附录F渗透测试报告

F.1 综述

F.1.1 概述

XXXX信息安全测试有限公司对123提供的123进行渗透性测试，利用各种主流的攻击技术对目标系统进行安全检查，以及时发现存在的安全漏洞和风险点。

F.1.2 测试目的

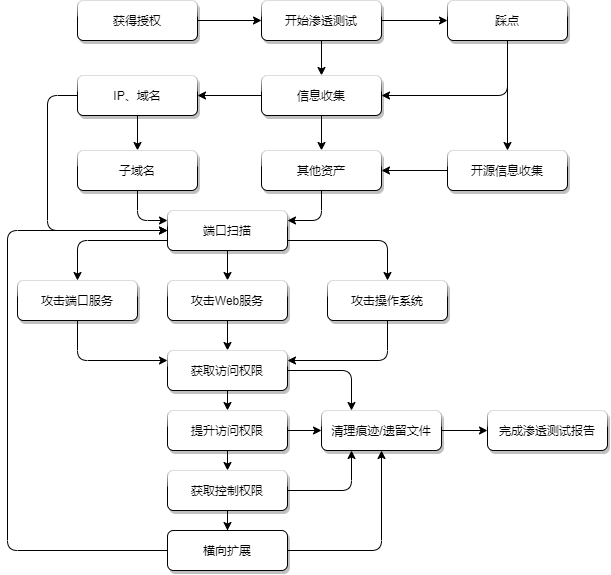
渗透测试是完全模拟黑客可能使用的攻击技术和漏洞发现技术，对目标系统的安全作深入的探测，发现系统最脆弱的环节。通过渗透测试能够直观的让管理人员知道自己网络所面临的安全问题，在攻击发生之前对系统进行修复和完善，做到防患于未然。

F.1.3 测试范围

本次江苏正信针对目标进行的渗透测试，将在可控的范围内模拟外部黑客攻击，主要安全渗透对象如下表所示：

|  |  |
| --- | --- |
| 对象 | 123 |
| 说明 | 黑盒测试 |

F.1.4 测试流程



F.1.5 测试方法

渗透测试过程借鉴了黑客攻击的手法和技巧，在可控的范围内为证明网络防御能够按照预订计划正常运行而提供的一种检测方法，它可以高度精确的反映客户系统面临的风险。在渗透测试的过程中，通过充分暴露和发掘潜在的漏洞，能直观的让管理人员知道自己维护的系统中仍然存在的安全缺陷。

渗透测试是模拟黑客攻击的测试行为，入侵者攻击途径是差异化的。本次安全渗透测试，将利用网络安全扫描器、专用安全测试工具并结合工程师丰富的渗透经验，对测试对象进行模拟攻击，将侵入系统、获取敏感信息等等过程和细节总结生成报告。

F.1.6 渗透测试参考标准

为高质量的完成此次渗透测试工作，找到更多有安全隐患，明确系统现在所面临的安全威胁，服务过程中渗透测试人员不仅要具备丰富的实战经验，还需要掌握一套完整和正确的渗透测试方法和标准。在本次渗透测试过程中，服务团队主要遵循以下安全标准及流程。

F.1.6.1 OWASP（Open Web Application Security Project）

Open Web Application Security Project（OWASP）是一个开源的、非盈利的全球性安全组织，OWASP 被视为 web 应用安全领域的权威参考，致力于应用软件的安全研究。目前 OWASP 全球拥有 140 个分会近四万名会员，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。美国联邦贸易委员会(FTC)强烈建议所有企业需遵循 OWASP 十大 WEB 弱点防护守则，此外，OWASP TOP 10 为 WEB 漏洞扫描器漏洞研发的主要标准。

F.1.6.2 WASC（Web Application Security Consortium）

Web Application Security Consortium（WASC），是一个由安全专家、行业顾问和诸多组织的代表组成的国际团体。WASC 组织的关键项目之一是“Web 安全威胁分类”，也就是将 Web 应用所受到的威胁、攻击进行说明并归纳成具有共同特征的分类。该项目的目的是针对 Web 应用的安全隐患，制定和推广行业标准术语。

F.1.6.3 ATT&CK（Adversarial Tactics, Techniques, and Common Knowledges）

ATT&CK模型是MITRE在洛克希德-马丁公司提出的KillChain模型的基础上，构建的一套更细粒度、更易共享的知识模型和框架。该框架理顺了攻击者渗透网络、入侵主机、提升权限、隐秘移动和渗漏数据的攻击链。

F.1.7 渗透测试工具

在渗透测试过程中所使用商业扫描工具和开源/免费软件渗透测试工具包括但不仅限于：

|  |  |  |
| --- | --- | --- |
| 工具名 | 版本 | 描述 |
| BurpSuite | V2.0 | 用于攻击Web应用程序的集成平台 |
| Nmap | V7.70 | 端口扫描工具 |
| Tenable Nessus | V8.7 | 漏洞扫描工具 |
| SNETCracker | V1.0 | 开源的口令爆破工具 |
| JSFinder | V1.0 | JS敏感信息扫描工具 |
| Sqlmap | V1.5.11.9#dev | 开源sql注入工具 |
| hashcat | V6.2.5 | 用于对散列算法进行密码恢复的工具 |
| Ffuf | V1.3.1 | 开源的目录扫描、枚举工具 |
| metasploit framework | V 6.1.16 | MSF是一套完善的渗透测试框架，功能强大 |
| F8x | V1.5.9 | 用于红队环境自动化部署,支持多种场景 |
| kitsune | V1.3.6 | 多功能聚合扫描器 |
| POC-HUB | V8.6.8 | CMS、中间件、框架漏洞库 |

