**Pull Bears技术文档**

－**关于客户端与后台通信安全的建议**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 更新日期 | 更新版本 | 更新人 | 更新内容 | 审批人 |
| 2015/7/29 | 1.0.0 | 胡家驹 | 新建 |  |
|  |  |  |  |  |

# 文档说明

## 1．文档编写目的：

本文档编写的目的是为客户端与后台网络通信安全考虑及解决方案而编写。

## 2．文档适用范围：

本文档描述的内容适用于开发人员，也部分适用于本项目－《共时财经》中的技术人员、设计人员。本文档面向的读者为具有一定（智能移动终端）开发经验的人员，需要掌握一定量的基础开发语言，熟悉HTTP协议，SOAP协议，XML/JSON序列化、解析等常见互联网技术，掌握基于SSL的互联网安全传输（eg.RSA AES）方面知识。

## 3．文档实现目的作用：

为解决客户端与后台网络通信安全提供几点建议。如大家觉得有必要可讨论。

## 4．文档附加：

由于本人对PHP、java等后台有关编程规则不懂，如有错误、遗漏欢迎指正。本文档主要从前端角度考虑。

# 文档内容

## 概述：

目前前端与后台基于HTTP协议通信采取了HTTPS协议，这在前端通信机制中是比较安全的一个做法，相比与其他非金融类app采取此法而言金融类app采取HTTPS协议自然而然（废话）。但在传输过程中并未使用证书验证链接安全性，~~所以数据还是明文传输的~~（即使没有证书数据也可以实现加密，也可以直接访问HTTPS链接，此时代理抓包为不可读字符串，针对了解系统类实现原理者可能为明文，因为没有验证证书是否合法，所以无法绕过中间人伪造数据），我们的手机很多时候使用公用wifi上网，这时我们的数据是暴漏了，这时的数据传输是非常危险的，用户的隐私数据很容易被第三方获取到，客户端中使用https能有效的防止数据暴漏，防止第三方截获应用的通信数据。应用程序即使启用了HTTPS链接，但是尚未启用SSL 绑定，就可能会被中间人攻击，添加SSL证书绑定有助于避免应用程序遭受中间人攻击或其他安全漏洞。处理敏感的客户数据或财务信息的应用程序强烈建议启用SSL绑定并使用 HTTPS 连接通信。

如前端发起GET请求，如已登录则设置头部authorization（登录时返回的TOKEN，目前该TOKEN无时效性），如未登录则忽略此设置。饭后发起请求pull到数据。返回数据不做任何处理。

如前端发起POST请求，除了采取GET方式的设置外，如果是登录或注册接口则增加了设置了HTTPHeader的bear字段（注意这个字段是一个固定值X），进一步提高了安全性。返回数据不做任何处理。

## Discussion：

A．链接加密：  
 在客户端发起请求时，首先在HTTP握手时就建立加密链接，防止窃取数据，这个安全是基于协议层的（暂且这么称呼吧，其实是在传输层加密），那么了理论上在此链接有效时传输明文也是安全的。此方法就会用到CA证书验证、部署，CA较贵，好在公司已有CA证书，只需部署。

本步骤实现要点：

a）服务器部署TLS1.0以上协议，推荐TLS1.2或SSL3.0以上一些。（CA证书）

b）客户端需要持有CA公钥采取DER或CER编码方式的证书文件，客户端（iOS）如若使用AF框架强烈建议更新到2.5.3及以上版本。

B．数据加密：  
 在客户端发起请求时，针对数据采取AESA（我自己取得名字，其实是AES＋RSA）加密方式（AESA说明：随机生成AES的秘钥X，用X对明文采取AES加密得到内容W，对X进行RSA加密得到P，传输加密后的W和P），因为SSL的RSA方式不适合大数据的加密，RSA比较慢但安全性高（可自行查阅SSL安全相关资料，本文档如若没有特殊说明RSA默认1024位）。AES（可自行查阅AES安全相关资料，本文档如若没有特殊说明AES默认采取CBC模式）对称加密适合大数据（客户端有限）的加解密，AES处理速度快但安全性低。综合两者的优点，所以采取AESA方式加密数据。

本步骤实现要点：

a）自签名RSA证书私钥部署在服务器，公钥分发在客户端。

b）AES采取CBC模式，所以需要分组向量，可协商。

C．Token：  
 目前登录注册中POST头部bear固定值就是提高了安全性，但是android的高风险破解率导致了数据的不安全性。Token也是提高了安全性，但是缺少时效性，后台需维护userid与Token的映射关系。为了Token的安全性，那么Token最好不在网络中传输（详情可参阅相关博客：URL签名http://blog.csdn.net/newjueqi/article/details/44154791），如若Token失效则需要用户重新登录或解锁授权，不再叙述Token有关。

本步骤实现要点：

a）Token不在网络传输。

D．Authorization：

PHP、Java不太懂，我的理解是接口认证在有些应用场景是必须的，如第三方登录（Sina授权、QQ授权）等场景。Token作为加密数据之一传输也可实现。不再叙述。

E．返回数据：

目前返回数据未做任何处理，客户端pull到数据直接使用。链接已经加密，返回数据可以做RSA签名，防止中间人篡改数据、抵赖，确保源数据是从我们服务器发送。

F．客户端加密方式：

目前客户端用到的有MD5，Base64等方法，严格来讲这不是加密算法，只是哈希算法（Message Digest），安全性较低。如果使用这些Hash算法可考虑加盐方式。

G．防止重新打包签名：

这个方式我们之前没有采取过，因为下载渠道比较多（主要是android端），所以APK文件可能会被重新二次打包，在发起请求时取到APK签名数据，进行哈希算法运算，以字段键值对形式拼接到表单中，在接口中验证是否为正版APP发送的请求。

本步骤实现的要点：（需要测试）

a）版本发布时需后台记录APK签名数据，校验放式需协商。

H．敏感逻辑保护：

针对程序中敏感逻辑可采取C与OC混编，C与Java混编（需测试），可采取执行脚本进行代码混淆防止hacker进行class－dump推理分析。

## Conflict：

A．客户端网络变化、IP变化要求用户重新登录？！  
 B。GET数据时同一IP毫秒级（可讨论）访问时需要限制，读数据本身木有问题，防止恶意“攻击”浪费服务器性能？！

# 文档声明：

再次感谢您阅读本文档，由于本人水平有限如有错误、遗漏欢迎指正。