安全性需求

概述

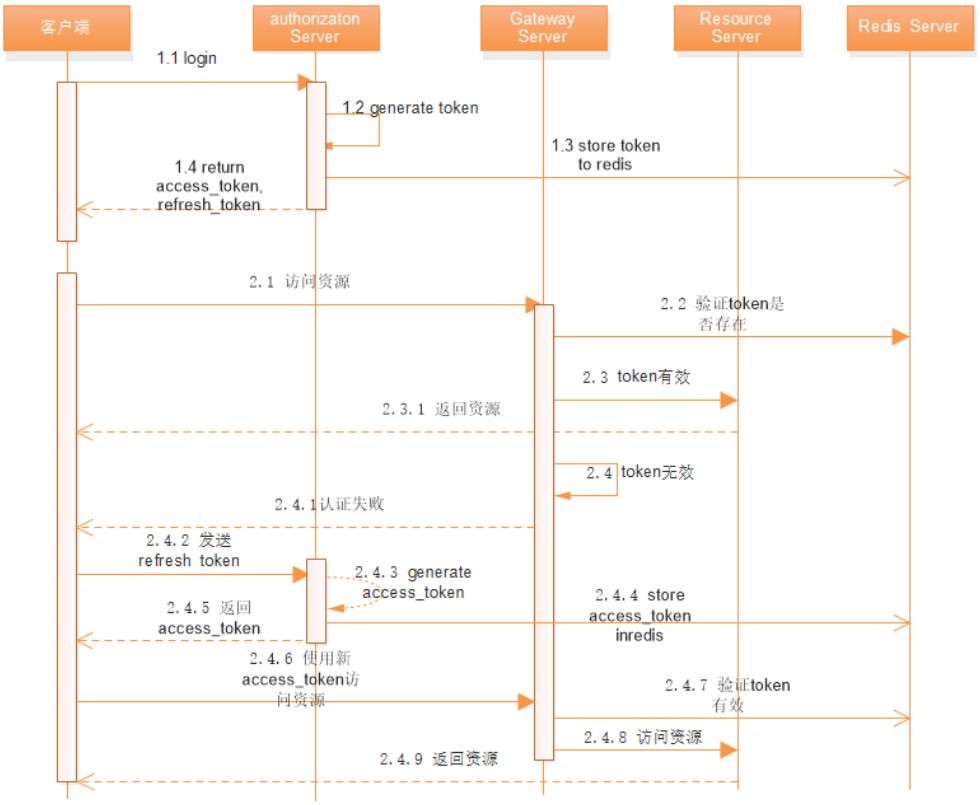
> 系统安全分为物理安全、网络安全、主机安全、应用安全、数据安全、管理安全等方面，本文档将逐一描述解决方案。

应用安全

应用层安全

> 第三代系统采用前后端分离的设计方式，前端采用了当前web流行的token认证方式，后台系统采用了无状态的应用设计。

token认证流程



1、token在redis中统一管理

2、token可配置失效时间

3、前端需要refresh token，增加安全性

前端安全

前端常见的安全性问题以及解决方案：

XSS（Cross-Site Scripting）脚本攻击漏洞；

- 将重要的cookie标记为http only, 这样的话Javascript 中的document.cookie语句就不能获取到cookie了.
- 只允许用户输入我们期望的数据。 例如: 年龄的textbox中，只允许用户输入数字。 而数字之外的字符都过滤掉。
- 对数据进行Html Encode 处理
- 过滤或移除特殊的Html标签， 例如: <script>, <iframe> , < for <, > for >, &quot for
- 过滤JavaScript 事件的标签。例如 "o nclick=", "onfocus" 等等。

