

CoderZYWang

所有努力的岁月，都不会被辜负，最差的结果，也不过是大器晚成！

目录视图

摘要视图

RSS 订阅

个人资料



王王王王王中尧

访问：128746次

积分：3955

等级：BLOG 5

排名：第5819名

原创：212篇 转载：50篇

译文：0篇 评论：12条

文章搜索

文章分类

C (26)

OC (56)

iOS\_Uikit (45)

Xcode (10)

iOS (67)

Chicken Soup (3)

iOS\_multithreading (10)

Network (11)

Unix (1)

iOS\_thirdPartyFramework (3)

Mac (2)

Swift 2.3 (4)

Swift 3.0 (4)

CocoaPods (2)

【1024程序员节】参加活动领暖心礼品

【观点】有了深度学习，你还学传统机器学习算法么？

【知识库】深度学习知识图谱上线啦

Network 【HTTPS请求/AFN】

标签：HTTPS请求 AFN

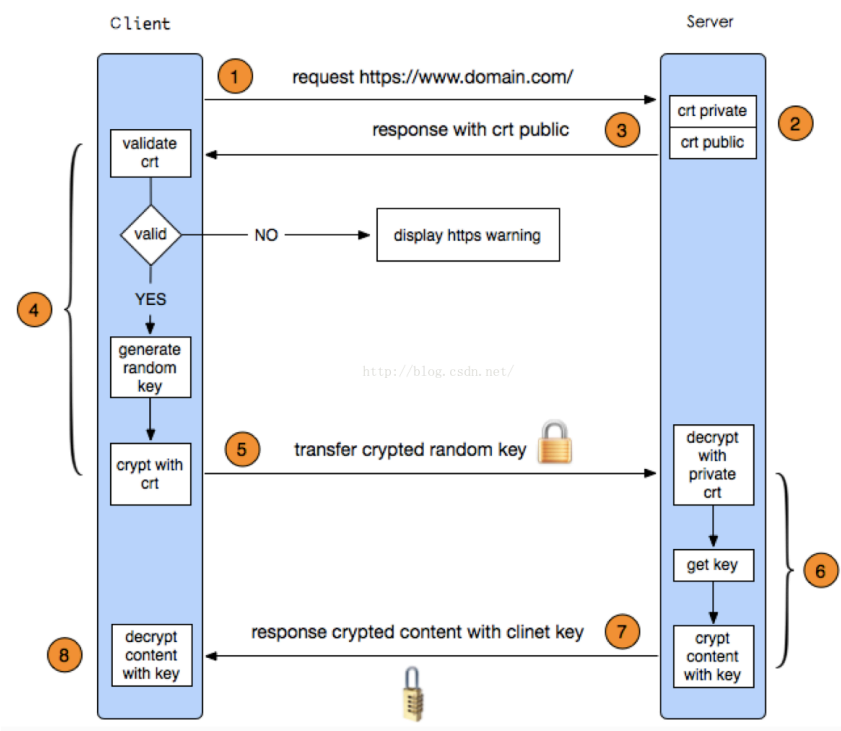
2016-07-20 14:37 183人阅读 评论(0) 收藏 举报

分类：Network (10)

版权声明：本文为博主原创文章，未经博主允许不得转载。

HTTPS请求过程

如下图：



```
graph LR
    subgraph Client
        direction TB
        C1[request https://www.domain.com/] -- 1 --> C2[validate crt]
        C2 -- NO --> C2W[display https warning]
        C2 -- YES --> C3[generate random key]
        C3 --> C4[encrypt with crt]
        C4 -- 5 --> C5[transfer encrypted random key]
        C5 --> C6[decrypt content with key]
    end
    subgraph Server
        direction TB
        S1[crt private] -- 2 --> S2[crt public]
        S2 -- 3 --> S3[response with crt public]
        S3 --> C2
        S4[decrypt with private crt] -- 6 --> S5[get key]
        S5 --> S6[encrypt content with key]
        S6 -- 7 --> S7[response encrypted content with client key]
        S7 -- 8 --> C6
    end
    C1 -- 1 --> S1
    S2 -- 2 --> S2
    S3 -- 3 --> C2
    C4 -- 5 --> S4
    S6 -- 7 --> C6
```

① 客户端输入网址https://www.domain..com，连接到server的443端口。

② 服务器返回一个证书（包含公钥、和证书信息，如证书的颁发机构，过期时间等），证书由服务器所拥有的私钥非对称加密生成。

③ 客户端对证书进行验证（首先会验证证书是否有效，比如颁发机构，过期时间等等）。

④ 如果客户端验证通过，客户端生成一个随机数，在用服务器返回的证书（公钥）进行加密传输。

⑤ 因为公钥是通过服务器的私钥生成，所以服务器是可以对客户端的传回的加密数据进行对称解密的。服务

iOS程序启动到运转 (0)

