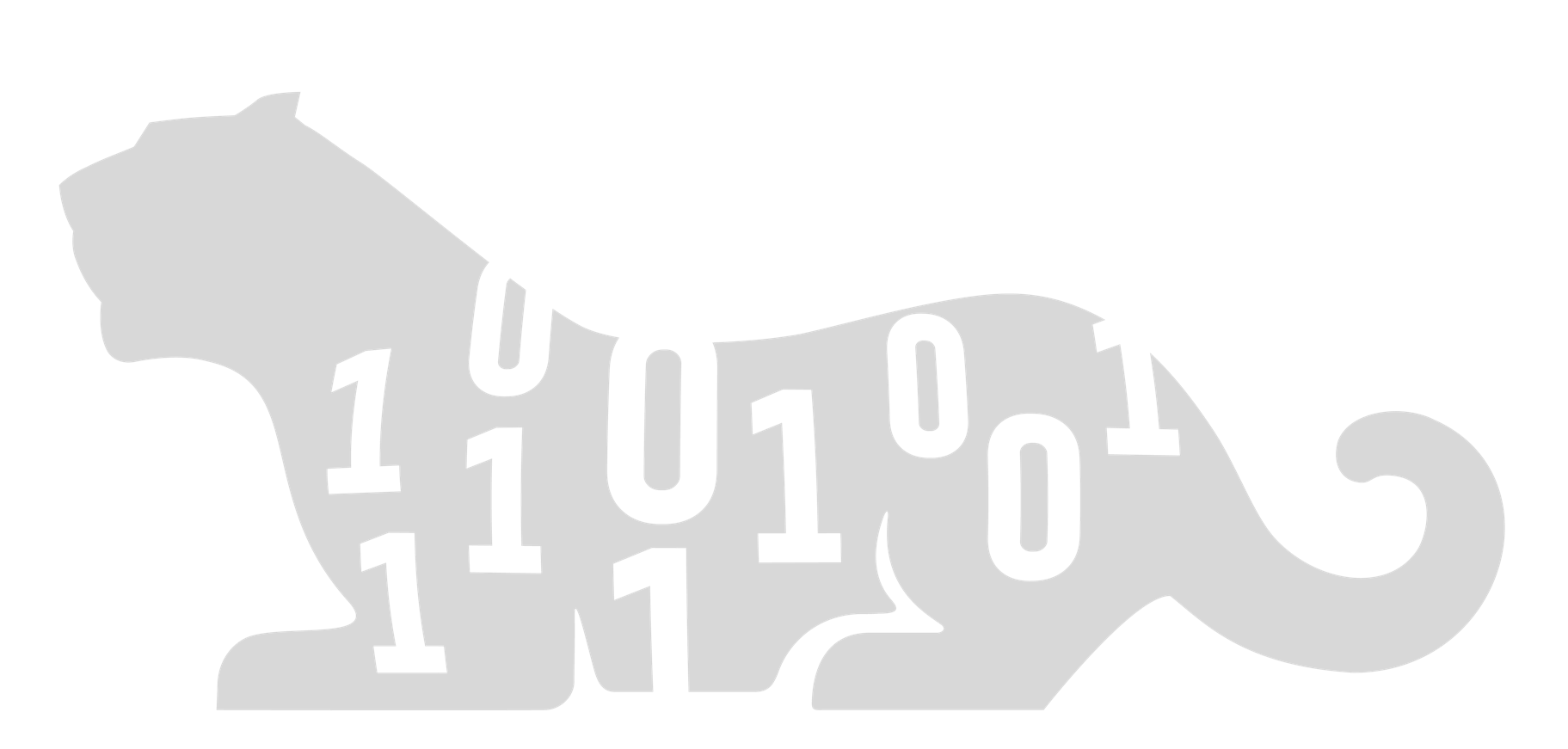


广州中医药大学深圳医院（福田）

体检系统

渗透测试报告



奇安信网神信息技术（北京）股份有限公司

2025年3月17日

版权说明

本文中出现的任何文字叙述、文档格式、插图、图片、方法、过程等内容，除另有特别注明，版权均为奇安信集团（指包括但不限于奇安信科技集团股份有限公司、网神信息技术（北京）股份有限公司、北京网康科技有限公司）所有，受到有关产权及版权法保护。任何个人、机构未经奇安信集团的书面授权许可，不得以任何方式复制或引用本文的任何片段。

本文件包含了来自奇安信集团的可靠、权威的信息，以及被检测单位信息系统的敏感信息，接受这份文件表示同意未经奇安信集团书面请求和书面认可，不得复制，泄露或散布这份文件。如果你不是有意接受者，请注意对这份文件内容的任何形式的泄露、复制或散布都是被禁止的。

|  |
| --- |
| 目录 |
| [一. 摘要 1](#_Toc31541)  [二. 项目概述 6](#_Toc7450)  [2.1 测试目的 6](#_Toc28193)  [2.2 测试范围 6](#_Toc14038)  [2.3 测试依据 6](#_Toc18823)  [2.4 测试工具 7](#_Toc10865)  [三. 问题归纳 8](#_Toc24173)  [3.1 拒绝服务攻击（低危） 8](#_Toc6460)  [3.2 敏感信息泄露-中危 9](#_Toc5050)  [四. 安全风险总结 11](#_Toc22002)  [4.1 安全概况 11](#_Toc8930)  [4.2 安全管理方面 11](#_Toc18232)  [4.3 安全技术方面 11](#_Toc9974)  [4.4 开发过程管理 12](#_Toc5872)  [4.5 安全架构设计基本原则 12](#_Toc30997)  [附录A 安全风险状况等级说明 15](#_Toc23899)  [附录B 漏洞等级状况说明 16](#_Toc28057) |

