题 目：业务安全风控中台的设计与实现

学院： 软件学院 专业： 软件工程 学生姓名： 学号：

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **项目背景** 2. 项目来源   本项目来源于本人自2021年9约开始在北京自如信息科技有限公司—信息安全部门的实习工作。实习过程中，本人主要负责该项目整体设计及后端编写。   1. 项目意义   随着互联网的发展，有一批黑色产业也在随之兴起，他们依靠互联网为媒介，利用非法手段从中获益，称之为网络黑产。2018年5月发布的《数字金融反欺诈白皮书》显示，2017年我国黑产从业人员超过150万人，年产值达千亿元级别。虽然近些年没有类似的数据统计，但是依据目前网络发展趋势，这个数字近几年一直保持增长趋势。网络黑产遍布在互联网的各个领域，不论平台为用户提供了怎样一款产品，只要有可以转化为收益的地方，他们就会通过恶意注册账号开展网络诈骗、在平台发布恶意违法内容、利用平台交易渠道进行虚假交易、薅取营销活动和新人礼包优惠等方式，对平台进行攻击，从而从中获益。2019年，黑灰产团伙通过过期的优惠券漏洞攻击拼多多，获利数千万元。  为了对抗网络黑产，减少公司损失，保障用户利益，风控产品也应运而生。风控产品主要分为两大类，基础安全风控和业务安全风控。基础安全风控主要为了保障指代码安全、信息安全和数据安全，业务安全风控主要为了保障业务过程中的账号安全、内容安全、营销安全、交易安全和资金安全。本项目所设计的业务安全风控中台，为业务安全风控产品，通过建立公司范围的风控体系，设定一系列风控策略和规则，在黑灰产攻击平台时迅速识别，进行拦截或者反攻击，从而保障平台安全。   1. 项目的研发现状（目前同类产品或解决方案中存在哪些不足，或现实情况下，存在哪些问题，例如：效率低下、手工操作等）   目前，公司内部的风控系统建设刚刚起步，曾经设置过公司层面的风控规则，小部分业务领域有自己的风控策略，但是存在以下问题：  （1）虽然小部分业务领域有自己的风控策略，但是均较为简单，如设置黑名单、限制同一用户某一操作单天次数上限等，容易被绕过。各个业务领域之间的风控策略和结果无法共享，若某一业务领域发现黑灰产账号，仅在该业务领域内对其进行防范，但对其余业务领域依旧存在风险。  （2）曾经的公司层面风控规则，仅仅是每天针对当天的用户操作数据，按照特定的规则进行统计，将统计数据导出为excel进行人工筛查，如果有疑似黑产用户，再进行相应调查。这种方式十分机械，筛查为纯手工操作，效率低下。规则简单、单一，所有规则为硬编码方式实现，不方便进行修改。并且规则和结果均受到时间限制，针对具有时间跨度的数据无法进行分析，不同时间产生的结果也不方便进行横向比较。由于这种方式过于机械，目前已不再使用。  （3）公司层面没有针对风控形成体系，即公司整体风险情况以及每个用户的风险情况无法进行评估和可视化展示。 | | | |
| **二、项目内容、项目目标与拟解决的关键问题**  **1. 项目的内容（范围）**  业务安全中台是搭建在用户和公司各个子系统之间的的风控平台，接收用户操作，并根据中台内部的规则体系对用户操作进行分析，将高风险的用户操作进行拦截，并针对中风险的用户操作向子系统提供防控建议，如发起短信校验、唤起实名认证、拒绝本次请求等等。  本项目功能框架如图所示：   1. **项目目标（即预期交付物）**   设计并实现一个业务安全中台。系统主要包括两个部分，外部安全防控和内部安全防控。  内部安全防控主要针对公司内部员工，前期已经针对公司内部系统的数据进行了分级处理，并且在一定保密级别的数据上进行了埋点，记录员工点击查看数据的操作日志。在该系统中，提供员工保密数据查看次数配置功能，可根据员工身份定义一定时间段内的查看次数上线阈值，当员工查看次数超过阈值后，向员工直系领导发送告警信息。同时系统支持告警记录搜索查看。  外部安全防控主要针对平台用户，包括四个模块：用户风险分数管理模块，黑白名单库管理模块，规则引擎管理模块，规则命中管理模块。  用户风险分数管理模块：每个平台用户都有一个在平台内的风险总分和所有业务线的小分，该模块实现所有用户的分数查看、用户操作查看以及平台所有用户分数分布查看功能。  除此之外，若用户进行了某些风险操作，会被系统标为风险用户，在该模块下可以单独查看所有风险用户的风险分数。  