某某云平台管理系统

网络安全评估-渗透测试报告

PEN-DOC-202409300643

PeTeReport 中文版

2024 年 09 月 30 日

Contents

1 项目概况 **3**
1.1 描述 . . . 3

2 执行摘要 **3**
2.1 漏洞汇总总结 . . . 3
2.1.1 按严重程度细分 . . . 3
2.1.2 按 CWE 类别细分 . . . 4
2.1.3 按 OWASP 类别细分 . . . 4
2.2 漏洞列表 . . . 5
2.3 渗透测试范围 . . . 5
2.3.1 测试范围 . . . 5
2.3.2 非测试范围 . . . 6
2.4 测试方法 . . . 6
2.5 修复建议 . . . 6

3 漏洞结果和风险分析 **8**
3.1 SQL 注入 . . . 8
3.2 多个账号存在弱密码 . . . 10
3.3 未加密的登录请求 . . . 13
3.4 账号枚举 . . . 15

1 项目概况

1.1 描述

某某云平台管理系统是一个全面的解决方案，旨在为企业提供一个高效、安全的环境来管理其云资源。该系统的主要功能包括用户权限管理、组织架构管理、菜单配置和日志管理，以确保企业能够有效地控制对云资源的访问，并保持操作的透明度和可追溯性。

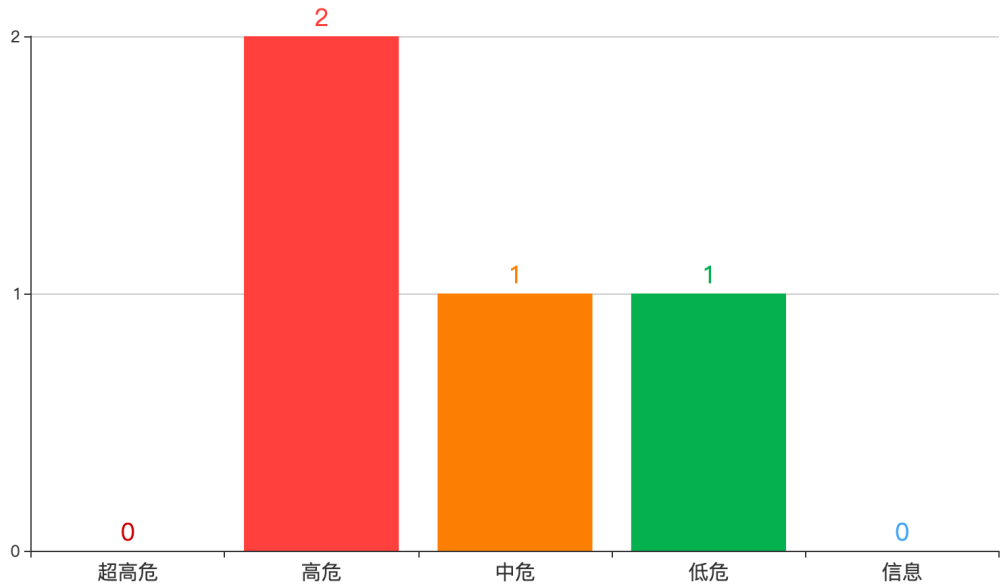
2 执行摘要

某某云平台管理系统是一个全面的解决方案，旨在为企业提供一个高效、安全的环境来管理其云资源。该系统的主要功能包括用户权限管理、组织架构管理、菜单配置和日志管理，以确保企业能够有效地控制对云资源的访问，并保持操作的透明度和可追溯性。