1. 摘要

经广中医渗透测试授权，奇安信网神信息技术（北京）股份有限公司于2025-01-16至2025-01-17对体检系统进行了安全渗透测试。

经过测试发现体检系统共存在安全漏洞2个，其中高危0个，中危1个，低危1个。

渗透测试结果如下：

**高危问题**：**0**个

**中危问题：1**个

**低危问题**：**1**个

漏洞类型测试结果如下：

| 测试分类 | 测试项 | 测试结果 |
| --- | --- | --- |
| Web安全 | SQL注入 | **通过** |
| 跨站脚本攻击（XSS） | **通过** |
| XML 外部实体（XXE）注入 | **通过** |
| 跨站点伪造请求（CSRF） | **通过** |
| 服务器端请求伪造（SSRF） | **通过** |
| 任意文件上传 | **通过** |
| 任意文件下载或读取 | **通过** |
| 任意目录遍历 | **通过** |
| .svn/.git源代码泄露 | **通过** |
| 信息泄露 | **不通过** |
| CRLF注入 | **通过** |
| 命令执行注入 | **通过** |
| URL重定向 | **通过** |
| Json劫持 | **通过** |
| 第三方组件安全 | **通过** |
| 本地/远程文件包含 | **通过** |
| 任意代码执行 | **通过** |
| Struts2远程命令执行 | **通过** |
| Spring远程命令执行 | **通过** |
| 缺少“X-XSS-Protection”头 | **通过** |
| flash跨域 | **通过** |
| HTML表单无CSRF保护 | **通过** |
| HTTP明文传输 | **通过** |
| 使用GET方式进行用户名密码传输 | **通过** |
| X-Frame-Options Header 未配置 | **通过** |
| 任意文件删除 | **通过** |
| 绝对路径泄露 | **通过** |
| 未设置HTTPONLY | **通过** |
| X-Forwarded-For伪造 | **通过** |
| 明文传输 | **通过** |
| 不安全的HTTP Methods | **通过** |
| 任意文件探测 | **通过** |
| 网络传输安全 | 加密方式不安全 | **通过** |
| 使用不安全的telnet协议 | **通过** |
| 业务逻辑安全 | 验证码缺陷 | **通过** |
| 反序列化命令执行 | **通过** |
| 用户名枚举 | **通过** |
| 用户密码枚举 | **通过** |
| 用户弱口令 | **通过** |
| 会话标志固定攻击 | **通过** |
| 平行越权访问 | **通过** |
| 垂直越权访问 | **通过** |
| 未授权访问 | **通过** |
| 业务逻辑漏洞 | **通过** |
| 短信炸弹 | **通过** |
| Flash未混淆导致反编译 | **通过** |
| 中间件安全 | 中间件配置缺陷 | **通过** |
| 中间件弱口令 | **通过** |
| Jboss反序列化命令执行 | **通过** |
| Websphere 反序列化命令执行 | **通过** |
| Jenkins反序列命令执行 | **通过** |
| JBoss远程代码执行 | **通过** |
| Webloigc反序列化命令执行 | **通过** |
| Apache Tomcat样例目录session操纵 | **通过** |
| 服务器安全 | 文件解析代码执行 | **通过** |
| 域传送漏洞 | **通过** |
| Redis未授权访问 | **通过** |
| MongoDB未授权访问 | **通过** |
| 操作系统弱口令 | **通过** |
| 数据库弱口令 | **通过** |
| 本地权限提升 | **通过** |
| 已存在的脚本木马 | **通过** |
| 永恒之蓝 | **通过** |
| mssql信息探测 | **通过** |
| windows操作系统漏洞 | **通过** |
| 数据库远程连接 | **通过** |
| 权限分配不合理 | **通过** |
| 漏扫专用漏洞库-高危 | HTTP.sys远程代码执行漏洞 | **通过** |
| 存储型跨站脚本攻击（XSS） | **通过** |
| SNMP使用默认团体字符串 | **通过** |
| 任意用户密码修改/重置 | **通过** |
| 服务器端请求伪造(SSRF) | **通过** |
| XML注入 | **通过** |
| 任意文件下载 | **通过** |
| 文件包含 | **通过** |
| SQL注入漏洞 | **通过** |
| Apache Struts2远程命令执行(S2-045) | **通过** |
| Drupal版本过低导致多个漏洞 | **通过** |
| PHP版本过低导致多个漏洞 | **通过** |
| 弱口令 | **通过** |
| 垂直越权 | **通过** |
| Apache Tomcat文件包含(CNVD-2020-10487/CVE-2020-1938) | **通过** |
| Tomcat版本过低 | **通过** |
| Apache Shiro RememberMe 1.2.4 反序列化过程命令执行漏洞 | **通过** |
| Fastjson远程代码执行漏洞（CNVD-2019-22238） | **通过** |
| OpenSSL版本过低导致多个漏洞 | **通过** |
| 漏扫专用漏洞库-中危 | HOST头攻击 | **通过** |
| Flash跨域漏洞 | **通过** |
| IIS短文件名泄露 | **通过** |
| Apache Tomcat示例目录漏洞 | **通过** |
| Apache Tomcat examples 目录可访问导致多个漏洞 | **通过** |
| 框架注入漏洞 | **通过** |
| 目录遍历 | **通过** |
| rsync未授权访问 | **通过** |
| FTP匿名登录 | **通过** |
| 水平越权 | **通过** |
| 验证码绕过 | **通过** |
| phpinfo页面泄露 | **通过** |
| .Git源码泄露 | **通过** |
| SSL 3.0 POODLE攻击信息泄露漏洞(CVE-2014-3566) | **通过** |
| 短信轰炸 | **通过** |
| 链接注入漏洞 | **通过** |
| 验证码失效 | **通过** |
| 漏洞专用漏洞库-低危 | 管理后台泄露 | **通过** |
| 备份文件 | **通过** |
| 版本泄露 | **通过** |
| 内网IP泄露 | **通过** |
| jQuery库版本较低 | **通过** |
| 密码暴力破解风险 | **通过** |
| WEB-INF/web.xml泄露 | **通过** |
| Bazaar存储库泄露 | **通过** |
| Snoop Servlet信息泄露 | **通过** |
| . DS\_Store文件泄漏 | **通过** |
| . svn源代码泄露 | **通过** |
| 点击劫持：缺少X-Frame-Options标头 | **通过** |
| 没有CSRF保护的HTML表单 | **通过** |
| 目录列表 | **通过** |
| . idea目录信息泄露 | **通过** |
| 用户凭据明文传输 | **通过** |
| ASP.NET调试已启用 | **通过** |
| SSL/TLS存在Bar Mitzvah Attack漏洞（RC4密码套件） | **通过** |
| SSL/TLS存在FREAK攻击漏洞（CVE-2015-0204） | **通过** |
| 允许HTTP OPTIONS方法 | **通过** |
| 不安全的HTTP方法 | **通过** |
| 物理路径泄露 | **通过** |

