胖友们，注意安全啊！如何设计一个安全的对外接口？

芋道源码 2020-02-03

来源：ksfzhaohui

my.oschina.net/OutOfMemory/blog/3131916

前言

最近有个项目需要对外提供一个接口，提供公网域名进行访问，而且接口和交易订单有关，所以安全性很重要；这里整理了一下常用的一些安全措施以及具体如何去实现。

安全措施

个人觉得安全措施大体来看主要在两个方面，一方面就是如何保证数据在传输过程中的安全性，另一个方面是数据已经到达服务器端，服务器端如何识别数据，如何不被攻击；下面具体看看都有哪些安全措施。

1.数据加密

我们知道数据在传输过程中是很容易被抓包的，如果直接传输比如通过http协议，那么用户传输的数据可以被任何人获取；所以必须对数据加密，常见的做法对关键字段加密比如用户密码直接通过md5加密；现在主流的做法是使用https协议，在http和tcp之间添加一层加密层(SSL层)，这一层负责数据的加密和解密；

2.数据加签

数据加签就是由发送者产生一段无法伪造的一段数字串，来保证数据在传输过程中不被篡改；你可能会问数据如果已经通过https加密了，还有必要进行加签吗？数据在传输过程中经过加密，理论上就算被抓包，也无法对数据进行篡改；但是我们要知道加密的部分其实只是在外网，现在很多服务在内网中都需要经过很多服务跳转，所以这里的加签可以防止内网中数据被篡改；

3.时间戳机制

数据是很容易被抓包的，但是经过如上的加密，加签处理，就算拿到数据也不能看到真实的数据；但是有不法者不关心真实的数据，而是直接拿到抓取的数据包进行恶意请求；这时候可以使用时间戳机制，在每次请求中加入当前的时间，服务器端会拿到当前时间和消息中的时间相减，看看是否在一个固定的时间范围内比如5分钟内；这样恶意请求的数据包是无法更改里面时间的，所以5分钟后就视为非法请求了；

4.AppId机制

大部分网站基本都需要用户名和密码才能登录，并不是谁来能使用我的网站，这其实也是一种安全机制；对应的对外提供的接口其实也需要这么一种机制，并不是谁都可以调用，需要使用

接口的用户需要在后台开通appid，提供给用户相关的密钥；在调用的接口中需要提供appid+密钥，服务器端会进行相关的验证；

5.限流机制

本来就是真实的用户，并且开通了appid，但是出现频繁调用接口的情况；这种情况需要给相关appid限流处理，常用的限流算法有令牌桶和漏桶算法；

6.黑名单机制

如果此appid进行过很多非法操作，或者说专门有一个中黑系统，经过分析之后直接将此appid列入黑名单，所有请求直接返回错误码；

7.数据合法性校验

这个可以说是每个系统都会有的处理机制，只有在数据是合法的情况下才会进行数据处理；每个系统都有自己的验证规则，当然也可能有一些常规性的规则，比如身份证长度和组成，电话号码长度和组成等等；

如何实现

以上大体介绍了一下常用的一些接口安全措施，当然可能还有其他我不知道的方式，希望大家补充，下面看看以上这些方法措施，具体如何实现；

1.数据加密

现在主流的加密方式有对称加密和非对称加密；**对称加密** ：对称密钥在加密和解密的过程中使用的密钥是相同的，常见的对称加密算法有DES，AES；优点是计算速度快，缺点是在数据传送前，发送方和接收方必须商定好秘钥，然后使双方都能保存好秘钥，如果一方的秘钥被泄露，那么加密信息也就不安全了；**非对称加密** ：服务端会生成一对密钥，私钥存放在服务器端，公钥可以发布给任何人使用；优点就是比起对称加密更加安全，但是加解密的速度比对称加密慢太多了；广泛使用的是RSA算法；

两种方式各有优缺点，而https的实现方式正好是结合了两种加密方式，整合了双方的优点，在安全和性能方面都比较好；

对称加密和非对称加密代码实现，jdk提供了相关的工具类可以直接使用，此处不过多介绍；关于https如何配置使用相对来说复杂一些，可以参考本人的之前的文章HTTPS分析与实战

