| 3.5 | 验证码安全检测 | 安全 |
| 3.6 | 账号登录限制检测 | 安全 |
| 3.7 | Activity界面劫持检测 | 中危 |
| 3.8 | 安全注销检测 | 安全 |
| 3.9 | 随机键盘检测 | 不适用 |
| 3.10 | 内存访问和修改检测 | 安全 |
| 4.1 | 通信安全检测 | 通信加密检测 | 安全 |
| 4.2 | 证书有效性检测 | 安全 |
| 5.1 | WEB应用安全检测 | SQL注入检测 | 安全 |
| 5.2 | XSS跨站脚本检测 | 安全 |
| 5.3 | 文件上传漏洞检测 | 安全 |
| 5.4 | 其他WEB漏洞检测 | 安全 |
| 6.1 | 业务逻辑安全检测 | 短信/邮件轰炸攻击检测 | 安全 |
| 6.2 | 越权操作检测 | 安全 |
| 6.3 | 交易篡改检测 | 安全 |
| 6.4 | 其他业务漏洞检测 | 安全 |

### 渗透测试方法

XXX提供的业务漏洞检测将采用的渗透测试方法如下。

1. **信息搜集**

信息探测阶段包括信息收集，端口、服务扫描，计算机漏洞扫描，此阶段主要做渗透前的踩点用。

使用工具：

* Nmap，端口、服务扫描，系统信息探测
* nessus，系统漏洞扫描
* 天镜，Web漏洞扫描程序
* WVS，Web漏洞扫描程序
* burp，抓包测试，业务逻辑

1. **端口扫描**

通过对目标地址的TCP/UDP端口扫描，确定其所开放的服务的数量和类型，这是所有渗透测试的基础。通过端口扫描，可以基本确定一个系统的基本信息，结合安全工程师的经验可以确定其可能存在以及被利用的安全弱点，为进行深层次的渗透提供依据。

1. **口令猜测**

口令猜测也是一种出现概率很高的风险，几乎不需要任何攻击工具，利用一个简单的暴力攻击程序和一个比较完善的字典，就可以猜测口令。

对一个系统账号的猜测通常包括两个方面：首先是对用户名的猜测，其次是对密码的猜测。

1. **脚本测试**

脚本测试专门针对Web服务器进行。根据最新的技术统计，脚本安全弱点为当前Web系统尤其存在动态内容的Web系统存在的主要比较严重的安全弱点之一。利用脚本相关弱点轻则可以获取系统其他目录的访问权限，重则将有可能取得系统的控制权限。因此对于含有动态页面的Web系统，脚本测试将是必不可少的一个环节。

1. **人工渗透**

人工渗透，主要针对应用系统的业务逻辑漏洞进行安全测试，利用业务逻辑漏洞查找可准确、切实的找出业务中存在的安全隐患，避免被恶意用户利用，对系统造成重大损失。

渗透测试是指安全工程师尽可能完整地模拟黑客使用的漏洞发现技术和攻击手段，对目标网络、系统、主机、应用的安全性作深入的探测，发现系统最脆弱的环节的过程。渗透测试能够直观的让管理人员知道自己网络所面临的问题。

由XXX渗透测试小组模拟黑客使用的漏洞发现工具，从攻击者的角度对目标系统的网络、主机系统、应用服务的安全性作深入的非破坏性检测，以发现系统的薄弱环节。渗透测试通常能以非常明显、直观的结果来反映出系统的安全现状。

### 渗透测试报告模板

渗透测试是站在攻击者角度，模拟黑客可能使用的攻击技术和漏洞发现技术对委托单位信息系统进行的安全性测试过程。通过本地通信抓包、数据包分析、模糊测试、重放分析、自动化漏洞扫描器等多种渗透手段对列目标信息系统进行测试。以发现目标信息系统中的安全漏洞和风险点，并撰写渗透测试报告、提出整改建议。