CSRF（Cross-sit request forgery）漏洞；

- 增加token验证.因为cookie发送请求的时候会自动增加上，但是token却不会，这样就避免了攻击
- Referer验证。页面来源的判断

iframe安全隐患问题；

- 只允许安全的网站进行嵌入；

本地存储数据问题；

- local storge中存储的数据加密

第三方依赖的安全性问题；

- 严格管理第三方依赖，目前前端的第三方依赖为dev express，ng-zorro

HTTPS加密传输数据；

- 采用https协议传输，具体详情参见下面数据传输章节

> 上述前端问题均可在安全漏洞扫描中进行扫描。

后端安全

> 第三代系统后端采用openshift容器平台部署，应用部署在pod容器中，不提供对外访问的服务路由，只能通过api网关应用提供统一入口，增加了安全性。
>
> api网关统一对token进行验证。

API服务访问

> 用户访问数据都通过api服务的方式访问。api服务都采用https协议传输。
>
> 用户内网访问与外网访问都需要使用https协议。
>
> 外网访问需要购买CA认证的证书。

证书与服务器的一些配置要求如下：

- 服务器所有的连接使用TLS1.2以上的版本（openssl 1.0.2以上版本）；（TLS1.3对应opsnssl1.1.1版本）
- HTTPS证书必须使用SHA256以上哈希算法签名
- HTTPS证书必须使用RSA2048位或ECC256位以上公钥算法
- 使用前向加密技术

Nginx的配置范例：

```
  ssl on;
  ssl_certificate_key /etc/ssl/cert/;
  ssl_certificate /etc/ssl/cert/;

  ssl_ciphers "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-
RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:
ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-
AES256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA:
ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-GCM-SHA384:AES128-GCM-
SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:DES-CBC3-SHA:
HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";

  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_session_cache shared:SSL:10m;

  ssl_stapling on;
  ssl_stapling_verify on;
  resolver 8.8.4.4 8.8.8.8 valid=300s;
  resolver_timeout 10s;

  ssl_prefer_server_ciphers on;
  ssl_dhparam /etc/ssl/certs/dhparam.pem;

  add_header Strict-Transport-Security max-age=63072000;
  add_header X-Frame-Options DENY;
  add_header X-Content-Type-Options nosniff;
```

配置文件管理

> 第三代系统采用了配置管理服务器进行配置文件的管理。同时将配置文件放在git中做版本管理。
>
> 配置管理服务器与git服务器都需要用户授权的访问。
>
> 配置文件中的关键信息可以在配置管理服务器中进行加密处理，加密方式可采用对称加密或者非对称加密。

数据库安全

原则：

- 满足安全审计要求，实现对数据库的全面安全防护和安全审计
- 防范外部数据库安全风险，减少或避免数据安全事件的发生和影响
- 防范内部数据库安全风险，减少或避免数据安全事件的发生和影响
- 减少不必要的应用人员对数据库的直接访问

技术目标：

- 数据库操作实时监控和审计，验证客户端对数据库的所有数据操作
- 安全预警、监测并告警数据泄露，监测并告警异常查询
- 通过部署专用的数据库防火墙，实现对数据库的安全保护要求。
- 高危操作阻断，处理注入攻击等。补充有特征库
- 事后调查取证，注入攻击调查，数据泄露事件调查

实现方式：

- 部署上线前要进行相关的安全测试
- 严格管理数据库用户权限的分配，及时回收敏感权限
- 为系统提供定期补丁
- 对敏感数据进行加密，将秘钥存储在特定硬盘并严格管理
- 提高开发人员安全意识，减少可能会影响数据库的任何威胁
- 数据库敏感或危险操作需在多人监督下执行
- 如果需要，则可以采用数据库端成熟的数据库防火墙产品
- 系统发布，系统配置项等初始化，通过系统做；减少应用人员对数据库的直接访问。
- GDPR的被遗忘权，应该有程序专门处理这种需求；还要遵守监管要求，数据不能简单删掉，一般都是从系统中转移到另外的系统中存储