整体而言，系统在本次渗透测试实施期间的安全风险状况为“预警状态”。(系统安全风险状况等级的含义及说明详见附录A)。

1. 项目概述
   1. 测试目的

通过本项目的成功实施，在坚持科学、客观、公正原则的基础上，全面、完整地了解社康中心智能服务系统当前的安全状况，分析系统所面临的各种风险，模拟攻击者可能利用的漏洞，根据测试结果发现系统存在的安全问题，并对严重的问题提出加固的建议。

本次测试预期达到的目标为：

* 发现授权渗透测试目标系统的安全漏洞
* 针对发现的漏洞提供加固方案及防护建议
  1. 测试范围

本次渗透测试的范围如下：

|  |  |  |
| --- | --- | --- |
| 序 号 | 名 称 | 备 注 |
| 1 | 体检系统 | <https://fgh.szftzy.com:8001/wechat/#/index> |

* 1. 测试依据

安全测试服务将参考下列规范进行工作。

* 信息安全技术 信息安全风险评估规范（GB/T 20984-2007）
* 信息技术 信息安全管理实用规则（GB/T 19716-2005)(ISO/IEC 17799:2000)
* 信息技术 安全技术 信息安全管理实用规则（GB/T 22081-2016 ）
* GB T 31509-2015 信息安全技术 信息安全风险评估实施指南
* 信息系统审计标准（ISACA）
* OWASP OWASP\_Testing\_Guide\_v3
* OWASP OWASP\_Development\_Guide\_2005
* OWASP OWASP\_Top\_10\_2010\_Chinese\_V1.0
* 奇安信渗透测试最佳实践
* GB∕T 30279-2020 信息安全技术 网络安全漏洞分类分级指南
* GB∕T 37931-2019 信息安全技术 Web应用安全检测系统安全技术要求和测试评价方法
* JR∕T 0213-2021 金融网络安全 Web 应用服务安全测试通用规范
* JRT 0092-2019中国金融移动支付 移动金融客户端应用软件安全管理规范
* 金融Web应用渗透测试技术规范
  1. 测试工具
* 拓扑分析工具：DNS Sweep、Nslookup 等
* 自动化扫描工具：Nessus、AIScanner 等
* 端口扫描、服务检测：Nmap、SuperScan 等
* 嗅探分析工具：Ethereal、Entercap、Dsniff 等
* Exploiting 利用工具：Metasploit Framework 等
* 应用缺陷分析工具：Gatling自动化渗透测试系统、SQLMAP 等
* 静态分析工具：MobSF等
* 动态分析工具：天问供应链分析平台等

