Linux日志审计系统

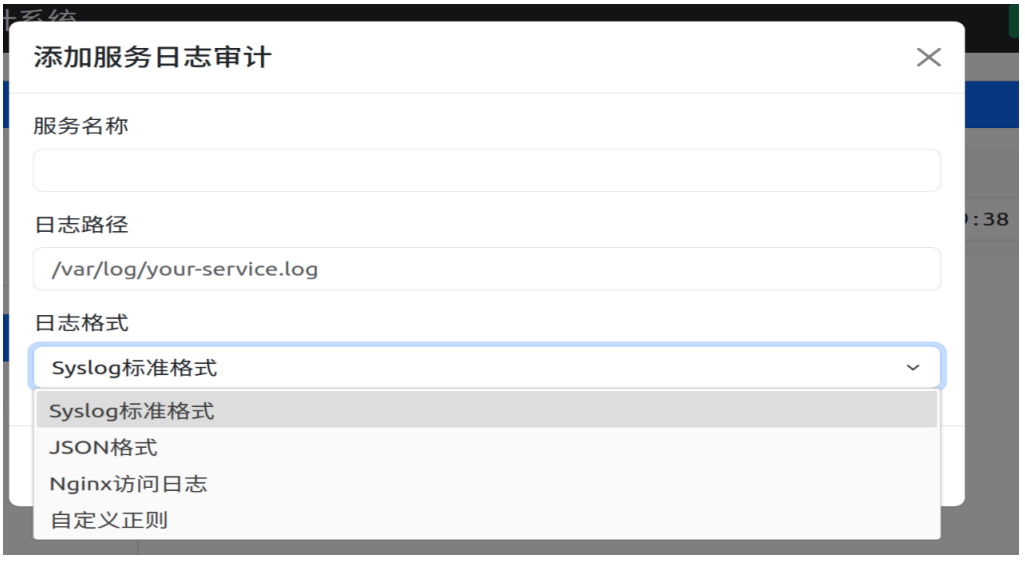
项目简介

Linux日志审计系统是一个基于Flask框架开发的Web应用程序，专注于对Linux系统日志进行集中管理和智能分析。系统通过采集、存储和分析服务器日志数据，帮助管理员快速识别系统异常和安全事件，同时提供可视化报表和告警功能

技术栈

- **后端框架**: Flask
- **前端技术**: HTML, CSS, JavaScript(*chart.js*) ,Bootstrap,
- **数据库**: MySQL

功能概述

1. 日志采集
    - **多源日志采集**
        - 支持syslog标准协议采集
        - 支持自定义应用日志采集
        - 支持Nginx/Apache等Web服务日志
        - 