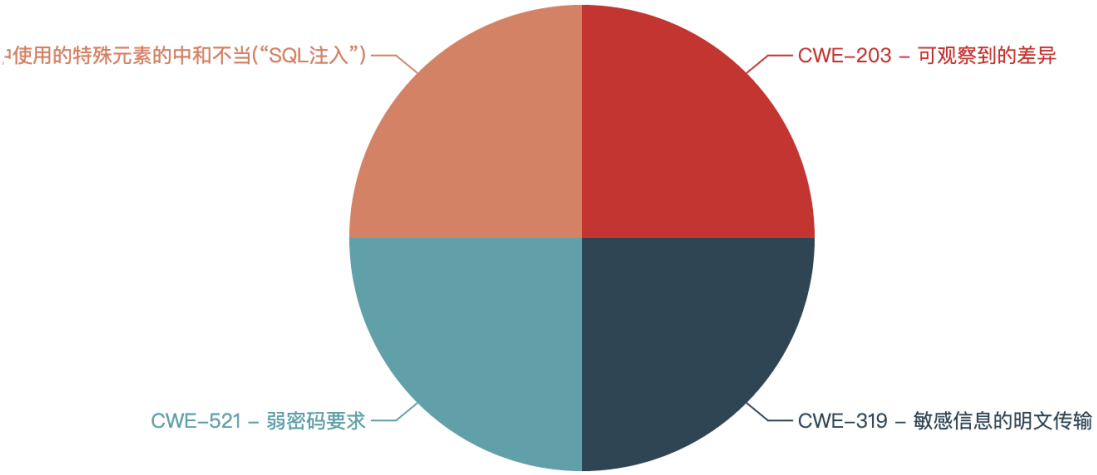
本次渗透测试旨在评估某某云平台管理系统的安全性，识别潜在的安全漏洞和弱点，并验证系统对各种攻击的防御能力。

2.1 漏洞汇总总结

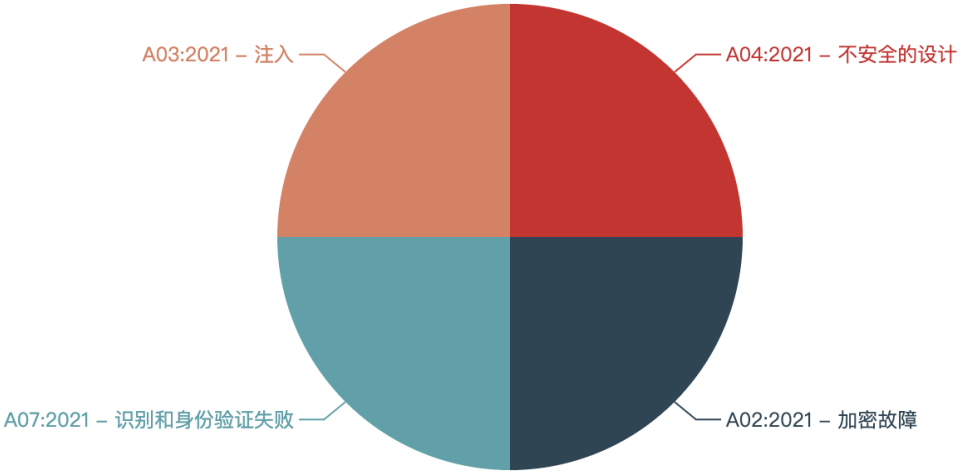
2.1.1 按严重程度细分



2.1.2 按 CWE 类别细分



2.1.3 按 OWASP 类别细分



2.2 漏洞列表

1、【高危】 SQL 注入

2、【高危】多个账号存在弱密码

3、【中危】未加密的登录请求

4、【低危】账号枚举

2.3 渗透测试范围

2.3.1 测试范围

本次渗透测试的范围主要集中在某某云平台管理系统的四个关键组件上：用户权限管理、组织架构管理、菜单配置和日志管理。每个组件的测试重点和目标如下：

- 用户权限管理：
  - 测试目标：验证用户权限分配和访问控制机制的有效性。
  - 测试内容：检查是否存在权限提升漏洞，即低权限用户是否能够访问或修改只有高权限用户才能访问的资源。同时，测试用户角色和权限的分配是否正确执行，以及是否能够防止未授权访问。
- 组织架构管理：
  - 测试目标：评估组织架构配置的安全性。
  - 测试内容：检查是否存在安全漏洞，如 SQL 注入，可能导致攻击者操纵或访问组织架构数据。同时，测试组织架构的修改是否被适当地记录和监控。
- 菜单配置：
  - 测试目标：确保菜单配置的安全性和稳定性。
  - 测试内容：检查菜单配置是否存在任何安全漏洞，如跨站脚本攻击（XSS），以及验证菜单配置是否可以被未授权用户修改。
- 日志管理：
  - 测试目标：验证日志记录功能的完整性和安全性。
  - 测试内容：检查日志记录是否全面，是否能够记录所有关键用户操作和安全相关事件。同时，评估日志管理功能是否能够防止未授权访问和篡改。