1. 问题归纳
   1. 拒绝服务攻击（低危）

**目标名称**

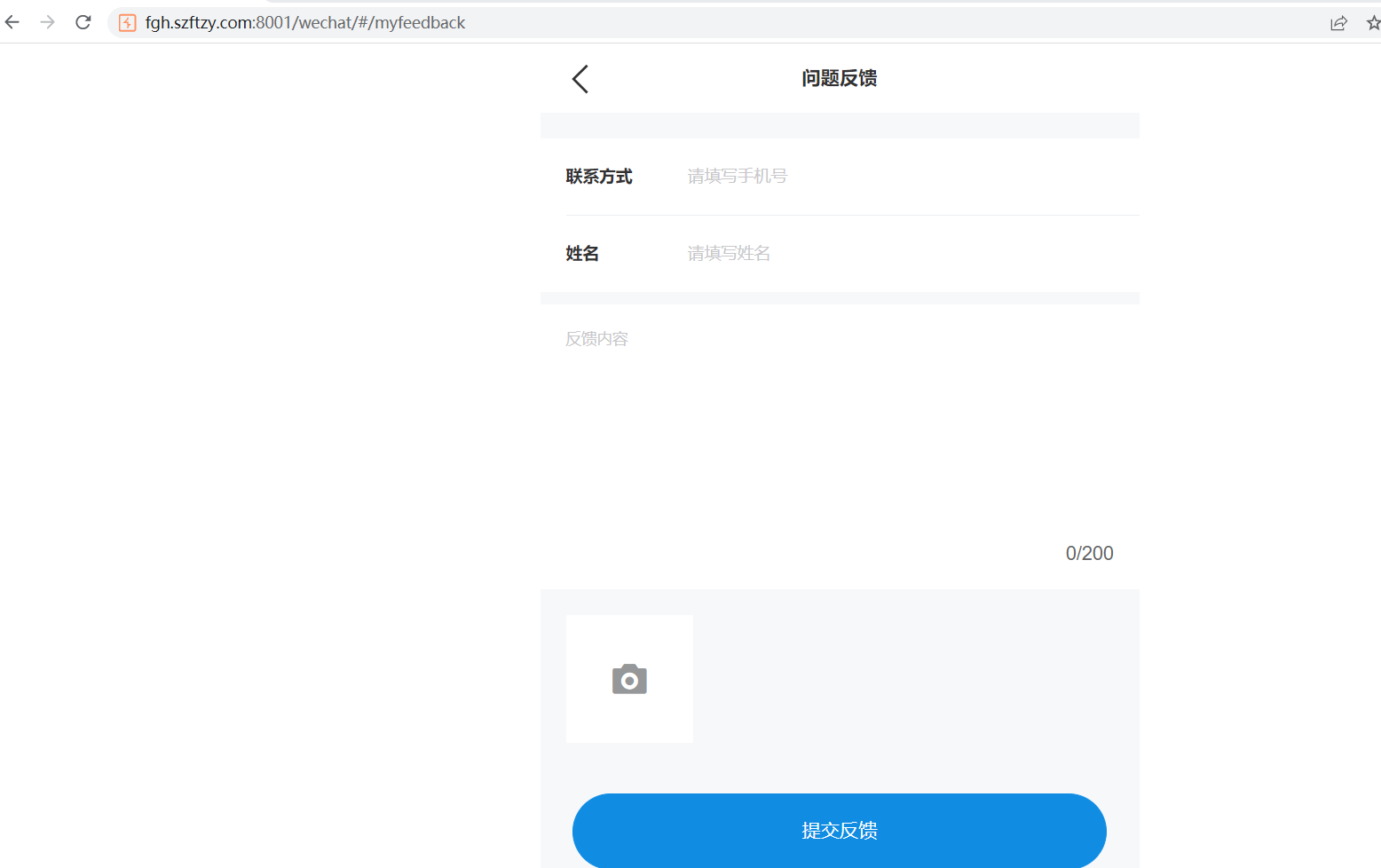
体检系统

**漏洞URL**

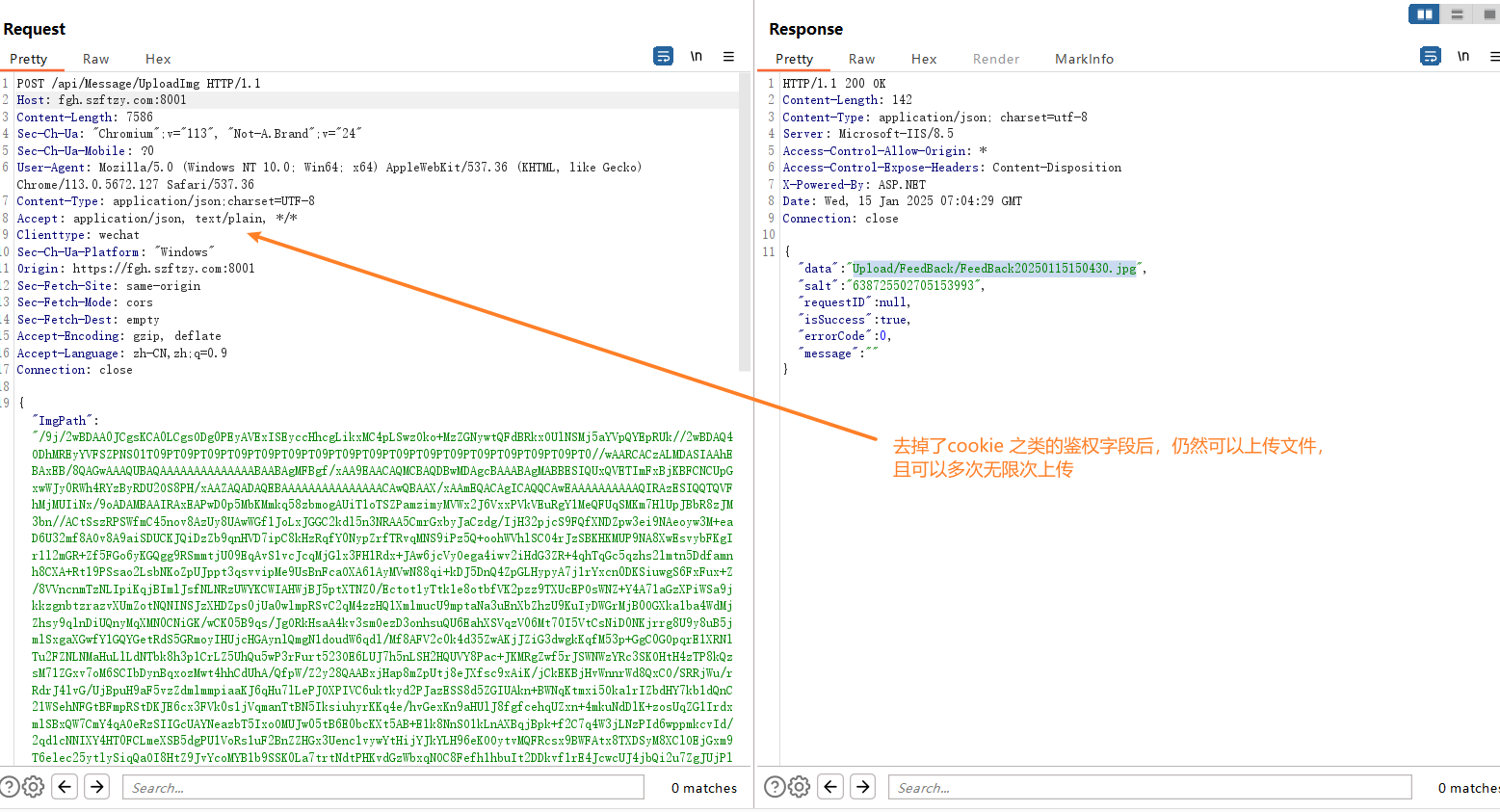
<https://fgh.szftzy.com:8001/wechat/#/myfeedback>

**漏洞过程**

功能点，问题反馈中的文件上传



去掉了 Cookie 与Token 鉴权字段，仍然可以无条件无限上传文件，消耗服务器的存储资源



**漏洞描述**

拒绝服务攻击漏洞的危害在于能够通过耗尽目标系统的计算资源、带宽或其他关键资源，导致服务无法正常响应用户请求，从而使业务中断或完全瘫痪。这种攻击会严重影响组织的正常运营，造成经济损失、客户流失以及品牌信誉的损害，特别是在关键服务或高并发应用场景下更为显著。此外，攻击者还可能利用拒绝服务攻击作为掩护，隐藏其他恶意活动，例如数据窃取或系统入侵，进一步加剧安全风险。

