保密级别：商业机密

**深信服正式**

**渗透测试报告**



**深信服科技股份有限公司**

* **文档说明**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称 | 深信服正式渗透测试报告 | | |
| 保密级别 | 商密 | 文档版本号 | V1.0 |
| 制作人 | 王永乐 | 制作日期 | 2019-03-24 |
| 复审人 | 欧阳思裕 | 复审日期 | 2019-03-24 |
| 扩散范围 | 限“深信服”、深信服正式 | | |

* **适用范围**

本次渗透测试是由深信服正式授权，由深信服科技股份有限公司（以下简称“深信服”）对深信服正式进行的安全风险深度评估，根据评估结果提交技术报告，用于对该网站系统的安全状况做出安全评估和加固建议，仅限于“深信服”、深信服正式内部人员传阅。

* **版本变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 修改日期 | 版本 | 说明 | 修改人 |
|  |  |  |  |
|  |  |  |  |

目 录

[目 录 3](#_Toc532374016)

[报告摘要 4](#_Toc532374017)

[1. 渗透测试说明 6](#_Toc532374018)

[1.1 测试时间与人员 6](#_Toc532374019)

[1.2 测试范围 6](#_Toc532374020)

[1.3 漏洞评级标准 6](#_Toc532374021)

[1.4 综合风险评级标准 7](#_Toc532374022)

[2. 问题总览 7](#_Toc532374023)

[2.1 风险总览 7](#_Toc532374024)

[2.2 关键漏洞概况 8](#_Toc532374025)

[3. 渗透测试工作内容 9](#_Toc532374026)

[3.1 测试步骤 9](#_Toc532374027)

[3.2 测试工具 9](#_Toc532374028)

[3.3 测试漏洞种类 9](#_Toc532374029)

[4. 渗透测试漏洞细节 11](#_Toc532374030)

[4.1 xx系统 11](#_Toc532374031)

[4.1.1 XXXX漏洞 11](#_Toc532374032)

[4.1.2 任意帐号密码重置漏洞 11](#_Toc532374033)

[4.2 xx系统 13](#_Toc532374034)

[4.2.1 XXXX漏洞 13](#_Toc532374035)

[4.2.2 XXXX漏洞 14](#_Toc532374036)

[5. 致谢 14](#_Toc532374037)

[6. 了解更多 14](#_Toc532374038)

报告摘要

应深信服正式单位邀请，深信服安全服务团队于2019-03-22至2019-03-24，对深信服正式系统进行了全面的渗透测试。测试手段主要通过模拟黑客攻击手法对深信服正式系统开展测试并发现安全隐患；本次安全测试共发现了1个安全漏洞，按漏洞评级分布为：高危1个，中危0个，低危0个。

深信服安全服务团队发现存在的高危安全漏洞主要为：

1) 网站存在SQL注入漏洞。Web程序代码中对于用户提交的参数未做过滤就直接放到SQL语句中执行，导致参数中的特殊字符打破了SQL语句原有逻辑，黑客可以利用该漏洞执行任意SQL语句，如查询数据、下载数据、写入webshell、执行系统命令以及绕过登录限制等。

依据综合风险评级标准，本次深信服正式系统安全测试综合风险评级：高危

从本次渗透测试结果来看，深信服正式单位在安全防护方面存在不足，**难以抵御有组织的、高级别的安全攻击。**

建议深信服正式单位从以下三个方面进行安全整改，进一步提高网络安全防护和管理水平：

1. 在安全防护方面，建议完善安全基线并全面进行安全加固；加强办公终端管控，完善办公网络数据安全保护措施。
2. 在威胁发现方面，建议进一步加强APT攻击检测和威胁监测能力，加强日常安全检查力度，完善等级保护和风险评估机制，对安全加固和防护效果定期进行检查和评估，及时识别和消除风险。
3. 在安全管理方面，建议继续加强全体系信息安全工作的整体管理和组织协调，强化人员安全意识，严格落实各项规章制度，继续加强安全开发和上线前安全测试工作，不断提升运维管理水平。