2.数据加签

数据签名使用比较多的是md5算法，将需要提交的数据通过某种方式组合和一个字符串，然后通过md5生成一段加密字符串，这段加密字符串就是数据包的签名，可以看一个简单的例子：

```
str：参数1={参数1}&参数2={参数2}&……&参数n={参数n}$key={用户密钥};
MD5.encrypt(str);
```

注意最后的用户密钥，客户端和服务端都有一份，这样会更加安全；

3.时间戳机制

解密后的数据，经过签名认证后，我们拿到数据包中的客户端时间戳字段，然后用服务器当前时间去减客户端时间，看结果是否在一个区间内，伪代码如下：

```
long interval=5*60*1000；//超时时间
long clientTime=request.getparameter("clientTime");
long serverTime=System.currentTimeMillis();
if(serverTime-clientTime>interval){
    returnnew Response("超过处理时长")
}
```

4.AppId机制

生成一个唯一的AppId即可，密钥使用字母、数字等特殊字符随机生成即可；生成唯一AppId根据实际情况看是否需要全局唯一；但是不管是否全局唯一最好让生成的Id有如下属性：**趋势递增** ：这样在保存数据库的时候，使用索引性能更好；**信息安全** ：尽量不要连续的，容易发现规律；关于全局唯一Id生成的方式常见的有类snowflake方式等；

5.限流机制

常用的限流算法包括：令牌桶限流，漏桶限流，计数器限流；**1.令牌桶限流**令牌桶算法的原理是系统以一定速率向桶中放入令牌，填满了就丢弃令牌；请求来时会先从桶中取出令牌，如果能取到令牌，则可以继续完成请求，否则等待或者拒绝服务；令牌桶允许一定程度突发流量，只要有令牌就可以处理，支持一次拿多个令牌；**2.漏桶限流**漏桶算法的原理是按照固定常量速率流出请求，流入请求速率任意，当请求数超过桶的容量时，新的请求等待或者拒绝服务；可以看出漏桶算法可以强制限制数据的传输速度；**3.计数器限流**计数器是一种比较简单粗暴的算法，主要用来限制总并发数，比如数据库连接池、线程池、秒杀的并发数；计数器限流只要一定时间内的总请求数超过设定的阀值则进行限流；

具体基于以上算法如何实现，Guava提供了RateLimiter工具类基于基于令牌桶算法：

```
RateLimiter rateLimiter = RateLimiter.create(5);
```

以上代码表示一秒钟只允许处理五个并发请求，以上方式只能用在单应用的请求限流，不能进行全局限流；这个时候就需要分布式限流，可以基于redis+lua来实现；

6.黑名单机制

如何为什么中黑我们这边不讨论，我们可以给每个用户设置一个状态比如包括：初始化状态，正常状态，中黑状态，关闭状态等等；或者我们直接通过分布式配置中心，直接保存黑名单列表，每次检查是否在列表中即可；