黑白名单库管理模块：针对用户id、手机号、IP地址、设备号（IMEI）设立4个黑白名单，在规则引擎中可以设立针对黑白名单定义相应规则。在该模块中，实现对黑白名单的增删改查。  规则引擎管理模块：该模块为项目核心。规则，即计算用户风险分数变化的依据。如某一账号1分钟内登陆次数大于5次；某一设备号5分钟内登陆账号个数超过3个。同时，规则构成规则组，规则组内和组间可以设置不同的执行逻辑，如顺序执行、条件分支。  在该模块中，实现对所有规则的查看，修改已有规则，新增规则，管理规则和规则组的生效开关及执行顺序。  规则命中管理模块：实现所有规则命中情况的统计查看功能，包括每条规则的出发次数、命中次数、命中率以及具体命中名单。   1. **拟解决的关键问题**   （1）规则引擎管理模块中规则自由配置的设计与实现问题：  由于业务逻辑复杂，相应的风控规则也十分复杂，因此必须采用可配置的方式实现规则引擎的逻辑，在规则执行过程中可动态对规则进行调整。  一般来说，规则包括特征、操作符、阈值三个要素。如对于规则“某一账号1分钟登陆次数大于5次”来说，特征为“同一用户id一分钟内登录次数”，操作符为“大于”，阈值为“5”。触发规则后，后续应该有对应的惩罚措施，如降低信用分数、加入黑名单等等。  根据业务场景的不同，不同的规则之间需要进行复用，同时所有的规则以及惩罚措施需要根据业务需要进行动态调整，因此需要设计并一套机制，方便进行调整和配置。  （2）数据量增大后的处理速度及系统性能提升：  规则的执行基础是数据，而随着规则体系与业务领域的不断扩大，需要记录、分析的数据量也是庞大的。根据统计，自如信息科技旗下所有产品每天仅登陆注册数据约40万条，再加上用户订单数据、操作数据等等，单日数据量上千万，数据规模是庞大的。在如此大量的数据处理过程中，如何将规则执行导致的延迟控制在用户和业务方可接受范围内。  （3）在开发时简便高效对数据库中的实体进行操作：  该系统逻辑十分复杂，涉及到的实体数量也非常多，但是在数据库中对于实体的操作均可抽象为增删改查，因此需要实用工具实现对数据库的高效操作。  （4）频繁读取数据库导致的处理延迟解决：  在执行规则时，无法避免的需要反复读取数据库以获取数据进行计算，关系型数据库的处理速度无法应对该系统过于频繁的请求。 | | | |
| **三、拟采取的解决方案及可行性分析**  **1. 理论依据（可选）**  （1）Spring Boot：  Spring Boot时近年来业内兴起的基于Spring架构体系的一种全新开发框架，其诞生之初便用来简化Spring框架的繁杂配置化工作。该框架通过约定的方式（yml/properties文件、缩进等）来进行配置，从而使开发人员避免复杂的配置定义。同时，Spring Boot继承了Spring的MVC设计模式，提高了开发效率。  （2）Flink流式计算引擎：  Apache Flink是由Apache软件基金会开发的开源流式处理框架，用于对无界和有界数据流进行有状态计算。相比于大数据计算引擎Spark而言，Flink在流式计算方面有着优势，不仅计算延迟低，且吞吐量也有着明显的优势。  **2. 拟采用的技术路线**  系统后端采用springBoot框架进行搭建，前端采用vue框架，数据库采用mysql+redis实现，通过redis进行缓存提高读写速度。数据处理部分采用flink大数据框架。  针对2.3中罗列的关键问题，下面给出拟采用技术方案。  （1）规则引擎管理模块中规则自由配置的设计与实现问题：  将事件、规则和规则组抽象为实体。一个事件对应0或多个规则组，一个规则组内包含0或多个规则。规则的属性包括特征、特征对应时间窗口、操作符、阈值、处罚措施，其中，特征和处罚措施均通过预先定义写入系统。规则组实体中，包括规则列表以及组内执行顺序两个属性。事件实体中，包括规则组列表以及规则组执行顺序。这三个实体中的属性均可人为配置和修改，通过这种方法实现规则引擎的自由配置与执行。  （2）数据量增大后的处理速度及系统性能提升：  将规则执行与规则特征的计算进行分离。业务方通过MQ消息队列将用户数据传入系统，flink订阅数据，按照特征需要完成原子粒度的聚合，并将计算得到的指标写入redis。在规则执行过程中，根据特征对应的时间窗口从redis中读取需要的数据，再根据操作符和阈值进行判断是否命中。