F.1.8 遗留信息

渗透测试过程中未遗留任何威胁信息，未在目标系统遗留任何文件（如木马、后门）。

F.2 测试综述

F.2.1 测试对象与环境

IP/域名：

|  |
| --- |
| 123 |

发现的安全问题

|  |  |
| --- | --- |
| 主要安全问题 | 1.Redis未授权访问漏洞 |
| 2.Mongodb未授权访问漏洞 |
| 次要安全问题 | 无 |
| 其它问题 | 无 |

F.2.2 测试内容汇总

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项目 | 测试内容 | 测试目的 | 测试结果 |
| 身份验证 | 用户登录 | 检查用户登录功能是否安全，登录口令是否可被爆破 | 安全 |
| 用户注册 | 检查网站注册功能是否存在漏洞，是否可以注册已存在用户、null用户等，是否可以重复注册已有的账号 | 不涉及 |
| 密码修改与重置 | 检查密码修改、密码重置功能可能存在的安全问题 | 不涉及 |
| 验证码绕过 | 检查验证码功能是否可被绕过，可爆破 验证码是否泄露 | 安全 |
| 会话安全 | 会话固定 | 检查是否存在会话固定漏洞 | 不安全 |
| 会话失效 | 检查会话是否一定时间后超时失效、是否注销和退出后立即生效 | 安全 |
| 会话令牌分析 | 分析会话令牌，判断是否可预测、可逆向，是否可被恶意利用 | 安全 |
| CSRF跨站请求伪造 | 检查网站重要表单是否使用Referer头、图形验证码、短信验证等随机验证方式来避免CSRF中间人攻击 | 安全 |
| 输入处理 | SQL注入 | 对网站各功能参数进行注入测试，判断是否存在SQL注入漏洞 | 安全 |
| XSS跨站脚本攻击 | 对核心业务http请求中的各参数进行测试 判断是否存在XSS漏洞 | 安全 |
| 远程命令/代码执行 | 检查是否存在命令/代码执行漏洞 | 安全 |
| XXE注入 | 检查是否存在xml外部实体注入漏洞 | 安全 |
| SSRF服务端请求伪造 | 检查是否存在可攻击内网的SSRF漏洞 | 安全 |
| 任意文件读取 | 检查是否存在漏洞可读取任意文件 | 安全 |
| 任意文件上传 | 检查上传功能是否允许向服务器上传任意文件包括可执行文件、大型资源文件、钓鱼文件 | 安全 |
| 任意文件修改/删除 | 检查文件编辑模块是否存在删除、修改 任意文件的问题 | 安全 |
| URL重定向 | 检查是否存在URL跳转漏洞 | 安全 |
| 文件包含 | 检查是否存在文件包含漏洞 | 不涉及 |
| 中间件安全 | 存在安全风险的版本 | 检查所使用的中间件版本是否存在已知的安全风险 | 安全 |
| 错误页面自定义 | 检查网站是否自定义错误页面 | 安全 |
| 控制台弱口令 | 检查中间件控制台是否弱口令或存在漏洞 | 安全 |
| 目录浏览问题 | 检查中间件配置是否合规 | 安全 |
| 危险的http方法 | 检查中间件是否开启危险的http方法 | 安全 |
| 业务安全 | 短信接口存在缺陷 | 检查短信接口是否存在安全漏洞，如短信劫持、短信轰炸、短信伪造、短信绕过等漏洞等 | 安全 |
| 业务逻辑漏洞 | 检查网站业务处理流程，看流程中是否存在逻辑不严谨导致的安全隐患 | 安全 |
| 支付交易流程漏洞 | 检查支付交易功能是否存在安全漏洞，是否可通过篡改交易金额等方式实现0元购物 | 不涉及 |
| 薅羊毛漏洞 | 检查网站福利机制、兑换机制、套现机制等，看是否存在薅羊毛漏洞 | 不涉及 |
| 多步骤功能漏洞 | 检查多阶段功能，是否存在跳阶问题，如注册功能需要4步骤，检查是否可以绕过其中的部分步骤，实现注册 | 不涉及 |
| 客服、留言互动安全 | 检查网站互动模块是否存在安全漏洞，是否可通过向客服或留言板后台推送恶意代码 | 不涉及 |
| 访问控制 | 功能滥用 | 检查是否有功能模块由于开发不当可被非法利用 | 安全 |
| 平行越权漏洞 | 检查各功能模块用户权限控制问题，用户是否可以获取其他用户的权限 | 安全 |
| 垂直越权漏洞 | 检查各功能模块用户权限控制问题，用户是否可以获取更高权限用户的权限 | 安全 |
| 未授权访问 | 检查网站是否存在组件未授权访问漏洞 | 安全 |
| 信息泄露 | 敏感文件泄露 | 检查网站目录中是否存在网站备份文件、说明文件缓存文件、测试文件等，导致网站源码、配置信息泄露 | 安全 |
| 敏感注释信息 | 检查页面注释中是否包含敏感信息或测试代码 | 安全 |
| 管理后台地址泄露 | 检查网站后台路径是否进行隐藏 | 安全 |
| Git、svn、cvs安全 | 检查代码管理方式是否存在信息泄露等安全隐患 是否在互联网上泄露源码信息 | 安全 |
| 云主机安全 | 安全组配置 | 检查云主机安全组是否合理配置 | 安全 |
| 远程登录口令安全 | 检查云主机的登录口令是否存在弱口令 | 安全 |
| 非业务端口开放 | 检查云主机是否有非业务端口对外开放 | 安全 |
| 数据库登录安全 | 检查数据库是否允许外部远程连接 是否存在弱口令 | 安全 |
| 通信安全 | HTTPS协议传输 | 检查是否使用https加密传输 | 不安全 |
| 中间人劫持漏洞 | 检查网站数据传输是否存在中间人劫持风险 | 安全 |
| GET方式传输敏感数据 | 检查网站关键参数是否使用安全的post方式传输 | 安全 |
| 主机安全 | 操作系统安全 | 是否存在操作系统漏洞 | 安全 |