2. 安全审计与分析
    - **规则引擎审计**
        - 内置100+安全审计规则
        - 支持自定义规则扩展

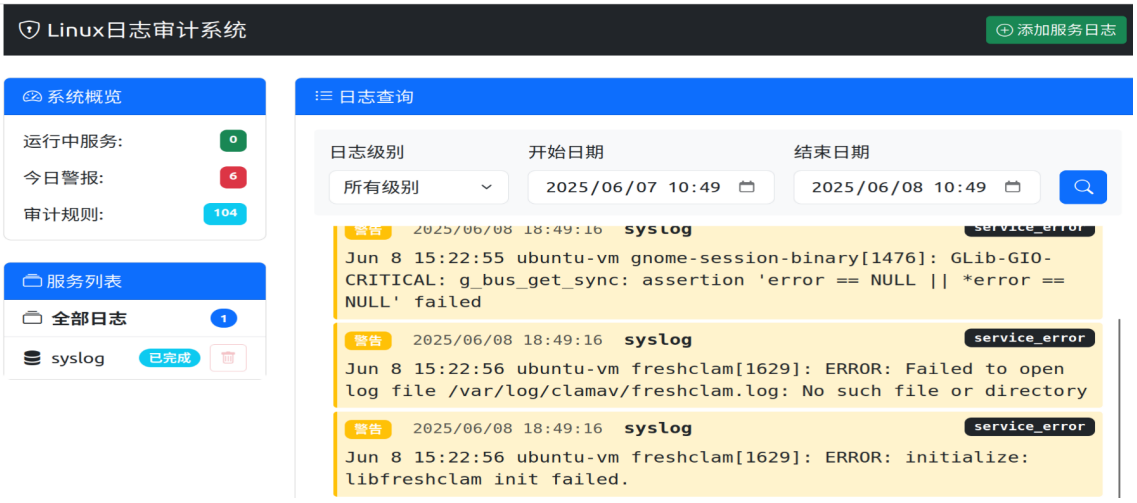
          可以通过json规则表自行导入

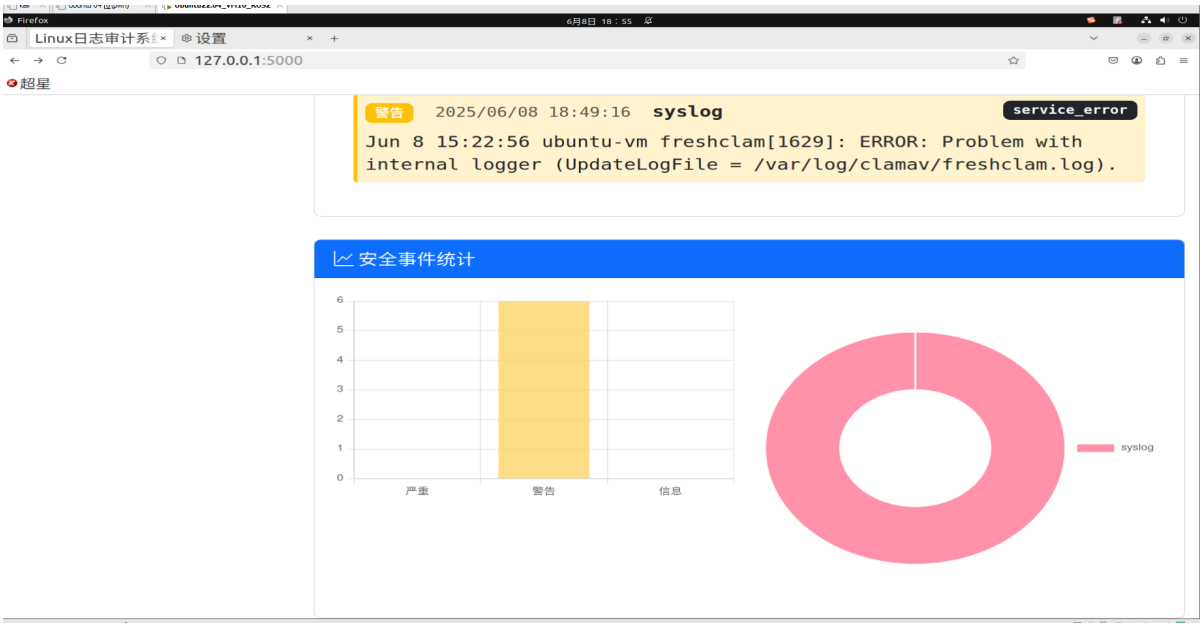
s.py | stats.py | app.py | requirements.txt | __init__.py | index.html | rules.json | config.py

```json
{
    "rules": [
        {
            "name": "failed_login",
            "pattern": "Failed password for .* from .* port \\d+",
            "level": "critical",
            "description": "SSH登录失败"
        },
        {
            "name": "sudo_command",
            "pattern": "sudo: .* : .* command not found",
            "level": "warning",
            "description": "sudo命令执行失败"
        },
        {
            "name": "disk_space",
            "pattern": "No space left on device",
            "level": "critical",
            "description": "磁盘空间不足"
        },
        {
            "name": "service_error",
            "pattern": "error|Error|ERROR",
            "level": "warning",
            "description": "服务错误"
        },
        {
            "name": "permission_denied"
```