通过这种方法减小规则执行过程中数据处理的时间，提升系统性能。  （3）在开发时简便高效对数据库中的实体进行操作：  项目采用mybatis框架减少传统的JDBC操作，通过实体类和对应Mapper接口的编写实现对数据库的操作。同时，系统的数据流转采用Dto（Data Transfer Object 数据传输对象）进行，DTO 是一组需要跨进程或网络边界传输的聚合数据的简单容器，可以减少参数传递个数。  （4）频繁读取数据库导致的处理延迟解决：  项目采用mysql+redis的形式解决读写延迟问题。需要写入数据时，首先将数据写入mysql数据库，并刷新redis缓存数据。读取数据时，首先从redis中进行查询，若查询不到结果再前往mysql数据库进行查找，并把内容更新进入redis。由于redis的读写速度远远高于mysql，使用redis进行缓存可以提高读写速度。  **3. 可行性分析**  （1）前期基础：  ① 对于内部安全防控模块，公司前期已经完成内部系统埋点。除此之外，针对外部用户的主要操作，如登陆注册、下单、抢券等，也有完整的埋点数据。这些数据均可直接使用。  ② 公司内部有完整的MQ消息队列技术，接口完善可以直接调用。  ③ 公司内部有完整的权限管理平台，不同角色在本系统上的权限控制可以直接在权限管理平台上进行配置。  （2）理论基础：  本项目所用的主要技术包括spring boot、vue、mysql+redis、flink框架均为工业生产中非常成熟的技术。主要难点为flink大数据处理分析，而风控场景是flink主要应用场景之一，且基于 Apache Flink 和规则引擎的实时风控解决方案在业内也有先例，因此技术实现方面具有可行性。  （3）已有的积累  本人在校期间多次接触web项目，虽然没有使用过springBoot框架，但也有所了解。同时，熟悉mysql和redis的相关操作。在大数据处理方面，曾经接触过spark框架，对于大数据处理有一定经验。 | | | |
| 主要参考文献：  [1] 陈亚军, 肖海斌. 基于Spring Boot和Vue框架的水库雨水情监测系统设计与实现[J]. 软件 2021,42(09),29-31+41.  [2] 刘广轩,黄山,胡佳丽,段晓东.面向Flink流处理框架的主动备份容错优化[J/OL].浙江大学学报(工学版):1-9[2022-02-23].  [3] 封宇,周杰. 基于Flink的景区人数监测平台设计[J]. 电子技术. 2021(11).  [4] 马云,七七.当互联网业务安全来到反欺诈风口[J].上海信息化,2017(09):63-65.  [5] 常昊. 互联网金融风控系统贷中风控和预警功能的设计与实现[D].南京大学,2020.DOI:10.27235/d.cnki.gnjiu.2020.001604. | | | |
| 毕业设计（论文）进度安排： | | | |
| 序号 | 毕业设计（论文）各阶段内容 | 时间安排 | 备注 |
| 1 | 参与完成整个系统的需求分析； | 2021.10.8至2021.10.17 |  |
| 2 | 参与完成系统的总体设计； | 2021.10.18至2021.11.12 |  |
| 3 | 设计并实现内部安全防控模块； | 2021.11.13至2021.12.10 |  |
| 4 | 设计并实现用户风险分数管理模块 | 2021.12.11至2021.12.31 |  |
| 5 | 设计并实现黑白名单库管理模块 | 2022.1.1至2022.1.10 |  |
| 6 | 设计并实现规则引擎模块 | 2022.1.11至2022.2.28 |  |
| 7 | 设计并实现规则命中管理模块 | 2022.3.1至2022.3.20 |  |
| 8 | 进行集成测试、确认测试、系统测试； | 2022.3.21至2022.4.10 |  |
| 9 | 撰写毕业论文 | 2022.4.11至2022.5.1 |  |
| 指导教师意见：  填写说明：查阅资料是否全面，提出的研究方案和计划进度是否可行，还有什么需要注意和改进的方面，是否同意按学生提出的计划进行等。（**填写后请删除该说明）** | | | |

指导教师（审核签名）： 审核日期： 年 月 日