security 安全性

阻止恶意抢票

安全性：场景1：阻止恶意抢火车票行为

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 外部恶意软件 |
| 刺激 | 以用户身份持续购票或者快速购票 |
| 制品 | 购票模块 |
| 环境 | 系统运行时 |
| 响应 | 系统冻结此用户的账号，阻止其购票行为 |
| 响应度量 | 当恶意软件发出恶意请求时，要在1s内检测。  在1s内冻结其账户操作。 |

阻止黑客攻击

安全性：场景2：阻止黑客攻击

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 对系统的攻击行为 |
| 刺激 | 黑客对系统的数据攻击、DOS攻击等 |
| 制品 | 系统 |
| 环境 | 系统运行时 |
| 响应 | 系统检测到攻击，同时采取相应保护措施，避免攻击范围扩大，消除攻击带来的影响 |
| 响应度量 | 在1s之内检测到正在进行的攻击。  在1s内采取安全措施。  在1s内消除攻击带来的影响。 |

availability 可用性

应对系统错误

可用性：场景1：应对系统错误

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 系统内部的运行或者是用户的操作 |
| 刺激 | 大量用户的操作导致系统压力增大或者是系统自身运行时出现错误 |
| 制品 | 系统 |
| 环境 | 系统运行时 |
| 响应 | 如果压力过大则将压力转移到一些设备上。如果是出错，那么快速修复错误。 |
| 响应度量 | 诊断异常状况的时间不超过2s  将请求压力转移到其他设备上的时间不超过3s  快速修复错误的时间不超过2s  错误修复成功率不低于90% |

## 第N次迭代

#### 选择元素

第N次迭代选择的元素是服务器模块。

#### 选择ASR

第N次迭代选择的ASR是安全性、互操作性、可用性、易修改性。

1. 因为12306的用户群体非常广大，所以安全性尤为重要，如果12306的服务器模块被黑客成功攻击，那么会对整个系统都会是非常大的打击，而且可能引发更大的隐患。
2. 因为12306系统设计财务流通、身份识别，所以需要与银行系统与公安系统进行交互，用于财务转账、身份识别，所以互操作性对于12306系统来说很重要。
3. 因为12306的用户群体非常广大，所以系统要保持高可用性，这就需要服务器端一直要保持高可用的状态。
4. 因为12306业务逻辑可能会发生变化，，所以这使得在服务器端进行修改造成的对其他的模块的影响要尽可能小，所以12306的易修改性要尽可能好。

#### 候选策略表和决策

|  |  |  |  |
| --- | --- | --- | --- |
| 策略 | Pros | Cons | 是否采用 |
| 攻击发生时收回数据访问权限 | 在遭受攻击时，启动自我防御机制。极大增加了系统的安全性。 | 增加了失效时间，降低了系统的可用性。 | 采用。增加攻击侦测模块，当侦测到攻击发生时，服务器模块将会拒绝一切外界访问，直到确认安全。 |
| 数据加密 | 对数据加密，保证了数据的安全性。 | 对数据的加密需要消耗额外的资源。 | 采用。因为与银行系统的交互的信息可能被拦截和监听，所以加密是必须的。 |
| 用户认证 | 能够有效阻止一些外部的攻击，增加了安全性。 | 验证工作消耗额外资源。同时可能会有不短猜测以获取验证身份的行为。 | 不采用，因为此时处理的是服务器模块的安全性事务，用户认证会由客户端进行工作，这边再做就是重复工作。 |
| 限制访问 | 限制访问范围，增加了对敏感数据的保护，提高了安全性。 | 消耗大量额外的资源，同时也需要系统构建者额外的工作量。 | 采用。对于查询的用户，将其权限在只能查看自己想找的信息之内。对于其他操作的用户，同样如此。不透露多余的任何信息。 |
| 保证冗余的服务器和网络连接 | 保证了系统失效的时候能够立即采取措施， | 需要大量的计算资源作为候补。 | 采用。如果当前正在运行的服务器突然失效，则立即使用冗余的服务器和网络连接。 |
| ping/echo | 能随时对系统发起询问请求。 | 通讯量是heartbeat方案的双倍。 | 不采用。因为采用ping/echo的机制的检查消耗通讯量大。 |
| heartbeat | 定期检测系统是否已经失效。消耗资源较少。 | 在系统内部要实现额外的heartbeat相关逻辑。 | 采用。因为采用heartbeat的机制所进行的检查是定期的，可靠，而且消耗通讯量较少。 |
| monitor | 能即时监控整个系统，有利于立即应用系统失效的情况。 | monitor一直监测整个系统，消耗了大量的系统资源。 | 不采用。因为monitor机制占用了太多的系统资源。 |
| 服务定位（locate） | 采用目录命名机制，方便了与外界系统的交互行为。 | 维护服务目录需要大量的资源。 | 采用。在一个已知的目录服务里面，去定位自己想找到的服务，比如银行消费服务、公安人员监测服务。 |
| 裁剪接口 | 通过接口很好地限制地系统与外界的交互范围。 | 对接口的修改影响整个全局的功能。 | 不采用。12306系统的服务器端主要是作为服务的使用者而不是服务的提供者。 |
| 模块分解 | 将职责不同的部分分解出来，使得他们相对于其他部分独立，更加容易独立于其他的部分进行修改。 | 分解要消耗一定的成本，同时可能会增加复杂度。 | 采用。可以将服务器中不同逻辑部分拆分来进行分解。 |
| 延迟绑定 | 到最后才决定执行的动作。 | 越往后就越有多种可能性，会使得复杂度提高。使得成本变高，测试效率变低。 | 不采用。提高了复杂度。 |
| 重构（refactor） | 通过抽象出共同的service，可以降低内聚性，使得代码的效率更高。 | 成本较大。 | 不采用。成本太大。 |