2.3.2 非测试范围

- 第三方服务和集成：不测试与第三方服务的集成，如外部身份验证系统、支付网关或其他第三方 API。
- 物理安全措施：不包括对物理安全措施的测试，如服务器机房的访问控制或监控系统的评估。
- 网络基础设施：不涵盖对网络基础设施的渗透测试，如防火墙、路由器和交换机。

2.4 测试方法

在某某云平台管理系统的渗透测试中，将采用一系列的测试方法来全面评估系统的安全性。这些方法包括：

- 信息收集：收集关于目标系统的信息，包括域名、IP 地址、开放的端口和服务等。这可以通过使用工具如 Nmap、Whois 和 DNSenum 来完成。
- 漏洞扫描：使用自动化工具（如 Nessus、OpenVAS）扫描目标系统，以识别已知的漏洞和安全弱点。
- 配置审查：审查系统配置，包括网络设备、操作系统、数据库和应用服务器配置，以识别不当配置或安全策略的缺失。
- 手动测试：在自动化工具无法充分检测的情况下，进行手动测试。这包括：
  - SQL 注入测试：尝试通过输入恶意 SQL 代码来操纵数据库。
  - 跨站脚本（XSS）测试：注入恶意脚本，以验证系统是否能够防止这些脚本在用户浏览器上执行。
  - 跨站请求伪造（CSRF）测试：验证系统是否能抵御来自恶意网站的请求。
  - 权限提升测试：检查是否可以绕过权限控制，执行只有更高权限用户才能执行的操作。
- 社会工程学测试：通过模拟钓鱼攻击或其他社会工程学手段，测试员工的安全意识和反应。
- 无线网络安全测试：如果适用，对无线网络进行安全测试，包括接入点安全性评估和无线网络渗透测试。
- 物理安全测试：如果适用，评估物理安全措施，如访问控制、监控系统和环境控制。
- 日志分析和监控：分析系统日志，以识别异常行为或潜在的安全事件。
- 报告和反馈：编写详细的渗透测试报告，包括发现的漏洞、风险评估和修复建议。与系统管理员和开发团队合作，确保所有识别的问题都得到妥善解决。

2.5 修复建议

- 用户权限管理：
  - 强化权限验证机制，确保用户只能访问其授权范围内的资源。
  - 实施最小权限原则，只授予用户完成其任务所必需的权限。
  - 定期审查和更新用户权限，确保权限分配的准确性和及时性。
- 组织架构管理：
  - 强化组织架构的访问控制，确保只有授权用户能够创建、修改或删除组织架构信息。
  - 实施强密码策略和多因素认证，以增强账户安全性。

- 菜单配置：
  - 强化菜单配置的访问控制，确保只有授权用户能够修改菜单配置。
  - 实施输入验证和输出编码，以防止跨站脚本攻击（XSS）。
- 日志管理：
  - 增强日志记录功能，确保所有关键操作都被适当地记录和监控。
  - 实施日志分析和监控机制，以便及时发现和响应安全事件。
- SQL 注入漏洞：
  - 实施输入验证和参数化查询，以防止恶意输入。
  - 使用专业的 Web 应用防火墙（WAF）来检测和阻止 SQL 注入攻击。
- 跨站脚本攻击（XSS）：
  - 实施输入验证和输出编码，以防止恶意脚本的注入和执行。
  - 使用内容安全策略（CSP）来限制资源的加载和脚本的执行。
- 跨站请求伪造（CSRF）：
  - 实施 CSRF 令牌机制，以防止恶意网站发起的伪造请求。
  - 使用 SameSite 属性来限制 Cookie 的发送，以减少 CSRF 攻击的风险。
- 物理安全：
  - 强化物理安全措施，如访问控制、监控系统和环境控制。
  - 定期进行物理安全审计和演练，以确保安全措施的有效性。
- 安全意识和培训：
  - 定期为员工提供安全意识和培训，以提高他们对安全威胁的认识和应对能力。
  - 建立安全事件报告和响应机制，以便及时发现和应对安全事件

3 漏洞结果和风险分析

3.1 SQL 注入

严重程度: 高危
CWE: 89 - SQL 命令中使用的特殊元素的中和不当 (“SQL 注入”)
OWASP: 3 - 注入

【描述】

SQL 注入漏洞发生在应用程序将用户输入作为 SQL 查询的一部分时，没有进行适当的验证或转义，允许攻击者控制数据库查询。

【位置】

- https://github.com/feishi-1/petereport-zh
- https://github.com/feishi-1/petereport-zh/tree/main/app