**修复建议**

建议对问题反馈中文件上传功能进行鉴权处理，或者限制每日可上传的次数，不允许攻击者可以无条件无限上传文件

**复测结果**

* 1. 敏感信息泄露-中危

**目标名称**

体检系统

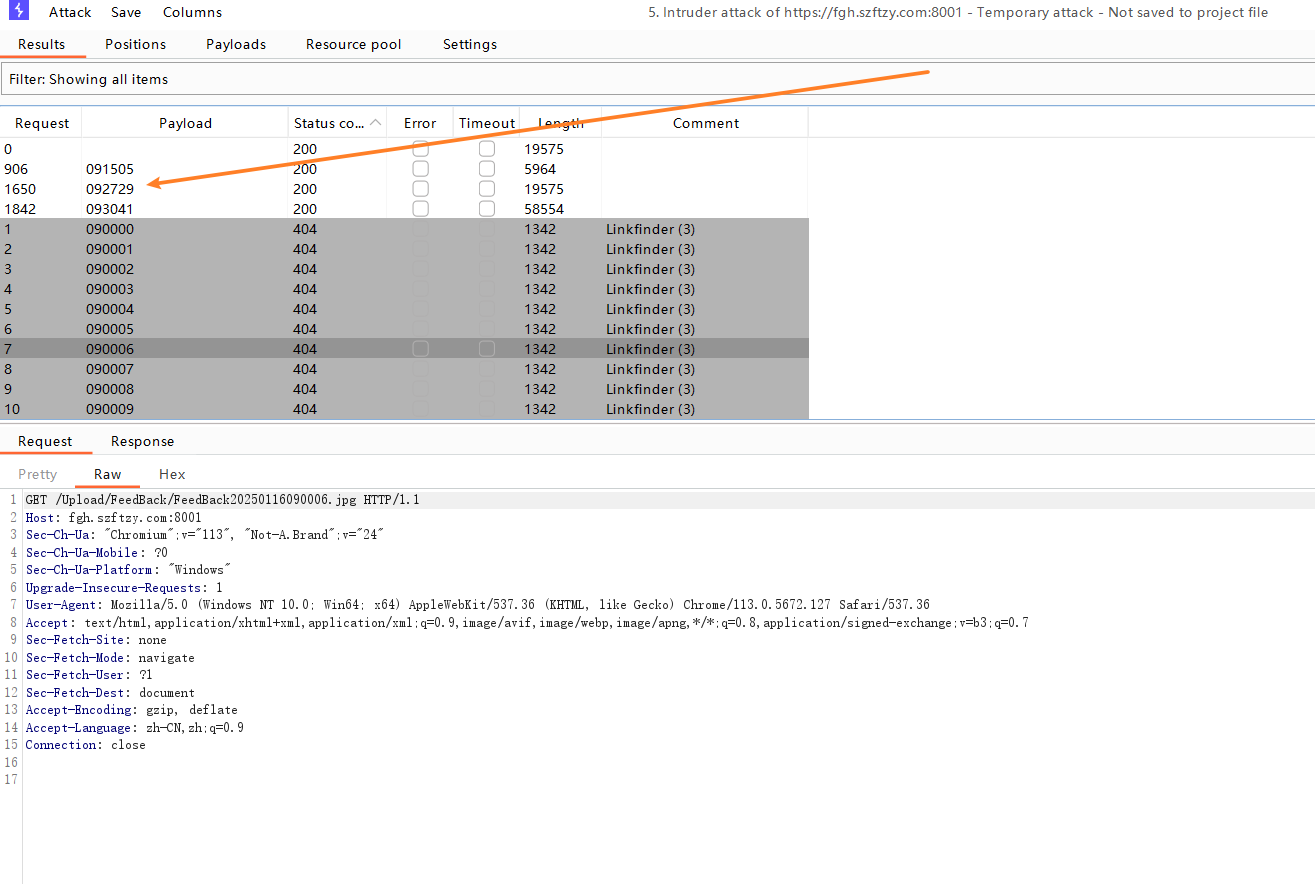
**漏洞URL**

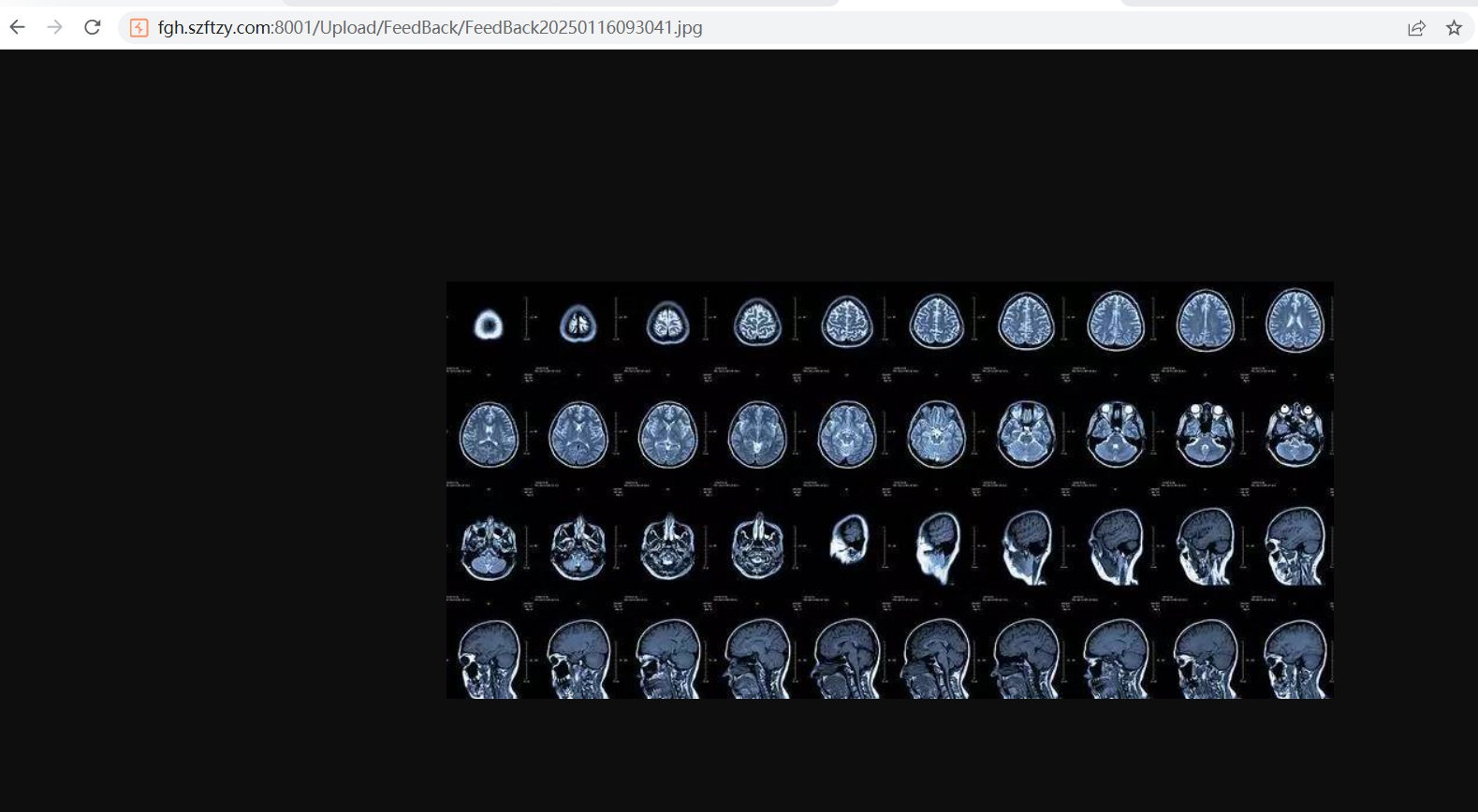
<https://fgh.szftzy.com:8001/Upload/FeedBack/FeedBack20250115145658.jpg>