#### 第N次迭代结果

如图所示，新增元素说明如下。

攻击侦测模块：侦测非正常访问，在攻击发生时禁止外界访问服务器模块。

数据加密模块：对服务器与外部系统交互时传送的数据进行加密和解密工作。

冗余的服务器与网络连接模块：作为候补，等待当前运行的服务器失效或者网络连接失效而发挥作用。

heartbeat模块：监测服务器模块是否失效，以尽快做出反应。

访问权限控制模块：监测用户的访问范围，使用户的访问范围始终被限制在必须需要访问的范围内。

服务定位模块：用于处理与外部系统的交互，将外部系统视为服务的提供者，从而简化系统的工作。

服务器端内部拆分为列车调度模块与用户票务模块。

## 第N次迭代

#### 选择元素

第N次迭代选择的元素是请求总线模块。

#### 选择ASR

第N次迭代选择的ASR是性能、可用性。

1.因为12306的用户群体非常广大，所以经常会出现高并发的情况，这时候要考查系统的性能，在压力情况下，系统要能够正常运行而不是崩溃，所以请求总线需要对于请求能够合理分派，保持负载均衡。

2.因为12306购票系统不能出现长时间失效的情况，而如何请求总线一旦失效，那其实整个系统也就因为这一个组件而失效。因此确保请求总线的可用性是非常重要的。

#### 候选策略表和决策

|  |  |  |  |
| --- | --- | --- | --- |
| 策略 | Pros | Cons | 是否采用 |
| 智能分派 | 能够识别出请求对应的具体broker，将其直接分派到对应的broker上，从而实现智能分派，避免造成单个broker的压力过大。 | 需要额外的工作量。 | 采用。 |
| 队列分类 | 对于user broker再构建一个user broker的请求队列，对于schedule broker构建一个schdule broker的请求队列。提高了效率。 | 耗费资源。 | 采用。 |
| 增加资源 | 提高了请求总线的效率，从而提高处理请求效率。 | 耗费资源。 | 采用。 |
| 限制执行时间。 | 给每一个请求的处理确定一个执行时间的上限，避免单个请求耗费时间过多，造成性能下降。 | 可能会错失一些请求造成一些请求没有回应。 | 不采用。错失请求的范围扩大会演变为失效情况。 |
| 请求总线冗余 | 防止正在运行的请求总线的失效情况。如果其失效，则用冗余的总线来代替它。 | 耗费资源。 | 采用。 |
| ping/echo | 能随时对系统发起询问请求。 | 通讯量是heartbeat方案的双倍。 | 不采用。 |
| heartbeat | 定期检测系统是否已经失效。消耗资源较少。 | 在系统内部要实现额外的heartbeat相关逻辑。 | 采用。在server对请求响应时，不发出heartbeat以免占用资源，在server不处理请求的空闲时刻，发出heartbeat来确保其没有失效。 |

#### 第N次迭代结果

如图所示，新增元素说明如下。

智能识别模块。用于识别正在处理的请求它对应的是user broker还是schedule broker，并把请求放入对应的broker的请求队列中。

user broker请求队列。用来给对于user broker的请求进行排队。

schedule broker请求队列。用来给对于schedule broker的请求进行排队。

总线冗余模块。提供冗余，防止当前总线失效。

heartbeat监测模块。用于检测系统是否失效，帮助其在失效情况下采取相应措施。