# 渗透测试说明

## 1.1 测试时间与人员

本次渗透测试按照事先约定规避风险的时间段开展，如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **测试工作时间段** | | | |
| 起始时间 | 2019-03-22 | 结束时间 | 2019-03-24 |

本次渗透测试实施人员，如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参测人员名单** | | | | | |
| 姓名 | 王永乐 | 所属部门 | 深信服安全服务团队 | 联系方式 | 13423456123 |
| 姓名 | 尹普 | 所属部门 | 深信服安全服务团队 | 联系方式 | 13423456123 |

## 1.2 测试范围

本次渗透测试范围，如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **渗透测试范围** | | | |
| 编号 | 应用系统名称 | 应用系统URL | 应用系统IP |
| 1 | 门户系统 | http://www.sangfor.com.cn/ | 47.1.3.29 |

## 1.3 漏洞评级标准

|  |  |
| --- | --- |
| **漏洞评级标准** | |
| 漏洞级别 | 依据 |
| **高危** | 1）直接或间接获取服务器权限的漏洞。包括但不限于远程任意命令执行、代码执行、任意文件上传获取 webshell、缓冲区溢出、SQL注入获取系统权限、XXE执行系统命令等。  2）严重的信息泄漏漏洞。包括但不限于SQL注入漏洞获取数据、可获取大量企业核心业务数据等接口问题引起的敏感信息泄露。  3）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、重要后台弱密码、获取大量内网敏感信息的SSRF等。  4）严重的逻辑设计缺陷和流程缺陷。包括但不限于任意用户登录漏洞、任意帐号密码更改漏洞、绕过限制修改用户资料及执行用户操作、涉及金钱的越权操作、支付逻辑绕过等。  5)任意文件操作漏洞。包括但不限于任意文件读、写、删除、下载、任意文件包含等。  6）需要强烈的用户交互才能获取用户身份信息的漏洞，包括但不限于存储型跨站脚本漏洞，关键业务操作的CSRF漏洞等。 |
| **中危** | 1）比较严重的信息泄漏漏洞。包含敏感信息文件备份泄露、源代码泄露、数据库账号密码泄露、vpn账号泄露、UC-Key泄露、SVN信息泄漏、Github信息泄露、HEARTBLEED漏洞、DNS域传送漏洞等。  2）普通的逻辑设计缺陷和流程缺陷。包括但不限于未设置验证码或者验证码未刷新导致的撞库漏洞、会话管理类漏洞、任意次数短信发送、任意手机号或者邮箱注册。  3）特定类型的文件上传漏洞，包括但不限于上传html文件等。  4）需用户交互方可影响用户的漏洞。包括但不限于普通CSRF、DOM型跨站脚本漏洞等。 |
| **低危** | 1）普通信息泄漏漏洞。包括但不限于管理后台对外、明文传输密码、目录遍历、系统路径遍历、 IIS短文件名泄露、phpinfo、异常信息泄露、目标系统Banner信息可被识别等。  2）难以利用但存在安全隐患的漏洞、只能造成轻微影响的漏洞。包括但不限于3389对外开放、url跳转、反射型跨站脚本漏洞、难以利用的SQL注入点、crossdomain.xml配置问题等。 |

## 1.4 综合风险评级标准

|  |  |
| --- | --- |
| **综合风险评级标准** | |
| 风险级别 | 依据 |
| **高风险** | 存在1个及以上高危漏洞，或3个以上中危漏洞的系统 |
| **中风险** | 存在1个及以上中危漏洞，或5个以上低危漏洞的系统 |
| **低风险** | 存在5个以内低危漏洞，或未检测到漏洞的系统 |

