受 控：C级

文档编号：TYSSTHJJ-CSBG-V1.1

项目编号：DPtech-TYSSTHJJ-2024

**太原市生态环境局**

**重点污染源自动监控与基础数据库系统**

**渗透测试报告**



**杭州迪普科技股份有限公司**

**Hangzhou DPtech Technologies Co., Ltd**

**版权所有 侵权必究 All rights reserved**

**修订记录**

| **日期** | **修订版本** | **修改描述** | **作者** |
| --- | --- | --- | --- |
| 2024/12/20 | V1.0 | 初稿编写 | 刘亮亮 |
| 2024/12/20 | V1.1 | 文档审核 | 杨中林 |

**目 录**

[一、 任务概述 1](#_Toc185604693)

[二、 测试范围 1](#_Toc185604694)

[三、 人员安排 1](#_Toc185604695)

[四、 测试结果汇总 1](#_Toc185604696)

[五、 漏洞详细描述及加固建议 2](#_Toc185604697)

[5.1 重点污染源自动监控与基础数据库系统 2](#_Toc185604698)

[5.1.1 存储型xss漏洞（高危） 2](#_Toc185604699)

[5.1.2敏感信息泄露漏洞（高危） 4](#_Toc185604700)

尊敬的太原市生态环境局各位领导：

受贵单位安全测试委托和授权，我公司组织专业渗透测试团队针对贵单位B/S架构网站，进行渗透测试，模拟黑客利用各种主流的攻击技术进行非破坏性的攻击测试，已发现系统当前存在的安全漏洞和风险点，并提出系统安全加固意见。

# 任务概述

本次安全测试根据太原市生态环境局信息系统测试工作需要，利用各种主流的攻击技术对目标系统做模拟攻击测试，以发现系统中的安全漏洞和风险点，并提出系统安全加固意见。

本次渗透所用工具为Wireshark、OD、IDA、WindDB、Appscan、dirsearch、AWVS、迪普漏扫设备Scanner 1000、Kali中自带渗透工具sqlmap、nc、curl、burp suite和迪普安全团队编写的脚本等。

# 测试范围

|  |  |
| --- | --- |
| **应用系统名称** | **访问域名/地址** |
| 重点污染源自动监控与基础数据库系统 | http://10.126.35.42:8080/jointframe/login |

# 人员安排

管理人员：杨中林

实施人员：迪普科技安全服务团队

# 测试结果汇总

**业务系统安全漏洞统计表**

|  |  |  |  |
| --- | --- | --- | --- |
| **漏洞名称** | **漏洞个数** | **漏洞等级** | **漏洞修复情况** |
| 存储型xss漏洞 | 1 | 高危 | 待修复 |
| 敏感信息泄露漏洞 | 1 | 高危 | 待修复 |

# 漏洞详细描述及加固建议

## 5.1 重点污染源自动监控与基础数据库系统

### 5.1.1 存储型xss漏洞（高危）

|  |  |  |  |
| --- | --- | --- | --- |
| **风险名称** | 存储型xss漏洞 | **风险级别** | 高危 |
| **风险描述** | 存储型跨站脚本攻击最为常见的场景是将跨站脚本写入文本输入域中，如留言板、博客或新闻发布系统的评论框。当用户浏览留言和评论时，浏览器执行跨站脚本代码。 | | |
| **测试过程** | 【漏洞URL地址】：  <http://10.126.35.42:8080/jointframe/equManSys/base/ctc__%2Fjointos%2Fapp%2Fbaseweb%2Ffieldequip!equipBrand.page%3FmenuId%3D4028864d7aa43432017aa43b2ff50000>  【漏洞证明】：  登录系统，访问设备管控系统/基础数据管理/设备品牌管理    新增设备并抓包    将equipment修改 111“onmouseover=print()    鼠标掠过产品行，触发js    另外，存在两处由于页面没有删除功能点，测试时候无法删除的数据：  1、设备管控系统/基础设备管理/设备品牌管理    2、门禁系统/成员管理 | | |
| **整改建议** | 1.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。  2.过滤输入的数据，对例如：“ ‘ ”，“ “ ”，” < “，” > “，” on\* “，script、iframe等危险字符进行严格的检查。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。  3.不仅验证数据的类型，还要验证其格式、长度、范围和内容。  4.不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。  5.对输出到页面的数据进行相应的编码转换，如HTML实体编码、JS编码等。  6.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检查。 | | |

### 5.1.2敏感信息泄露漏洞（高危）

|  |  |  |  |
| --- | --- | --- | --- |
| **风险名称** | 敏感信息泄露漏洞 | **风险级别** | 中危 |
| **风险描述** | 攻击者可能通过泄露的敏感信息进行进一步攻击 | | |
| **测试过程** | 【漏洞URL地址】：  <http://10.126.35.42:8080/jointos/app/automonitorcfg/parmAction!getList.page?_=1734675935924>  【漏洞证明】：  登录系统，选择系统管理/参数管理    泄露了短信接口，mongodb账号密码，ftp账号密码等信息  通过短信接口发送短信    返回内容如上    成功调用接口发送短信验证码。 | | |
| **整改建议** | 不将配置信息返回到前端。 | | |