# XXX系统产品安全测试报告

## 介绍

### 编写目的

本报告为XXX系统产品的安全测试报告，目的在考察软件安全性、测试结论以及测试建议。

## 测试概要

### 测试方法和测试工具

本次安全测试，主要使用了账号安全管理、权限管理、安全日志、访问控制安全、输入安全、缓冲区溢出、SQL注入、跨站脚本攻击等安全测试方案。

针对以上提供的测试方案进行对应测试用例以及测试脚本编写,并使用APPScan作为安全测试工具。

#### 账号安全管理

1. 账号的惟一性：系统中账号名称具有惟一性
2. 普通用户、工程师、管理员具有惟一性；
3. 是否有可能绕过系统的账号惟一性校验
4. 账号不能写死在代码中，须提供可管理机制

#### 权限管理

1. 采用基于角色的账号权限管理模型
2. 授权和用户角色数据存放在服务器端

#### 安全日志

1. 对安全事件以及操作事件记录日志
2. 检查系统是否对以上所以操作记录相应日志记录，包括用户ID、用户IP、时间、事件类型、操作来源（APP、网页）等；

#### 访问控制安全

需要验证用户身份以及权限的页面，复制该页面的url地址，关闭该页面以后，查看是否可以直接进入该复制好的地址；

例：从一个页面链接到另一个页面的间隙可以看到url地址，直接输入该地址，可以看到自己没有权限的页面信息；

#### 输入安全

XXX系统系统主要对没有被验证的输入进行如下测试：

数据类型（字符串、整形、实数等）允许的字符集、最小和最大的长度、是否允许空输入、参数是否为必须、是否允许重复、数值范围、特定的值（枚举型）特定的模式（正则表达式）等；

#### 缓冲区溢出

没有加密关键数据：

例：view-source：http地址可以查看源代码

在页面输入密码，页面显示为加密字符\*\*\*\*，右键鼠标，查看源文件就可以看到刚刚输入的密码。

#### SQL注入

例：一个验证用户登录的页面

如果使用的sql语句为：

Select \* from A where username=’ ’ + username+’ ’ and password……

SQL输入or 1=1——就可以不输入任何password进行攻击，或者是半角状态下的用户名与密码均为：‘or’‘=’。

#### 跨站点脚本攻击

分析：攻击者使用跨站脚本来发送恶意代码给没有发觉的用户，窃取他机器上的任意资

料。

测试方法：

* HTML标签：<…>…</…>
* 转义字符：&（&）；<(<); >(>);(空格)；
* 脚本语言：

<script. Language=‘javascript’>

…Alert(‘’)

</script>

* 特殊字符：‘’<> /
* 最小和最大的长度
* 空输入

#### 测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| **工具名称** | **用途** | **生产厂商** | **版本** |
| APPScan | 自动化安全测试 | IBM | 9.3.0.5 |

## 测试组织

### 测试人员

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **角色** | **职责** |
| 1 | 邓臻 | 测试组长 | 环境搭建、用例审核、进度把控、报告编写 |
| 2 | 梁烨 | 测试组员 | 用例编写、用例执行 |
| 3 | 廖素萍 | 测试组员 | 用例编写、用例执行 |
| 4 | 何锦珊 | 测试组员 | 用例编写、用例执行 |
| 5 | 杨欢 | 测试组员 | 用例编写、用例执行 |

## 测试结果及缺陷分析

### 遗留缺陷列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **缺陷ID** | **缺陷概要** | **遗留原因分析** | **预防与改进措施** | **修改情况** |
|  | 登录页面填写密码后：鼠标右键查看源代码可直接查看到登录密码 | 无加密，只覆盖到了程序表面 | 加密 | 已关闭 |
|  | 复制链接可直接接入：复制管理员账号管理的链接，使用普通用户登录系统，进入此链接，进入成功。 | 对权限控制没做限制处理 | 控制用户权限，无权限的一律显示无权限页面。 | 已关闭 |
|  | SQL盲注 endTime | 未对用户输入正确执行危险字符清理 | 输出编码：在有风险的情况下仍需要使用动态生成的查询字符串或命令，对参数正确的加引号并将这些参数中任何特殊字符转义。 | 已关闭 |
|  | 存储的跨站点脚本编制 | 未对用户输入正确执行危险字符清理 | 使用不允许此弱点的经过审核的库或框架。 | 已关闭 |
|  | 跨站点请求伪造 | 应用程序使用的认证方法不充分 | 使用两次提交的cookie方法 | 已关闭 |
|  | SQL注入文件写入 | 未对用户输入正确执行危险字符清理 | 过滤危险字符 | 已关闭 |
|  | Blind SQL注入（基于时间） | 未对用户输入正确执行危险字符清理 | 使用自动实施数据和代码之间的分离的结构化机制 | 已关闭 |

## 测试结论

1. 本次测试覆盖全面，测试数据基础合理，测试有效。
2. SQL注入测试，已执行测试用例，问题回归后测试通过。
3. 跨站点脚本测试，测试发现已对相关特殊字符进行转义，测试通过。
4. 权限测试，已严格对相关角色进行权限控制，测试通过。

综合以上结论得出本次安全测试通过。

## 测试建议

在系统研发时便需要逐步对相关字符进行转义，可列成安全文档，定义如何防范各种安全漏洞，以便在开发项目阶段直接杜绝问题。



**扫描二维码，免费领取《400页800题软件测试面试宝典》，**

**更多测试干货也可随时咨询领取**