【影响】

攻击者可能能够读取、修改、删除数据库中的数据，甚至在某些情况下，能够执行数据库管理系统中的任意命令。

【证明与验证】

Figure 1: SQL 注入.png

【修复建议】

- 使用预编译的语句（prepared statements）、存储过程或参数化查询。
- 对输入进行严格的验证和转义。
- 实施最小权限原则，限制数据库账户的权限。

3.2 多个账号存在弱密码

严重程度：高危
CWE: 521 - 弱密码要求
OWASP: 7 - 识别和身份验证失败

【描述】

该漏洞源于系统中存在多个使用弱密码的账号，特别是在管理后台。弱密码通常指容易被猜测或通过简单的暴力破解方法获取的密码。这种情况严重威胁系统安全，因为攻击者可以轻易获取系统访问权限，尤其是管理员权限。一旦攻击者成功登录，他们可以进行各种恶意操作，包括数据窃取、系统配置更改、植入恶意代码等。对于管理后台账户，这个问题更为严重，因为它可能导致整个系统被完全控制。

【位置】

- https://github.com/feishi-1/petereport-zh
- https://github.com/feishi-1/petereport-zh/tree/main/app

【影响】

- 未授权访问：攻击者可轻易获取系统访问权限。
- 数据泄露：敏感信息可能被窃取或篡改。
- 系统完整性受损：攻击者可修改系统设置和数据。
- 权限提升：普通用户账户被破解后可能被用来获取更高权限。
- 声誉损害：系统安全事件可能导致严重的声誉和财务损失。
- 业务中断：管理后台被入侵可能导致服务中断。
- 法律风险：可能违反数据保护法规，导致法律责任。

【证明与验证】

Figure 2: 多个账号存在弱密码.png

【修复建议】

- 密码策略强化：
  - 实施强密码策略，要求密码包含大小写字母、数字和特殊字符。
  - 设置最小密码长度（建议至少 8 个字符）。
  - 禁止使用常见的弱密码和易猜测的密码。
- 管理后台安全：
  - 立即修改所有管理后台的弱密码。
  - 实施双因素认证（2FA）或多因素认证（MFA）。
  - 限制管理后台的访问 IP 范围。
- 定期密码更新：
  - 强制用户定期更改密码，尤其是管理员账户。
  - 实施密码历史检查，防止重复使用旧密码。
- 访问控制：
  - 实施最小权限原则，限制用户只能访问必要的功能。
  - 对关键操作实施额外的身份验证。
- 监控和审计：

  - 实施登录尝试监控和异常检测机制。
  - 定期进行账户安全审计。

- 用户教育：

  - 对所有用户进行安全意识培训，强调强密码的重要性。
  - 提供密码管理器使用指导。

3.3 未加密的登录请求

严重程度: 中危
CWE: 319 - 敏感信息的明文传输
OWASP: 2 - 加密故障

【描述】

此漏洞存在于 Web 应用的登录功能中。当用户尝试登录时，敏感信息如用户名和密码以明文形式通过网络传输。由于缺乏加密保护，攻击者可以通过网络嗅探或中间人攻击轻易地截获这些凭证。这种不安全的传输方式严重威胁了用户的账户安全和隐私。

【位置】

- https://github.com/feishi-1/petereport-zh
- https://github.com/feishi-1/petereport-zh/tree/main/app

【影响】

- 用户凭证可能被窃取，导致未授权访问。
- 可能导致个人隐私泄露和身份盗窃。
- 违反数据保护法规，可能面临法律和财务风险。
- 损害公司声誉和用户信任。
- 可能导致更广泛的系统入侵，如果攻击者获得有特权账户的访问权。
- 增加跨站请求伪造（CSRF）攻击的风险。

【证明与验证】

Figure 3: 未加密的登录请求.png

【修复建议】

- 实施 HTTPS 加密：在整个网站范围内强制使用 HTTPS，特别是登录页面和所有涉及敏感数据传输的页面。
- 使用 TLS 1.2 或更高版本，确保使用强加密算法。
- 在客户端对登录数据进行加密：
  - 使用 JavaScript 库（如 CryptoJS）在发送前加密敏感数据。
  - 实施公钥加密，使用服务器提供的公钥加密数据。
- 服务器端加密：确保服务器端也对敏感数据进行加密存储。
- 实施安全的会话管理，使用安全的、HttpOnly 和 Secure 标志的 cookies。
- 考虑使用双因素认证（2FA）或多因素认证（MFA）增加额外的安全层。