- 通过内置规则匹配告警

3. 可视化视图
    - 系统概览
    - 服务列表
    - 安全事件统计





系统工作流程

通过提交需要审计日志服务报表。后端调用内置数据库rules表规则进行日志匹配。然后返回审计记录日志，数据，统计信息至前端界面。

数据库报表设计

表名	字段说明	约束/索引
services	记录监控的服务信息: • `name`: 服务名称 • `log_path`: 日志路径 • `log_format`: 日志格式 • `custom_regex`: 自定义正则 • `active`: 激活状态	主键: `id` 注释: 包含各字段用途说明
rules	存储审计规则: • `name`: 规则名称 • `pattern`: 匹配模式 • `level`: 告警级别 • `description`: 规则描述	主键: `id` 注释: 包含规则创建时间戳
alerts	记录触发的告警: • `service_id`: 关联服务 • `rule_name`: 触发规则 • `log_entry`: 原始日志 • `level`: 告警级别	主键: `id` 外键: `service_id` 关联services表 索引: 自动创建的外键索引
audited_logs	存储已审计日志: • `service_id`: 关联服务 • `log_entry`: 日志内容 • `audit_time`: 审计时间	主键: `id` 外键: `service_id` 唯一索引: `unique_log` 防止重复日志(限制前255字符)

表名	字段说明	约束/索引
log_positions	记录日志读取位置： • `service_id`：关联服务 • `file_path`：文件路径 • `last_position`：最后读取位置	主键：`id` 外键：`service_id` 唯一索引：`unique_file`确保每个服务文件唯一

关键设计说明：

1. **外键关系**：`alerts`、`audited_logs`、`log_positions`均通过`service_id`关联`services`表，并设置`ON DELETE CASCADE`
2. **唯一性控制**：
   • `audited_logs`表通过`unique_log`索引避免重复存储相同日志
   • `log_positions`表通过`unique_file`确保每个服务的日志文件只记录一个位置
3. **注释完善**：所有表和字段均包含详细的注释说明（COMMENT）

安装与部署

环境准备

- Python 3.10.12
- Flask 2.0.3
- MySQL 8.0
- 安装必要的依赖包
- Ubuntu 22.04.5 LTS

安装步骤

1. **解压**log_audit.zip:

```
unzip log_audit.zip
cd log_audit
```

2. **创建并激活虚拟环境**:

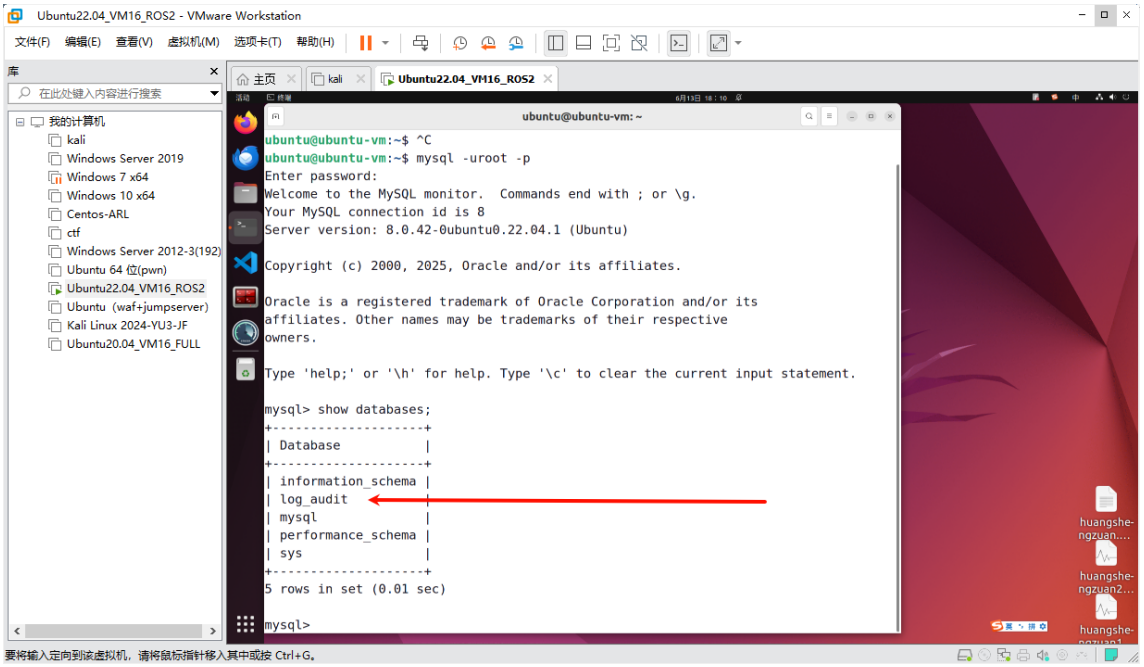
```
python -m venv venv
source venv/bin/activate  # Windows: venv\Scripts\activate
```