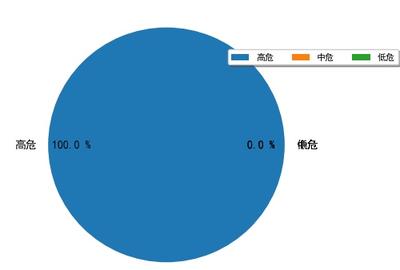
# 问题总览

## 2.1 漏洞总览

本次渗透测试总计发现漏洞1个，整体漏洞分布如下图所示：

|  |  |
| --- | --- |
| 漏洞级别 | 漏洞数量 |
| 高危 | 1 |
| 中危 | 0 |
| 低危 | 0 |

漏洞整体分布图



## 2.2 关键漏洞概况

本次渗透测试发现的高危漏洞如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **系统名称** | **漏洞名称** | **漏洞影响** | **漏洞级别** |
| 1 | 门户系统 | SQL注入漏洞 | Web程序代码中对于用户提交的参数未做过滤就直接放到SQL语句中执行，导致参数中的特殊字符打破了SQL语句原有逻辑，黑客可以利用该漏洞执行任意SQL语句，如查询数据、下载数据、写入webshell、执行系统命令以及绕过登录限制等。 | 高危 |

**所有发现漏洞的详细说明和建议请参见第4章节渗透测试漏洞细节。**

# 渗透测试工作内容

## 3.1 测试步骤

深信服安全服务团队对测试范围内的资产进行全面的评估。通过模拟黑客入侵的方式识别Web应用程序相关的安全漏洞，并提供解决此类漏洞的建议。渗透测试通过以下七个步骤进行：

|  |  |
| --- | --- |
| **编号** | **测试步骤** |
| 1 | 前期交互阶段：与客户进行交流、讨论，来确定渗透测试的范围与目标 |
| 2 | 情报搜集阶段：对目标客户的系统进行一系列踩点工作 |
| 3 | 威胁建模阶段：主要使用在情报搜集阶段所获取的信息，来标识出目标系统上可能存在的安全漏洞与弱点 |
| 4 | 漏洞分析阶段：主要是从前面几个环节获取的信息，并从中分析和理解哪些攻击途径可行 |
| 5 | 渗透攻击阶段：主要是针对目标系统实施已经经过了深入研究和测试的渗透攻击，进行针对性的安全测试 |
| 6 | 后渗透攻击阶段：从已经攻陷了客户的一些系统开始，将以特定业务系统为目标，标识出关键的基础设施，并寻找客户组织最具价值的信息和资产，并需要演示出能够对客户组织造成最重要业务影响的攻击途径 |
| 7 | 报告阶段：提交渗透测试报告，阐明客户系统中存在的安全隐患以及专业的修复建议 |

