保密级别：商业机密

**场上的1**

**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 | 场上的1渗透测试报告 | | |
| 保密级别 | 商密 | 文档版本号 | V1.0 |
| 制作人 | 尹普 | 制作日期 | 2019-02-16 |
| 复审人 | 欧阳思裕 | 复审日期 | 2019-02-16 |
| 扩散范围 | 限“深信服”、场上的1 | | |

* **适用范围**

本次渗透测试是由场上的1授权，由深信服科技股份有限公司（以下简称“深信服”）对场上的1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”、场上的1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

应场上的1单位邀请，深信服安全服务团队于2019-02-10至2019-02-16，对场上的1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场上的1系统开展测试并发现安全隐患；本次安全测试共发现了9个安全漏洞，按漏洞评级分布为：高危2个，中危4个，低危3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) 网站存在存储型XSS漏洞。Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。2) 网站存在存储型XSS漏洞。Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。

依据综合风险评级标准，本次场上的1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，场上的1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。**

建议场上的1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 | 2019-02-10 | 结束时间 | 2019-02-16 |

本次渗透测试实施人员，如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参测人员名单** | | | | | |
| 姓名 | 尹普 | 所属部门 | 深信服安全服务团队 | 联系方式 | 13463472821 |
| 姓名 | 王永乐 | 所属部门 | 深信服安全服务团队 | 联系方式 | 13463472822 |

## 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 | 为1 | http://127.0.0.1:5000/report/add | 1.1.1.1 |
| 2 | 为1231 | http://127.0.0.1:52000/report/add | 1.1.1.11 |

## 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 |

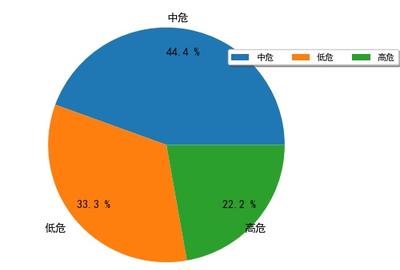
# 问题总览

## 2.1 漏洞总览

本次渗透测试总计发现漏洞9个，整体漏洞分布如下图所示：

|  |  |
| --- | --- |
| 漏洞级别 | 漏洞数量 |
| 高危 | 2 |
| 中危 | 4 |
| 低危 | 3 |

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 | 系统1 | 存储型XSS漏洞 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 | 高危 |
| 2 | 系统1 | 存储型XSS漏洞 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 | 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 为1

4.1.1 暴力破解

|  |
| --- |
| 漏洞级别-低危 |
| 暴力破解 |

|  |  |
| --- | --- |
| URL/IP | http://127.0.0.1:5000/report/ad |
| 漏洞描述 | 由于没有对登录页面进行相关的防暴力破解机制，如无验证码、有验证码但验证码未在服务器端校验以及无登录错误次数限制等，导致攻击者可通过暴力破解获取用户登录账户及口令，从而获取网站登录访问权限！ |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 1、添加验证码机制，加入图片（验证码动态生成且满足随机性）或者短信验证码（验证码具备超时时限一般为1分钟，且在该时限内错误次数超过3次则进行锁定1分钟后方能重新获取验证码，超时后验证码自动失效）！  2、验证码必须在服务器端进行校验，客户端的一切校验都是不安全的！ |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82806181 |

4.1.2 暴力破解

|  |
| --- |
| 漏洞级别-低危 |
| 暴力破解 |

|  |  |
| --- | --- |
| URL/IP | http://127.0.0.1:5000/report/ad |
| 漏洞描述 | 由于没有对登录页面进行相关的防暴力破解机制，如无验证码、有验证码但验证码未在服务器端校验以及无登录错误次数限制等，导致攻击者可通过暴力破解获取用户登录账户及口令，从而获取网站登录访问权限！ |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 1、添加验证码机制，加入图片（验证码动态生成且满足随机性）或者短信验证码（验证码具备超时时限一般为1分钟，且在该时限内错误次数超过3次则进行锁定1分钟后方能重新获取验证码，超时后验证码自动失效）！  2、验证码必须在服务器端进行校验，客户端的一切校验都是不安全的！ |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82806181 |

4.1.3 存储型XSS漏洞

|  |
| --- |
| 漏洞级别-高危 |
| 存储型XSS漏洞 |

|  |  |
| --- | --- |
| URL/IP | http://127.0.10.1:500/1.aspx |
| 漏洞描述 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义    （1）假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。    （2）不仅要验证数据的类型，还要验证其格式、长度、范围和内容。    （3）不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。    （4）对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。 |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82718255 |

4.1.4 存储型XSS漏洞

|  |
| --- |
| 漏洞级别-高危 |
| 存储型XSS漏洞 |

|  |  |
| --- | --- |
| URL/IP | http://127.0.10.1:500/1.aspx |
| 漏洞描述 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义    （1）假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。    （2）不仅要验证数据的类型，还要验证其格式、长度、范围和内容。    （3）不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。    （4）对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。 |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82718255 |