3. **安装依赖**:

```
pip install -r requirements.txt
pip install Werkzeug==2.0.3 --force-reinstall #降级 Werkzeug
```

4. **配置 MySQL 数据库**:
   在`config.py`文件中配置 MySQL 数据库连接：

```python
# config.py
DB_CONFIG = {
    'host': 'localhost',
    'port': 3306,
    'user': 'root',
    'password': 'root',
    'db': 'log_audit',
    'charset': 'utf8mb4'
}
```

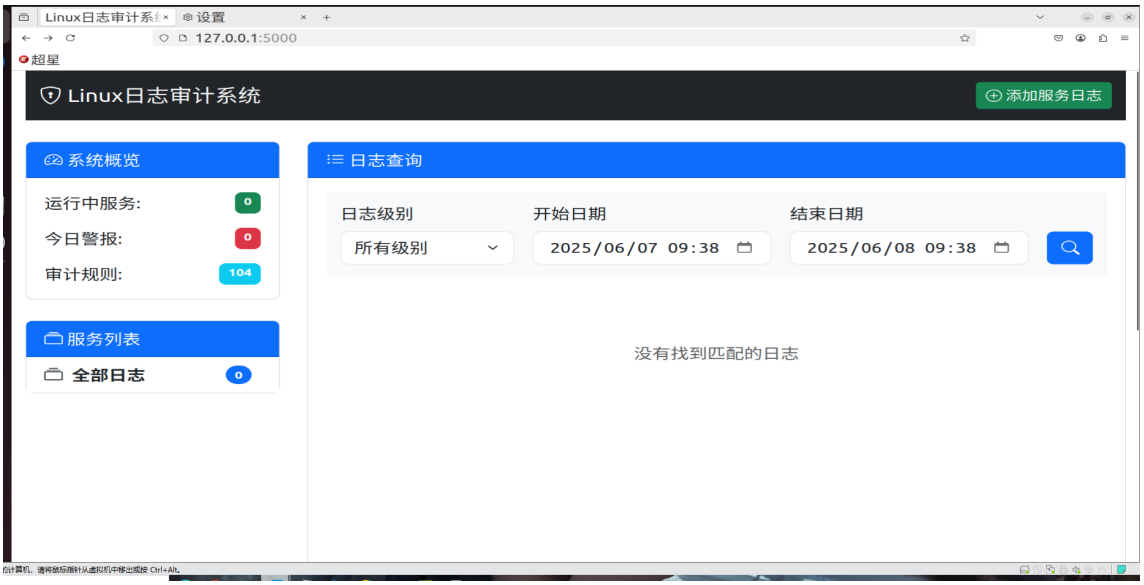



```sql
#这里得建数据库log_audit。
create database if not exists log_audit default charset=utf8;