应用系统安全设置

> 第三代系统设计了较为完善的安全管理配置，以及审计信息。具体详见需求文档【安全策略管理】

安全配置项

可配置管理项	英文	默认值	描述
强密码策略		强	必须为强密码。 要求必须包含大、小写字母、特殊字符及数字。例如Zxcv1234@,禁止使用连续小写字母和数字例如abc123
密码最小长度	Minimum password length	6	密码数据库实际存储是是6-20位，所以保存时需要校验最小和最大长度是否满足6-20，且最大长度需要大于等于最小长度
密码最大长度	Maximum password length	20	
最多不登录天数	No login days at most	90	超过时间未登录，用户状态改为停用
最长不操作时间（分）	Maximum no operation time (min)	10	大于等于10；超过则自动logout
密码最多使用天数	Passwords are used at most	90	大于等于90；不修改密码则用户状态改为停用
新旧密码重复	New and old password repetitions	1	大于等于1，小于等于5 例：如果维护1，那么就是新密码与上1次的密码不能一致。
密码过期前提醒	Password expiration reminder	Y	Y-是-Yes N-否-No
超期未登录停用前提醒	Account expiration reminder	Y	Y-是-Yes N-否-No
用户超期未登录停用与密码过期提醒天数 和 次数	Reminder frequency _days _times a day	默认7天，默认每天1次	天数：大于等于7天， 次数大于等于每天1次
启用密码错误锁定	Password error lock required	默认选中	只读，勾选
用户登录时，若连续错误密码_次，账号将被锁定_分钟后自动解锁，手动也可解锁	When the user logs in, if the continuous error password is_times, the account will be locked for_ minutes automatically, and the manual can also be unlocked	默认5次，10分钟	需要大于等于5次，大于等于10分钟
密码重置方式：	Password reset mode：		
通过邮箱验证	Email	默认选中	只读
启用登录验证码	Login verification code	默认选中	只读
用户登录时，连续输错密码，第_次，则开启登录验证码	When the user logs in, he continuously sends the wrong password _times, then must enter the login verification code.	3	默认3次，只读
客户端登录选项	Login options		

禁止同一账号多地同时登录	Prohibit multiple simultaneous login of the same account	默认空	勾选，可空。 如果禁止同时登录，则相同用户名，只能在一台电脑登录，后登陆的人会踢掉前面的用户；（防止密码泄露后无法登录）

系统中保存的密码等数据

> 系统中保存的程序连接的密码，如：ftp、连接串等，这类数据可以根据需求进行加密存储，以及掩码展示。

审计信息

第三代系统中提供了以下审计信息：

- [x] 数据审计

  基于数据变动的审计信息，将核心数据的增删改都记录下来，并提供查询页面；

- [x] 系统日志审计

  将系统运行的日志信息统一存储，并提供在线查询的功能；

- [x] 用户访问审计

  将所有用户的登录以及访问页面，api服务等的数据记录下来，并提供查询页面；

- [x] 特殊审计需求

  按照用户的需求，对角色权限的变动，提供格式化的审计数据；

- [x] 数据库审计（应用端）

  采用带有审计功能的数据库连接池组件（druid），开启sql审计，防火墙等功能，基于SQL语义分析来实现防御SQL注入攻击，可以拦截不被允许执行的sql语句，以及慢sql的监控等；