**漏洞过程**

依然是上面文件上传的功能点，上传文件后返回了文件的路径，如/Upload/FeedBack/FeedBack20250115145658.jpg，根据文件名的命名规则，攻击者可以自定义字典，遍历到**其他用户**提交的问题反馈的图片

如





**漏洞描述**

敏感信息泄露漏洞的危害在于可能导致组织或个人的机密信息被未授权的攻击者获取，包括用户数据、业务机密、系统配置、访问凭证等，这不仅会直接引发经济损失，还可能导致用户隐私被侵犯、竞争优势丧失，以及严重的法律与合规风险。此外，泄露的信息可能被攻击者利用进行进一步的攻击，如社会工程、账户劫持或供应链攻击，从而放大安全威胁的范围和深度，对组织的声誉和信任度造成长期的负面影响。

**修复建议**

建议随机化（无规律）命名用户上传的图片名称，让攻击者不可手动构造文件的名称

**复测结果**

1. 安全风险总结
   1. 安全概况

体检中心存在中危漏洞1个，低危漏洞1个。

* 1. 安全管理方面
* 进一步落实口令管理制度，建议使用独立的、随机生成的满足强度要求的口令，严禁使用弱口令、统一口令管理服务器，及时销毁临时、测试账户；
* 进一步加强人员安全意识教育，预防攻击事件发生。
* 信息系统上线前增加安全验收环节，保证系统安全上线；
* 信息系统下线后增加安全退出环境，保证相关信息、防火墙规则、数据的安全处理。
  1. 安全技术方面
* 运行环境部署需要进行加固处理，Web应用访问数据库的用户需要采用最小权限原则，防止攻击者利用数据库缺陷进行权限提升等操作.
* 关注各个应用系统所使用程序、组件、第三方插件等安全现状，及时更新相应的补丁版本；
* 加强对敏感服务器、配置文件、目录的访问控制，以免敏感配置信息泄露；
* 加强信息系统安全配置检查工作。
  1. 开发过程管理
* 在系统开发与运行维护的所有阶段（如：计划需求、设计、编码、测试、运行和维护）强制实施严格的变更控制，对变更的申请、审核、测试、批准、执行计划与具体实施提出明确要求，确保系统安全性与控制措施不被损害；

变更控制包括以下内容：

* + 审查变更控制措施和流程的完整性，确保未被修改和破坏；
  + 确保操作系统的更改不会对应用系统的安全性和完整性造成不良影响；
  + 确保系统文档在每次修改后得到及时更新，并确保旧文档被正确归档和处置；
  + 做好软件升级的版本控制，如保存历史版本；
  + 保留所有变更的审计跟踪记录；
  + 确保操作文档以及用户程序能在必要时被修改；
  + 确保及时更新业务连续性计划。
* 针对开发人员对源代码的管理，应严禁将源代码信息上传至互联网上的网络磁盘或源代码仓库，如GITHUB、网盘等；建议建立自运营的代码管理仓库，如有必要上传至互联网上，应对敏感关键信息进行屏蔽，如API地址、IP地址、数据库密码等，防止信息泄露；
* 应尽量避免修改厂商提供的软件包，如必须修改，应注意以下几点：
  + 评估软件包内置的控制措施和完整性流程遭受破坏的风险；
  + 应征得原厂商的同意，由原厂商提供标准的升级程序来实现软件包的更改。
* 在软件的原始采购、开发、使用和维护过程中，应采取如下防范控制措施：
  + 仅从信誉卓著的厂商处购买软件；
  + 尽量购买提供源代码的软件，以便进行检验，在投入使用之前检查所有源代码；
  + 使用通过权威机构评估测试的软件产品；
  + 一旦安装完毕，控制对源代码的访问和修改；
  + 安装并正确使用有关后门、特洛伊代码的检测和查杀工具。
  1. 安全架构设计基本原则

