# Web应用安全性分析书

AI 聊天室 Web 应用安全评估与计划解决方案

### 系统名称

AI 聊天室 Web 应用

### 安全等级

高优先级漏洞修复

## 1 系统架构与威胁面综述

本系统采用标准三层Web架构设计：

* **展示层**：用户界面，基于 Next.js 构建
* **业务逻辑层**：核心业务处理模块，处理用户注册、登录、聊天等逻辑
* **数据访问层**：持久化数据管理，存储用户信息、聊天记录、AI Agent配置等

此类典型分层结构虽然便于管理和扩展，但也带来了**多个攻击面暴露点**，攻击者可能通过展示层注入恶意输入，绕过业务逻辑控制，最终攻击数据库或接管会话系统。

## 2 主要安全性隐患分析

### 1. 前端展示层风险

**代码依据：**

<input type="text" className="..." /> // 用于密码输入

**发现的问题：**

* **明文密码输入框**：未使用 type="password"
* **未添加验证码**：注册页面未加入验证码机制，存在暴力注册风险
* **输入未做限制**：用户输入字段未进行长度、格式、关键词限制
* **缺少防XSS处理**：用户名/聊天内容可输入脚本
* **无按钮节流机制**：按钮未防止重复点击提交
* **未添加CSRF防护机制**：未见Token或Cookie双重提交策略

### 2. 业务逻辑层风险

* **认证绕过**：如果认证机制未使用Token/Session绑定客户端特征（如IP、User-Agent），可被劫持
* **越权访问**：管理员模块（编辑Agent、查看调用次数）未设置粒度控制，可能遭到水平或垂直越权
* **敏感操作无日志记录**：例如用户收藏Agent、修改密码等操作应记录审计日志
* **暴力破解未限制尝试次数**：注册/登录接口无频率限制，易被字典攻击
* **AI Agent交互输入未过滤**：聊天内容直接传给AI，未排除注入、命令构造型攻击

### 3. 数据层风险

* **SQL/NoSQL注入**：用户信息、聊天查询、模型调用记录等接口未明确采用ORM绑定写法
* **密码存储机制**：使用强加密算法（如BCrypt）
* **聊天记录泄露风险**：若用户聊天记录展示无权限隔离，存在用户隐私数据泄漏
* **未启用数据加密存储**：高敏数据如手机号、邮箱、登录IP等应做字段级加密

## 3 安全防护策略设计

### （一）前端安全策略（Next.js 实现）

|  |  |
| --- | --- |
| **安全需求** | **实施操作** |
| 密码字段保护 | 使用 <input type="password" /> |
| 前端输入校验 | 正则校验用户名（长度、特殊符号）、密码复杂度检查 |
| 防重复提交 | 注册按钮点击后禁用，避免多次请求 |
| 验证码防注册滥用 | 集成如 reCAPTCHA v2/v3 或自定义滑动验证码 |
| CSRF防护 | 服务端生成Token并存入Cookie，前端提交时附带Header |
| XSS防护 | 所有用户输出字段使用 DOMPurify 过滤或服务端HTML转义 |
| 安全Headers | 设置 CSP、X-Frame-Options、X-Content-Type-Options 等响应头 |

### （二）业务逻辑层安全策略（Spring Security 实现）

|  |  |
| --- | --- |
| **安全机制** | **Spring Security计划配置** |
| 用户认证 | 使用BCrypt对密码加密，支持JWT或Session |
| 权限控制 | 使用 @PreAuthorize 对接口分类控制（如用户、管理员） |
| 会话管理 | 防止Session固定攻击，启用IP/User-Agent绑定 |
| 登录保护 | 启用登录失败计数机制，封锁暴力破解IP |
| CSRF保护 | 默认启用CSRF Token，使用Header传输 |
| 审计日志 | 用户操作写入日志（注册、登录、收藏、删除等） |

**其他建议：**

* 使用自定义 AccessDeniedHandler 返回更明确的权限拒绝提示
* 登录成功返回随机Token

### （三）数据层安全策略

|  |  |
| --- | --- |
| **安全需求** | **实施方案** |
| 密码加密 | 强制使用BCrypt，禁用MD5/SHA1等弱算法 |
| 数据加密存储 | 对手机号、邮箱、调用记录等进行字段级AES加密 |
| ORM使用 | 所有SQL通过JPA/MyBatis等ORM框架进行参数绑定防注入 |
| 数据脱敏 | 展示用户信息时进行脱敏处理（如 187\*\*\*\*8888） |
| 访问隔离 | 聊天记录、模型调用记录基于用户ID隔离访问 |
| 日志监控 | 登录/注册失败、异常操作写入日志系统，供审计分析 |

## 4 安全扫描与漏洞修复

## （一）扫描结果概况

检测总漏洞数：**3 个**

* 高危漏洞：**1 个**
* 中危漏洞：**1 个**
* 低危漏洞：**1 个**

## （二）关键漏洞分析与修复建议

高危：XSS（跨站脚本攻击）**高危**

#### 位置：

如注册用户名、聊天输入框字段

#### 问题描述：

用户可输入 HTML/JS 代码，前端页面未过滤或编码，可能被执行

#### 可能后果：

* 弹出恶意窗口、盗取用户Cookie、冒充用户操作系统

后端输出数据前统一做 HTML 实体转义；前端使用库如 DOMPurify 清理用户输入；禁止直接将用户输入插入DOM结构中（如 innerHTML）

中危：密码输入框未加密显示**中危**

#### 位置：

注册页面 <input type="text">（密码框）

#### 问题描述：

密码明文输入，没有变成黑点，容易被偷看或录屏记录

#### 可能后果：

* 用户隐私泄露，弱密码容易被看到并重复使用

改为 <input type="password" />

低危：缺少验证码，注册易被刷号**低危**

#### 位置：

注册接口 /api/register

#### 问题描述：

没有验证码限制，攻击者可以用脚本大量注册账号

#### 可能后果：

* 系统被刷号，占用资源，攻击者伪造用户行为

前端页面加入验证码模块（如图形验证码、滑块、人机验证）；后端校验验证码有效性；限制IP每日注册次数（如一个IP每天不超过5个账号）

## （三）处理建议表

| **漏洞类型** | **严重程度** | **处理建议** |
| --- | --- | --- |
| XSS攻击 | 高 | 前后端统一转义 |
| 明文密码输入 | 中 | 使用密码输入框 |
| 缺少验证码 | 低 | 集成图形验证码 |