```

5. **运行服务器**:

```
source .venv/bin/activate
python app.py
```

6. **访问应用**:
   打开浏览器，输入 `http://127.0.0.1:5000/` 即可访问应用。第一次访问界面反应有点慢。



开发指南

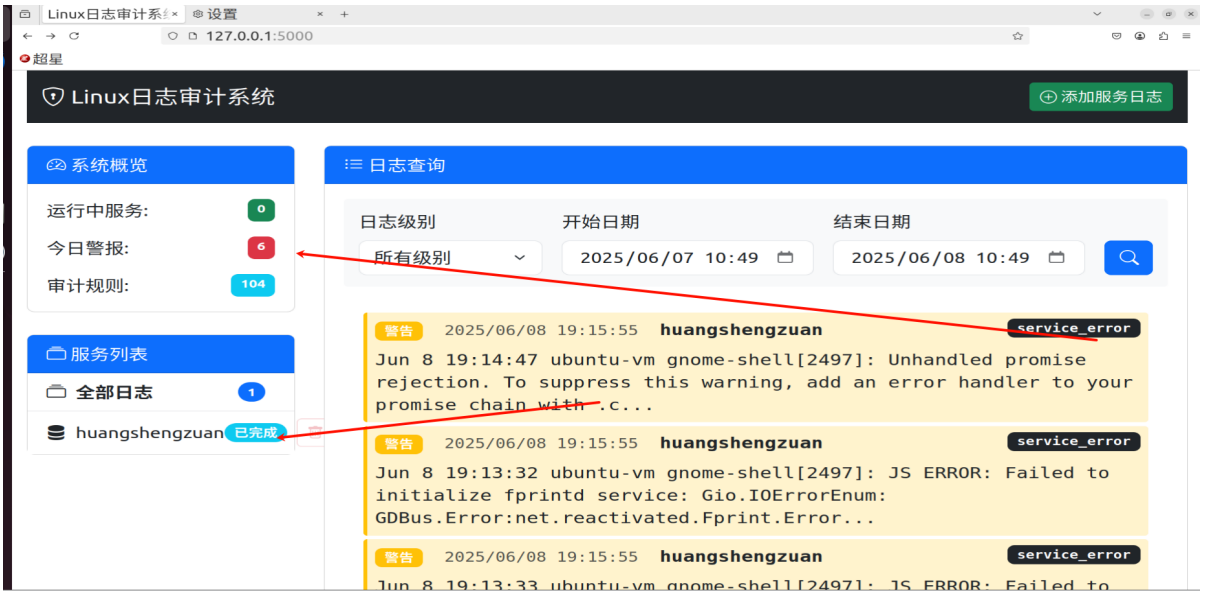
目录结构

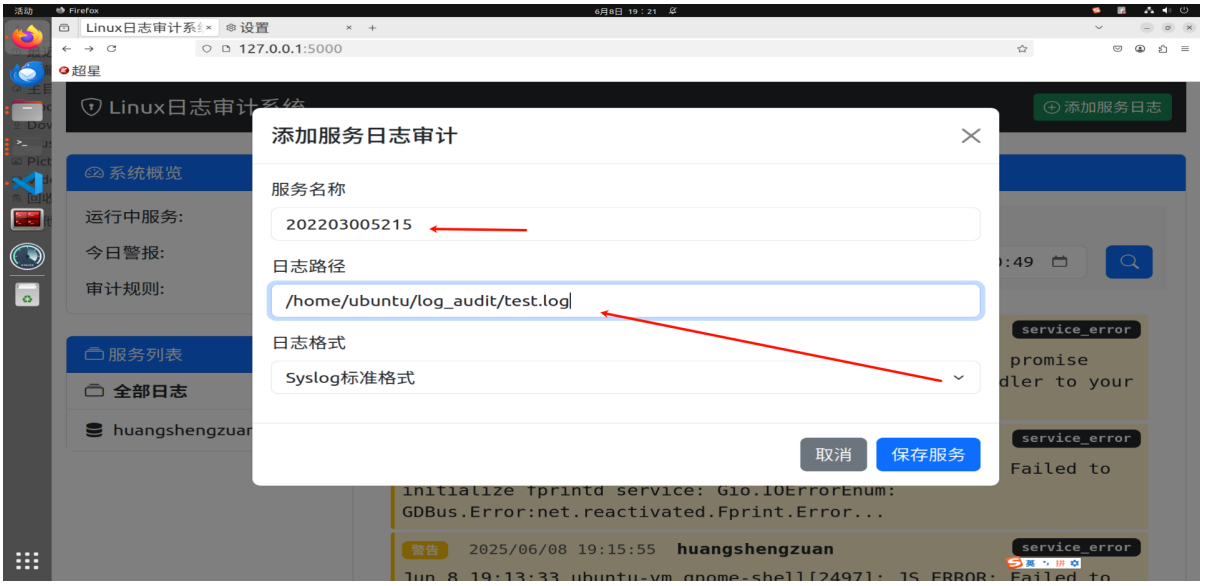
```
# Linux日志审计系统 (log_audit)

## 项目结构说明

### 核心目录
├── config/                          # 系统配置文件目录
|   ├── rules.json                   # 审计规则配置文件(JSON格式)
|   |   - 定义日志匹配规则和告警级别
|   └── services.json                # 服务监控配置文件(JSON格式)
|       - 定义需要监控的日志服务及其路径

├── config.py                        # 数据库配置文件
|   - 包含数据库连接参数(主机、端口、用户名、密码等)
|   - 可配置不同环境的数据库连接

├── db/                              # 数据库操作模块
|   ├── __init__.py                  # 数据库连接复用方法
|   |   - 提供统一的数据库连接池管理
|   |   - 实现连接重用和自动回收

### 数据模型层
├── models/                          # 数据模型层
|   ├── __init__.py                  # 模型包初始化文件
|   ├── alert.py                     # 告警模型
|   |   - 定义告警数据结构
|   |   - 处理告警的CRUD操作
|   ├── db_utils.py                  # 数据库工具
|   |   - 封装常用SQL操作
|   |   - 提供事务管理功能
|   ├── log.py                       # 日志模型
|   |   - 日志条目数据结构
|   |   - 日志解析和存储逻辑
|   ├── rule.py                      # 规则模型
|   |   - 审计规则定义
|   |   - 规则匹配引擎实现
```