在应用系统软件开发设计的过程中，对应用系统的总体设计应当满足如下安全原则：

|  |  |
| --- | --- |
| 原则 | 说明 |
| 最小权限原则  Least Privilege | 应用软件的每个模块如进程、用户只能访问当下所必需的信息或者资源。赋予每一个合法动作最小的权限，以保护数据以及功能避免受到错误或者恶意行为的破坏。 |
| 权限分离原则  Separation of Duties | 对业务的操作、管理和审计权限应该由软件中的不同角色的用户分别承担；普通用户和管理员用户信息应该存放在不同的数据表中。 |
| 深度防御原则  Defense in Depth | 在应用程序对业务数据进行处理的每个阶段都要考虑安全性问题，不能仅在某个阶段做安全防御，这样单点防御一旦被突破将造成安全风险。 |
| 容错保护原则  Fail Secure | 当程序出现故障时或系统异常当系统失败时，可以进入到一个失败保护的状态。如果用户请求失败，系统仍可保障安全。 |
| 单点异常终止原则  Single Point of Failure | 当用户提交数据超出预期时，应立即终止程序的执行，不要试图加以修正并继续执行下去。 |
| 外来代码安全原则  Least Third Party Components | 严格控制第三方函数与插件的使用，对外来代码必须进行详细的安全测试。 |
| 代码重用原则  Leveraging Existing Components | 尽可能的重用软件已有的模块，这样可以降低引入新的漏洞和攻击界面的可能性。 |
| 数据保护原则  Data Protection | 对用户数据的保护功能应涵盖用户数据存储的完整性、用户数据传输保密性、数据传输的访问控制、剩余信息的保护、数据反转操作等内容；应对系统中关键数据（如用户密码等）的存储和网络传输时应采用加密保护，实用加密加密算法应该符合国际标准、国家标准和业界标准。 |
| 可审计原则  Auditing | 在应用系统中设计审计日志记录的功能，并对应用系统产生的日志增加完备的审计功能。 |
| 开放设计原则  Open Design | 开放设计与“不开放即安全”的原则相对而言，认为设计本身不应具有神秘感。这一原则的具体表现可以参见应用于加密设计的Kerchoff定律，“系统不应单纯依赖私密性，若落入敌人手中则毫无优势可言”；开放设计以提高系统兼容性和可扩展性。 |
| 抗抵赖原则  Anti Repudiation | 对于涉及支付交易等重要的业务场景，系统设计应有效地防止通信双方抵赖，如采用电子证书签名等方式。 |
| 规范性  Standardization | 系统设计所采用的安全技术和安全产品应符合国际标准、国家标准和业界标准，为系统的扩展升级、与其他系统的互联提供良好的基础。 |
| 可扩展性  Scalability | 以当前业务安全需求为基础，充分考虑发展的需要，安全功能模块子系统以插件或接口方式以方便未来的扩展。 |
| 实用性  Practicable | 安全功能设计需要尽可能的考虑投入产出比，同时尽量控制对用户体验的影响。 |
| 符合性  Regulatory Compliance | 安全功能的设计尽可能的要符合国家规范、行业规范以及业界的通用标准，如等级保护等规范。 |

1. 安全风险状况等级说明

| 安全风险状况说明 | |
| --- | --- |
| 良好状态 | 信息系统处于良好运行状态，没有发现或只存在零星的低风险安全问题，此时只要保持现有安全策略就满足了本系统的安全等级要求。 |
| 预警状态 | 信息系统中存在一些漏洞或安全隐患，此时需根据评估中发现的网络、主机、应用和管理等方面的问题对进行有针对性的加固或改进。 |
| 严重状态 | 信息系统中发现存在严重漏洞或可能严重威胁到系统正常运行的安全问题，此时需要立刻采取措施，例如安装补丁或重新部署安全系统进行防护等等。 |
| 紧急状态 | 信息系统面临严峻的网络安全态势，对组织的重大经济利益或政治利益可能造成严重损害。此时需要与其他安全部门通力协作采取紧急防御措施。 |

1. 漏洞等级状况说明

| 安全风险状况说明 | |
| --- | --- |
| 低危漏洞 | 对系统造成较小的影响，攻击成本高，攻击场景较为苛刻，不会直接影响到系统的正常运行，攻击者可能无法通过该漏洞获得权限。 |
| 中危漏洞 | 对系统造成一般的影响，攻击成本一般，在特定场景下将可影响系统的正常运行，攻击者需要配合其他安全漏洞方可间接获得权限。 |
| 高危漏洞 | 对系统造成严重的影响，攻击成本低，一般情况下可直接利用而无需特定场景和要求，攻击者可直接利用该漏洞获得权限。 |