F.3 针对性检测结果

1. Redis未授权访问漏洞

测试接入点：

互联网

漏洞评级：

互联网

漏洞地址：

3

漏洞详情：

Redis默认情况下，会绑定在0.0.0.0:6379，如果没有进行采用相关的策略，比如添加防火墙规则避免其他非信任来源ip访问等，这样将会将Redis服务暴露到公网上，如果在没有设置密码认证（一般为空）的情况下，会导致任意用户在可以访问目标服务器的情况下未授权访问Redis以及读取Redis的数据。

造成的危害：

攻击者在未授权访问Redis的情况下，利用Redis自身的提供的config命令，可以进行写文件操作，攻击者可以成功将自己的ssh公钥写入目标服务器的/root/.ssh文件夹的authotrized\_keys文件中，进而可以使用对应私钥直接使用ssh服务登录目标服务器、添加计划任务、写入Webshell等操作。

利用过程：

3

3

修复建议：

1.禁用一些高危命令,常见如：flushdb，flushall，config，keys等。2.禁止使用root3.限启动redis服务，以低权限运行Redis服务。4.对redis访问启动密码认证。禁止外网访问Redis。5.添加IP访问限制，并更改默认6379端口。6.保证authorized\_keys文件的安全。

1. Mongodb未授权访问漏洞

测试接入点：

互联网

漏洞评级：

互联网

漏洞地址：

3

漏洞详情：

目标Mongodb服务存在未授权访问漏洞，开启MongoDB服务时不添加任何参数时,默认是没有权限验证的,登录的用户可以通过默认端口无需密码对数据库任意操作（增、删、改、查高危动作）而且可以远程访问数据库。

造成的危害：

使用默认空口令这将导致恶意攻击者无需进行账号认证就可以登录到数据服务器。

利用过程：

2

1

修复建议：

为MongoDB添加认证：MongoDB启动时添加--auth参数、为MongoDB添加用户。

F.3.X 【测试】2测试

测试接入点：

高危

测试详情：

测试过程中，对目标进行2测试。

测试目标：

2

测试截图：

2

测试结果：

经测试，未发现目标存在该漏洞。

F.4 安全控制建议

F.4.1 应用控制建议

1. 对网站的开发人员进行安全编码方面的培训，在开发过程避免漏洞的引入能起到事半功倍的效果。
2. 建议网络管理员、系统管理员、安全管理员关注安全信息、安全动态及最新的严重漏洞，特别是影响到Web站点所使用的系统和软件的漏洞，应该在事前设计好应对规划，一旦发现系统受漏洞影响及时采取措施。

F.4.2 主机控制建议

1. 建议网络管理员、系统管理员、安全管理员关注安全信息、安全动态及最新的严重漏洞。
2. 由于其他原因不能及时安装补丁的系统，考虑在网络边界、路由器、防火墙上设置严格的访问控制策略，以保证网络的动态安全。

F.5 渗透结果描述

本次对123123进行渗透测试，共涉及域名和IP共1个，通过渗透测试发现了2个安全漏洞，其中包含严重漏洞0个，高危漏洞2个，中危漏洞1个，低危漏洞1个，黑客成功利用这些漏洞后可能造成窃取管理员登录信息，成功入侵到管理系统后台，获取大量用户信息的问题。因此对系统整体安全风险评级为高危-不安全系统。

由于本次测试过程中采用保守测试方法，生产环境评估为高风险操作的功能，均与相关人员进行沟通，并在操作风险可控的情况下进行相关测试操作，以规避测试过程中的生产运营风险。