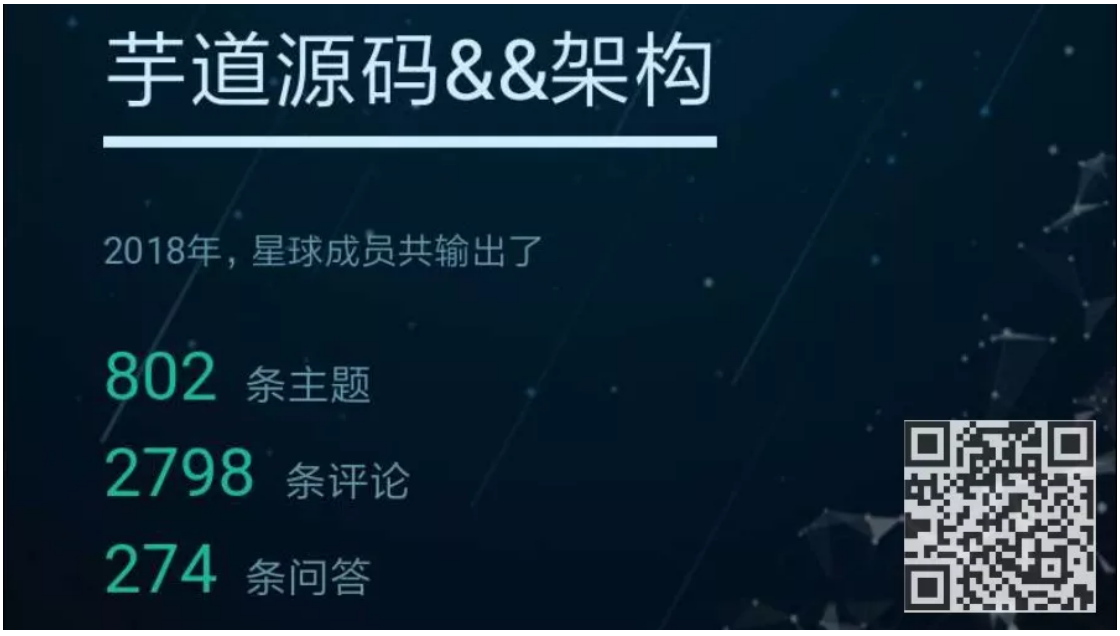
7.数据合法性校验

合法性校验包括：常规性校验以及业务校验；常规性校验：包括签名校验，必填校验，长度校验，类型校验，格式校验等；业务校验：根据实际业务而定，比如订单金额不能小于0等；

总结

本文大致列举了几种常见的安全措施机制包括：数据加密、数据加签、时间戳机制、AppId机制、限流机制、黑名单机制以及数据合法性校验；当然肯定有其他方式，欢迎补充。

欢迎加入我的知识星球，一起探讨架构，交流源码。加入方式，**长按下方二维码噢**：



已在知识星球更新源码解析如下：

《精尽面试题（附答案）》	《精尽学习指南（附视频）》
01. Dubbo 面试题	00. 精尽学习指南 —— 路线
02. Netty 面试题	01. Dubbo 学习指南
03. Spring 面试题	02. Netty 学习指南
04. Spring MVC 面试题	03. Spring 学习指南
05. Spring Boot 面试题	04. Spring MVC 学习指南
06. Spring Cloud 面试题	05. Spring Boot 学习指南
07. MyBatis 面试题	06. Spring Cloud 学习指南
08. 消息队列面试题	07. MyBatis 学习指南
09. RocketMQ 面试题	08. RocketMQ 学习指南
10. RabbitMQ 面试题	09. RabbitMQ 学习指南
11. Kafka 面试题	10. Kafka 学习指南
12. 缓存面试题	11. Redis 学习指南
13. Redis 面试题	12. MySQL 学习指南
14. MySQL 面试题	13. MongoDB 学习指南
15.【分库分表】面试题	14. Elasticsearch 学习指南
16.【分布式事务】面试题	15. 设计模式学习指南
17. Elasticsearch 面试题	16. Java【基础】学习指南
18. MongoDB 面试题	17. Java【并发】学习指南
19. 设计模式面试题	18. Java【虚拟机】学习指南
20. Java【基础】面试题	19. Linux 学习指南
21. Java【集合】面试题	20. 数据结构与算法学习指南
22. Java【并发】面试题	21. 计算机网络学习指南
23. Java【虚拟机】面试题	22. Maven 学习指南
24. Linux 面试题	23. Jenkins 学习指南
25. Git 面试题	24. Git 学习指南
26. 计算机网络面试题	25. Intellij IDEA 学习指南
27. Maven 面试题	26. Docker 学习指南
28. Jenkins 面试题	27. Kubernetes 学习指南
29. Zookeeper 面试题	28. Zookeeper 学习指南
30. Nginx 面试题	29. Nginx 学习指南

《Dubbo 源码解析》

01. 调试环境搭建
02. 项目结构一览
03. 配置 Configuration
04. 核心流程一览
05. 拓展机制 SPI
06. 线程池 ThreadPool
07. 服务暴露 Export
08. 服务引用 Refer
09. 注册中心 Registry
10. 动态编译 Compile
11. 动态代理 Proxy
12. 服务调用 Invoke
13. 调用特性
14. 过滤器 Filter
15. NIO 服务器
16. P2P 服务器
17. HTTP 服务器
18. 序列化 Serialization
19. 集群容错 Cluster
20. 优雅停机 Shutdown
21. 日志适配 Logging
22. 状态检查 Status
23. 监控中心 Monitor
24. 管理中心 Admin
25. 运维命令 QOS
26. 链路追踪 Tracing
27. Spring Boot 集成
28. Spring Cloud 集成