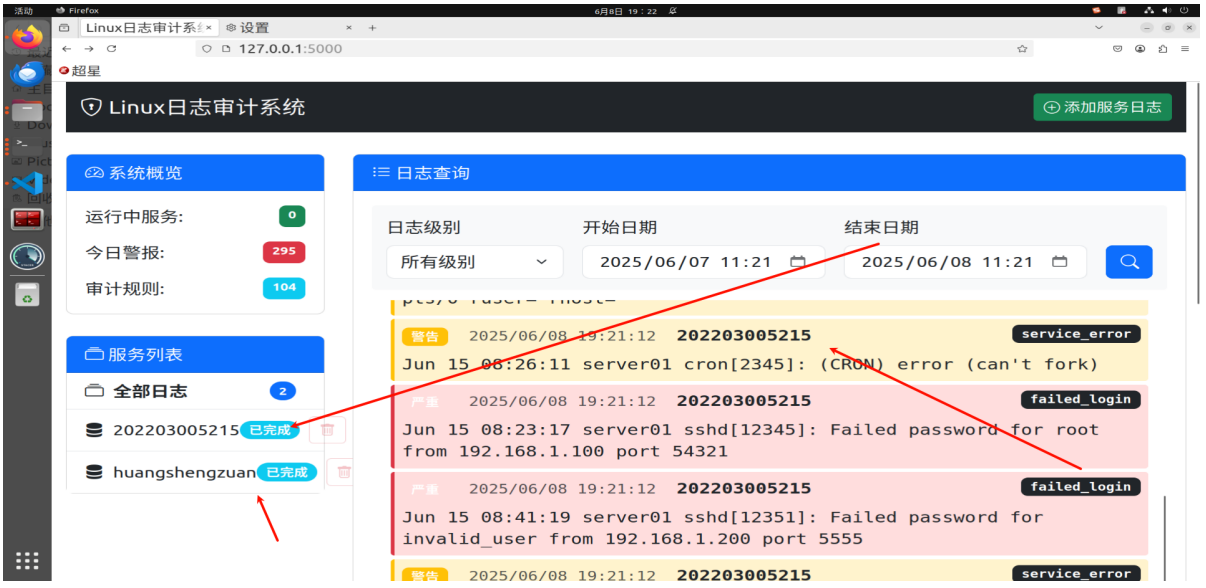
```
|   └── service.py                    # 服务模型
|       - 监控服务管理
|       - 服务状态检测逻辑

### 路由层
├── routes/                           # API路由层
|   ├── alerts.py                     # 告警相关API
|   |   - 告警查询/创建/删除接口
|   ├── index.py                      # 首页路由
|   ├── __init__.py                   # 路由初始化
|   ├── logs.py                       # 日志查询API
|   |   - 日志检索接口
|   |   - 日志分析接口
|   ├── services.py                   # 服务管理API
|   |   - 服务注册/注销接口
|   |   - 服务状态查询
|   └── stats.py                      # 统计信息API
|       - 系统运行统计
|       - 图表数据接口

### 主程序
├── __init__.py                       # 项目初始化文件
├── requirements.txt                  # 项目依赖列表
|   - Flask/PyMySQL等依赖包及版本
├── app.py                            # 应用入口
|   - Flask应用初始化
|   - 路由注册
|   - 中间件配置

### 前端资源
├── templates/                        # 前端模板
|   └── index.html                    # 主界面模板
|       - 基于Bootstrap的管理界面
|       - 实时日志展示区
|       - 统计图表区

### 测试资源
├── syslog/                           # 系统日志样例
|   - 用于测试的系统日志文件
└── test.log                          # 测试日志文件
    - 通用日志格式测试数据