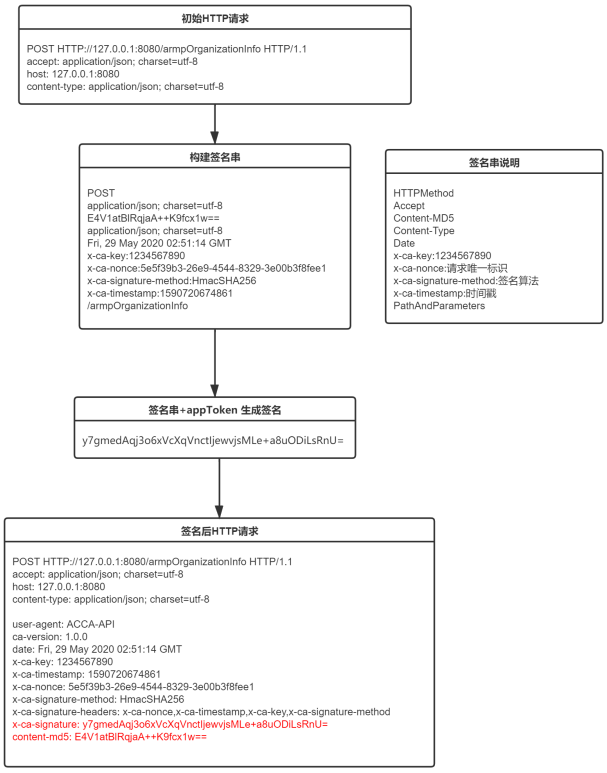
对外提供服务

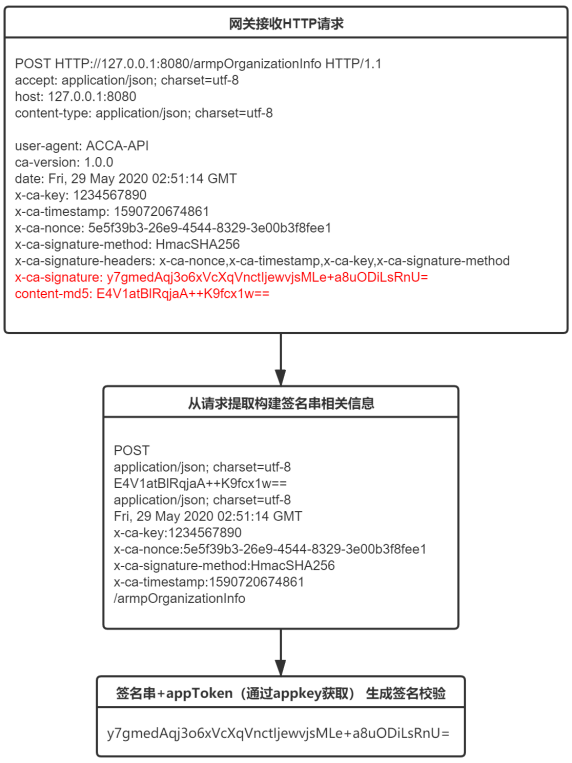
> 第三代系统会对外部系统暴露服务，可以从公网调用第三代系统的api，同时内部系统也需要调用第三代系统的api。
>
> 外网的调用采用最严格的api管理方法。详细参见对外网关服务

对外部的调用管理，采用ACCA提供SDK的方式，外部系统通过ACCA提供的SDK，发送请求，SDK中提供签名组件，将外部请求通过封装appKey及其它请求参数，用appToken进行HmacSHA256签名，访问ACCA服务。

ACCA网关收到**【对外域名】**的请求后

1. 判断该api是否存在及已上线
2. IP控制判断【可选】
3. 流量控制判断【可选】
4. 根据请求参数对请求进行签名校验
5. Token校验【可选】
6. 校验成功则正常访问后端请求





操作系统高权限用户相关

应用系统安装过程参见 运维文档。

系统安装过程中，除个别三方应用外，大部分不需要高权限用户。

系统在安装完成后，运行时都不需要高权限用户。

应用运行

应用系统无单点。参见应用架构 文档。保证运行时的安全。

代码安全

代码管理

采用产品研发部现有的管理流程。

静态代码中漏洞扫描

采用sonar进行静态代码漏洞扫描。

sonar中规则选用：

规则来源	规则数量
ACCA（代码规范）	1846
FindBugs	443
FindBugs + FB-Contrib	745
FindBugs Security Audit	121
FindBugs Security Minimal	91
Sonar way（soanr默认）	1321

例如，sql注入风险之类的都会被扫描到。

公司级漏洞扫描

满足公司漏洞扫描要求，并按照扫描结果进行漏洞的修改。

数据安全


第三代客运收入管理平...t+0.3.docx