## 3.2 测试工具

我们的测试方法需要使用商业扫描工具和开源/免费软件渗透测试工具。用于测试的工具（包括且不限）如下所示：

► Tenable Nessus

► BurpSuite

► Appscan

► Nmap

► Sqlmap注入工具

► Firefox hackbar插件

## 3.3 测试漏洞种类

|  |  |  |
| --- | --- | --- |
| **测试大类** | **测试项** | **测试目的** |
| 身份验证类 | 用户注册 | 检查用户注册功能可能涉及的安全问题 |
| 用户登录 | 检查用户登录功能可能涉及的安全问题 |
| 修改密码 | 检查用户修改密码功能可能涉及的安全问题 |
| 密码重置 | 检查忘记密码、找回密码、密码重置功能可能涉及的安全问题 |
| 验证码绕过 | 检测验证码机是否合理，是否可以被绕过 |
| 用户锁定功能 | 测试用户锁定功能相关的安全问题 |
| 会话管理类 | Cookie重放攻击 | 检测目标系统是否仅依靠cookie来确认会话身份，从而易受到cookie回放攻击 |
| 会话令牌分析 | Cookie具有明显含义，或可被预测、可逆向，可被攻击者分析出cookie结构 |
| 会话令牌泄露 | 测试会话令牌是否存在泄露的可能 |
| 会话固定攻击 | 测试目标系统是否存在固定会话的缺陷 |
| 跨站请求伪造 | 检测目标系统是否存在CSRF漏洞 |
| 访问控制类 | 功能滥用 | 测试目标系统是否由于设计不当，导致合法功能非法利用 |
| 垂直权限提升 | 测试可能出现垂直权限提升的情况 |
| 水平权限提升 | 测试可能出现水平权限提升的情况 |
| 输入处理类 | SQL注入 | 检测目标系统是否存在SQL注入漏洞 |
| 文件上传 | 检测目标系统的文件上传功能是否存在缺陷 ，导致可以上传非预期类型和内容的文件 |
| 任意文件下载 | 检测目标系统加载/下载文件功能是否可以造成任意文件下载问题 |
| XML注入 | 测试目标系统-是否存在XML注入漏洞 |
| 目录穿越 | 测试目标系统是否存在目录穿越漏洞 |
| SSRF | 检测目标系统是否存在服务端跨站请求伪造漏洞 |
| 本地文件包含 | 测试目标站点是否存在LFI漏洞 |
| 远程文件包含 | 测试目标站点是否存在RFI漏洞 |
| 远程命令/代码执行 | 测试目标系统是否存在命令/代码注入漏洞 |
| 反射型跨站脚本 | 检测目标系统是否存在反射型跨站脚本漏洞 |
| 存储型跨站脚本 | 检测目标系统是否存在存储型跨站脚本漏洞 |
| DOM-based跨站脚本 | 检测目标系统是否存在DOM-based跨站脚本漏洞 |
| 服务端URL重定向 | 检查目标系统是否存在服务端URL重定向漏洞 |
| 信息泄露类 | error code | 测试目标系统的错误处理能力，是否会输出详尽的错误信息 |
| Stack Traces | 测试目标系统是否开启了Stack Traces 调试信息 |
| 敏感信息 | 尽量收集目标系统的敏感信息 |
| 第三方应用类 | 中间件 | 测试目标系统是否存在jboss、weblogic、tomcat等中间件 |
| CMS | 测试目标系统是否存在dedecms、phpcms等CMS |

1. 渗透测试漏洞细节

4.1 门户系统

4.1.1 SQL注入漏洞

|  |
| --- |
| 漏洞级别-高危 |
| SQL注入漏洞 |

|  |  |
| --- | --- |
| URL/IP | http://www.sangfor.com.cn/index.php |
| 漏洞描述 | Web程序代码中对于用户提交的参数未做过滤就直接放到SQL语句中执行，导致参数中的特殊字符打破了SQL语句原有逻辑，黑客可以利用该漏洞执行任意SQL语句，如查询数据、下载数据、写入webshell、执行系统命令以及绕过登录限制等。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 代码层最佳防御sql漏洞方案：采用sql语句预编译和绑定变量，是防御sql注入的最佳方法。  （1）所有的查询语句都使用数据库提供的参数化查询接口，参数化的语句使用参数而不是将用户输入变量嵌入到SQL语句中。当前几乎所有的数据库系统都提供了参数化SQL语句执行接口，使用此接口可以非常有效的防止SQL注入攻击。  （2）对进入数据库的特殊字符（’”<>&\*;等）进行转义处理，或编码转换。  （3）确认每种数据的类型，比如数字型的数据就必须是数字，数据库中的存储字段必须对应为int型。  （4）数据长度应该严格规定，能在一定程度上防止比较长的SQL注入语句无法正确执行。  （5）网站每个数据层的编码统一，建议全部使用UTF-8编码，上下层编码不一致有可能导致一些过滤模型被绕过。  （6）严格限制网站用户的数据库的操作权限，给此用户提供仅仅能够满足其工作的权限，从而最大限度的减少注入攻击对数据库的危害。  （7）避免网站显示SQL错误信息，比如类型错误、字段不匹配等，防止攻击者利用这些错误信息进行一些判断。 |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82718718 |

1. 致谢

深信服安全服务团队感谢深信服正式在渗透测试过程中进行沟通、协调的部门和人员,是你们的大力配合，使得我们的工作能够顺利完成。

1. 了解更多

了解更多安全信息，或关于本文出现的漏洞、攻击方式等详细介绍与建议，可查看深信服安全中心的威胁维基或关注深信服科技公众号了解最新的安全情报。  
 威胁维基：http://sec.sangfor.com.cn/vulns/lst.html  
 公众号及微博：