3.4 账号枚举

严重程度: 低危
CWE: 203 - 可观察到的差异
OWASP: 4 - 不安全的设计

【描述】

账号枚举漏洞存在于系统登录页面。攻击者可以通过尝试登录时输入系统中存在的用户名配合错误密码，以及不存在的用户名配合错误密码，观察系统返回的不同错误信息来判断某个用户名是否存在于系统中。这种差异化的错误信息使得攻击者能够枚举出系统中存在的有效账号信息。

【位置】

- https://github.com/feishi-1/petereport-zh
- https://github.com/feishi-1/petereport-zh/tree/main/app

【影响】

- 攻击者可以识别出系统中有效的用户账号。
- 为进一步的攻击（如暴力破解、社会工程学攻击等）提供基础。
- 可能导致用户隐私泄露。
- 增加系统被成功入侵的风险。
- 可能违反某些安全标准和法规要求。

【证明与验证】

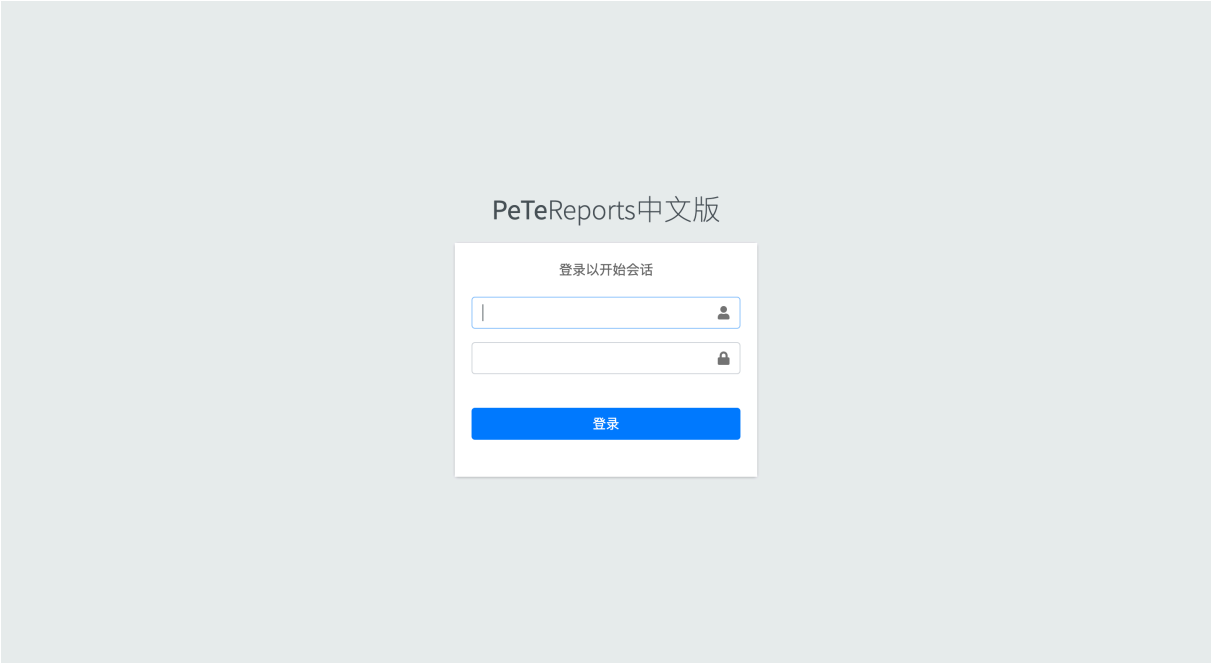


Figure 4: 账号枚举.png

【修复建议】

- 统一登录错误信息，无论是用户名不存在还是密码错误，都显示统一的消息，如”用户名或密码错误”。
- 实施登录尝试限制和延迟机制，防止大量快速尝试。
- 考虑使用验证码或其他挑战响应机制来防止自动化攻击。
- 实施多因素认证（MFA）以增加额外的安全层。
- 对登录尝试进行日志记录和监控，以检测可能的枚举攻击。