```

系统演习

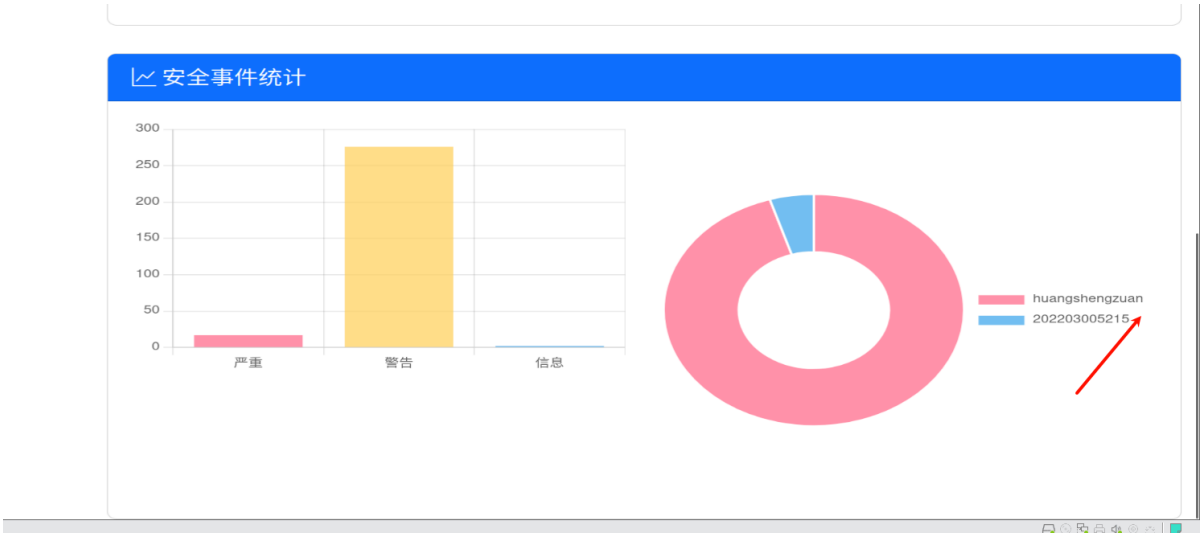
1.添加服务日志审计


Linux日志审计系统
添加服务日志
系统概览
运行中服务: 0
今日警报: 6
审计规则: 104
服务列表
全部日志 1
huangshengzuan 已完成
日志查询
日志级别
开始日期
结束日期
所有级别
2025/06/07 10:49
2025/06/08 10:49
警告 2025/06/08 19:15:55 huangshengzuan service_error
Jun 8 19:14:47 ubuntu-vm gnome-shell[2497]: Unhandled promise rejection. To suppress this warning, add an error handler to your promise chain with .c...
警告 2025/06/08 19:15:55 huangshengzuan service_error
Jun 8 19:13:32 ubuntu-vm gnome-shell[2497]: JS ERROR: Failed to initialize fprintd service: Gio.IOErrorEnum: GDBus.Error:net.reactivated.Fprint.Error...
警告 2025/06/08 19:15:55 huangshengzuan service_error


添加服务日志审计
服务名称
202203005215
日志路径
/home/ubuntu/log_audit/test.log
日志格式
Syslog标准格式
取消
保存服务


Linux日志审计系统
添加服务日志
系统概览
运行中服务: 0
今日警报: 295
审计规则: 104
服务列表
全部日志 2
202203005215 已完成
huangshengzuan 已完成
日志查询
日志级别
开始日期
结束日期
所有级别
2025/06/07 11:21
2025/06/08 11:21
警告 2025/06/08 19:21:12 202203005215 service_error
Jun 15 08:26:11 server01 cron[2345]: (CRON) error (can't fork)
严重 2025/06/08 19:21:12 202203005215 failed_login
Jun 15 08:23:17 server01 sshd[12345]: Failed password for root from 192.168.1.100 port 54321
严重 2025/06/08 19:21:12 202203005215 failed_login
Jun 15 08:41:19 server01 sshd[12351]: Failed password for invalid_user from 192.168.1.200 port 5555
警告 2025/06/08 19:21:12 202203005215 service_error

2.安全事件统计



3.数据库审计日志记录

```
mysql> use log_audit;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+--------------------+
| Tables_in_log_audit |
+--------------------+
| alerts             |
| audited_logs       |
| log_positions      |
| rules              |
| services           |
+--------------------+
5 rows in set (0.00 sec)

mysql> select * from log_positions;
+----+------------+-------------------------------+---------------+---------------------+---------------------+
| id | service_id | file_path                     | last_position | last_modified       | last_check          |
+----+------------+-------------------------------+---------------+---------------------+---------------------+
|  2 |          2 | /var/log/syslog               |       1206280 | 2025-06-08 19:14:02 | 2025-06-08 19:15:55 |
|  3 |          3 | /home/ubuntu/log_audit/test.log |          3813 | 2025-05-03 21:23:33 | 2025-06-08 19:21:12 |
+----+------------+-------------------------------+---------------+---------------------+---------------------+
2 rows in set (0.00 sec)

mysql>
```

项目特点

1. **模块化设计**：清晰的分层结构(MVC模式)
2. **配置化**：审计规则和服务监控完全通过JSON配置
3. **可扩展**：易于添加新的日志解析规则
4. **安全性**：数据库连接池管理和参数化查询

可以通过查看各模块的详细注释了解具体实现细节。

系统不足

1.前端静态样式调用网络资源，需联网才能使用。

2.审计功能可以引入实时审计更新。

3.搜索功能未完善。

总结

由于时间原因有许多地方或有不足。不过对我自己来说该项目能从零到有，也是相当锻炼到自己。