... 一共 **73+** 篇

《Spring Cloud 源码解析》

01. 网关 Spring Cloud Gateway 25 篇
02. 注册中心 Eureka 23 篇
03. 熔断器 Hystrix 9 篇
04. 配置中心 Apollo 32 篇
05. 链路追踪 SkyWalking 38 篇
06. 调度中心 Elastic Job 24 篇

《Netty 源码解析》

01. 调试环境搭建
02. NIO 基础
03. Netty 简介
04. 启动 Bootstrap
05. 事件轮询 EventLoop
06. 通道管道 ChannelPipeline
07. 通道 Channel
08. 字节缓冲区 ByteBuf
09. 通道处理器 ChannelHandler
10. 编解码 Codec
11. 工具类 Util

... 一共 **61+** 篇

《MyBatis 源码解析》

01. 调试环境搭建
02. 项目结构一览
03. MyBatis 初始化
04. SQL 初始化
05. SQL 执行
06. 插件体系
07. Spring 集成

... 一共 **34+** 篇

《Spring 源码解析》

01. 调试环境搭建
02. IoC Resource 定位
03. IoC BeanDefinition 载入
04. IoC BeanDefinition 注册
05. IoC Bean 获取
06. IoC Bean 生命周期
07. AOP 源码导读
08. Transaction 源码导读

... 一共 **46+** 篇

《Spring Boot 源码解析》

01. 调试环境搭建
02. 项目结构一览
03. SpringApplication
04. 自动配置
05. Condition
06. ServletWebServerApplicationContext
07. ReactiveWebServerApplicationContext
08. ApplicationContextInitializer
09. ApplicationListener

... 一共 **15+** 篇（努力更新中）

《Spring MVC 源码解析》

01. 调试环境搭建
02. 容器的初始化
03. 组件一览
04. 请求处理一览
05. HandlerMapping 组件
06. HandlerAdapter 组件
07. HandlerExceptionResolver 组件
08. RequestToViewNameTranslator 组件
09. LocaleResolver 组件
10. ThemeResolver 组件
11. ViewResolver 组件
12. MultipartResolver 组件
13. FlashMapManager 组件

... 一共 **24+** 篇

《数据库实体设计》

01. 商品模块
02. 交易模块
03. 营销模块
04. 公用模块

... 一共 **17+** 篇

芋道源码

《JDK 源码解析》

01. 调试环境搭建（一）入门
02. 调试环境搭建（一）进阶
03. 集合（一）ArrayList
04. 集合（二）LinkedList
05. 集合（三）HashMap
06. 集合（四）LinkedHashMap
07. 集合（五）HashSet
08. 集合（六）TreeMap
09. 集合（七）TreeSet

... 目前 9+ 篇（努力更新中）

《Redis 源码解析》

01. Redis 调试环境搭建
02. Redisson 调试环境搭建
03. Redisson 限流器 RateLimiter
04. Redisson 可重入分布式锁 ReentrantLock
05. Redisson 可靠分布式锁 RedLock

... 目前 9+ 篇（随缘更新中）

最近更新《芋道 SpringBoot 2.X 入门》系列，已经 20 余篇，覆盖了 MyBatis、Redis、MongoDB、ES、分库分表、读写分离、SpringMVC、Webflux、权限、WebSocket、Dubbo、RabbitMQ、RocketMQ、Kafka、性能测试等等内容。

提供近 3W 行代码的 SpringBoot 示例，以及超 4W 行代码的电商微服务项目。

获取方式：点“**在看**”，关注公众号并回复 **666** 领取，更多内容陆续奉上。

如果你喜欢这篇文章，喜欢，转发。
生活很美好，明天见(。·ω·。)/♡

阅读原文

喜欢此内容的人还喜欢

SpringBoot 集成 WebSocket，实现后台向前端推送信息

芋道源码

五一火车票秒光！用这个办法还能“捡漏”！

人民网科普

朗诵：愿你用力爱过，不负此生

百草园书店