4.1.5 暴力破解

|  |
| --- |
| 漏洞级别-低危 |
| 暴力破解 |

|  |  |
| --- | --- |
| URL/IP | http://127.0.0.1:5000/report/ad |
| 漏洞描述 | 由于没有对登录页面进行相关的防暴力破解机制，如无验证码、有验证码但验证码未在服务器端校验以及无登录错误次数限制等，导致攻击者可通过暴力破解获取用户登录账户及口令，从而获取网站登录访问权限！ |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 1、添加验证码机制，加入图片（验证码动态生成且满足随机性）或者短信验证码（验证码具备超时时限一般为1分钟，且在该时限内错误次数超过3次则进行锁定1分钟后方能重新获取验证码，超时后验证码自动失效）！  2、验证码必须在服务器端进行校验，客户端的一切校验都是不安全的！ |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82806181 |

4.1.6 IIS短文件名泄露漏洞

|  |
| --- |
| 漏洞级别-低危 |
| IIS短文件名泄露漏洞 |

|  |  |
| --- | --- |
| URL/IP | https://www.baidu.com/1.aspx |
| 漏洞描述 | Internet Information Services（IIS，互联网信息服务）是由微软公司提供的基于运行Microsoft Windows的互联网基本服务。 Microsoft IIS在实现上存在文件枚举漏洞，攻击者可利用此漏洞枚举网络服务器根目录中的文件。 危害：攻击者可以利用“~”字符猜解或遍历服务器中的文件名，或对IIS服务器中的.Net Framework进行拒绝服务攻击。黑客可通过该漏洞尝试获取网站服务器下存放文件的文件名，达到获取更多信息来入侵服务器的目的。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 修改Windows配置，关闭短文件名功能。  1.关闭NTFS 8.3文件格式的支持。该功能默认是开启的，对于大多数用户来说无需开启。  2.如果是虚拟主机空间用户,可采用以下修复方案：    1）修改注册列表HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisable8dot3NameCreation的值为1(此修改只能禁止NTFS8.3格式文件名创建,已经存在的文件的短文件名无法移除)。    2）如果你的web环境不需要asp.net的支持你可以进入Internet 信息服务(IIS)管理器 --- Web 服务扩展 - ASP.NET 选择禁止此功能。    3）升级net framework 至4.0以上版本。  3.将web文件夹的内容拷贝到另一个位置，比如D:\www到D:\www.back，然后删除原文件夹D:\www，再重命名D:\www.back到D:\www。如果不重新复制，已经存在的短文件名则是不会消失的。 |
| 参考链接 | https://www.freebuf.com/articles/web/172561.html |

4.1.7 IIS短文件名泄露漏洞

|  |
| --- |
| 漏洞级别-低危 |
| IIS短文件名泄露漏洞 |

|  |  |
| --- | --- |
| URL/IP | https://www.baidu.com/1.aspx |
| 漏洞描述 | Internet Information Services（IIS，互联网信息服务）是由微软公司提供的基于运行Microsoft Windows的互联网基本服务。 Microsoft IIS在实现上存在文件枚举漏洞，攻击者可利用此漏洞枚举网络服务器根目录中的文件。 危害：攻击者可以利用“~”字符猜解或遍历服务器中的文件名，或对IIS服务器中的.Net Framework进行拒绝服务攻击。黑客可通过该漏洞尝试获取网站服务器下存放文件的文件名，达到获取更多信息来入侵服务器的目的。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | 修改Windows配置，关闭短文件名功能。  1.关闭NTFS 8.3文件格式的支持。该功能默认是开启的，对于大多数用户来说无需开启。  2.如果是虚拟主机空间用户,可采用以下修复方案：    1）修改注册列表HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisable8dot3NameCreation的值为1(此修改只能禁止NTFS8.3格式文件名创建,已经存在的文件的短文件名无法移除)。    2）如果你的web环境不需要asp.net的支持你可以进入Internet 信息服务(IIS)管理器 --- Web 服务扩展 - ASP.NET 选择禁止此功能。    3）升级net framework 至4.0以上版本。  3.将web文件夹的内容拷贝到另一个位置，比如D:\www到D:\www.back，然后删除原文件夹D:\www，再重命名D:\www.back到D:\www。如果不重新复制，已经存在的短文件名则是不会消失的。 |
| 参考链接 | https://www.freebuf.com/articles/web/172561.html |

4.1.8 DOM型XSS

|  |
| --- |
| 漏洞级别-中危 |
| DOM型XSS |

|  |  |
| --- | --- |
| URL/IP | http://127.0.0.1:5000/report/add |
| 漏洞描述 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义    （1）假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。    （2）不仅要验证数据的类型，还要验证其格式、长度、范围和内容。    （3）不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。    （4）对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。 |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82718255 |

4.1.9 DOM型XSS

|  |
| --- |
| 漏洞级别-中危 |
| DOM型XSS |

|  |  |
| --- | --- |
| URL/IP | http://127.0.0.1:5000/report/add |
| 漏洞描述 | Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。 |

|  |
| --- |
| 漏洞验证过程： |

|  |  |
| --- | --- |
| 处置建议 | xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义    （1）假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。    （2）不仅要验证数据的类型，还要验证其格式、长度、范围和内容。    （3）不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。    （4）对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。 |
| 参考链接 | https://blog.csdn.net/wutianxu123/article/details/82718255 |

1. 致谢

深信服安全服务团队感谢场上的1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：