Python (1)

## 文章存档

2016年10月 (2)

2016年09月 (3)

2016年08月 (9)

2016年07月 (31)

2016年06月 (19)

展开

## 阅读排行

Network 【TCP/IP 四层

(12341)

iOS 【UIKit-UIPageCont

(2799)

Objective-C 【电商APP

(2707)

Objective-C 【在手动内

(2622)

12天学好C语言——记录

(2482)

iOS 【在iOS中自动检测

(1876)

Objective-C 【NSDiction

(1774)

Objective-C 【完整OC项

(1654)

Objective-C 【NSMutabl

(1650)

Objective-C 【NSArray

(1629)

## 最新评论

Swift 3.0 【Swift 3.0 环境下使用

王王王王王中尧:

@wangwei20151031:人家就是这样定义的啊。下面是我直接copy的AFN的写法: open...

Swift 3.0 【Swift 3.0 环境下使用

wangwei20151031: 你好, 为什么在swift3.0中调用

AFNetworking的POST和GET方法必须用小写? 谢谢

iOS 【UIKit-导航控制器 UINavigationController

sinat\_34781351: mark

Objective-C 【NSArray 的基本使用

ITjungle: 非常喜欢你的blog, 喜欢可以交流一下学习心得

Objective-C 【NSDictionary 概念

ITjungle: 好!

Objective-C 【NSMutableDictionary

ITjungle: 你的博客很精致, 受用

Objective-C 【电商APP应用代码

王王王王王中尧: @u010850027: 可以的!

Objective-C 【电商APP应用代码

丁国华: 青春跟年龄没有任何关系, 有的人在16、20岁的时候, 青春已经死亡, 他对生命已经没有任何期待。有的人到...

Objective-C 【在手动内存管理中

王王王王王中尧: @u010786678: 可以的~

Objective-C 【在手动内存管理中

傻丫头与科技: 青春跟年龄没有任何关系, 有的人在16、20岁的时候, 青春已经死亡, 他对生命已经没有任何期待。有的人到...

器拿到由客户端生成的随机数, 对要传递的数据使用随机数加密。

⑥ 客户端收到服务器使用随机数加密的数据进行解密。

苹果已经封装了HTTPS连接的建立、数据的加密解密功能, 我们直接可以访问https网站的, 但苹果并没有验证证书是否合法, 无法避免中间人攻击。要做到真正安全通讯, 需要我们手动去验证服务端返回的证书。

AFNetwork中的AFSecurityPolicy模块主要是用来验证HTTPS请求时证书是否正确。AFSecurityPolicy封装了证书验证的过程, 让用户可以轻易使用, 除了去系统信任CA机构列表验证, 还支持SSL Pinning方式的验证。

请求 信任的HTTPS 和 不信任的HTTPS 代码如下:

```
[objc]
01. //
02. // ViewController.m
03. // 0720-01HTTPS请求 (AFN) -01
04. //
05. // Created by 王中尧 on 16/7/20.
06. // Copyright © 2016年 wzy. All rights reserved.
07. //
08.
09. /*
10. 使用AFN HTTPS 请求
11. 如果发送的请求是HTTPS的请求,那么有两种情况
12. ----- 001 请求的网站所使用的证书是CA签名&加密的方式满足ATS的要求 那么我们在代码中什么都不需要做(比如
    说github官网, 就是信任的https)
13. 不管是信任, 还是不信任, 都要写: (设置反序列化方法)
14. // 设置反序列化方法
15. manager.responseSerializer = [AFHTTPResponseSerializer serializer];
16.
17. ----- 002 请求的网站所使用的证书是自签名,需要做如下操作: (比如说12306, 就是不信任的https)
18. 1) 禁用ATS特性
19. // 设置允许不受信任的证书 (该参数默认为no)
20. securityPolicy.allowInvalidCertificates = YES;
21. // 设置不验证域名 (该参数默认为yes)
22. securityPolicy.validatesDomainName = NO;
23.
24. 2) 需要在代码中信任证书(安装)
25. 在Info.plist 里面设置上两行属性(添加HTTP)
26.
27. 2017年1月1日 就不允许使用HTTP请求了
28. */
29.
30. #import "ViewController.h"
31. #import "AFNetworking.h"
32.
33. @interface ViewController ()
34.
35. @end
36.
37. @implementation ViewController
38.
39. - (void)touchesBegan:(NSSet<UITouch * >)touches withEvent:(UIEvent *)event {
40.     [self believe];
41.     // [self unBelieve];
42. }
43.
44. // 信任的HTTPS 百度
45. - (void)believe {
46.     // 创建会话管理者
47.     AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];
48.
49.     // 设置反序列化方法
50.     manager.responseSerializer = [AFHTTPResponseSerializer serializer];
51.
52.     [manager GET:@"https://www.baidu.com" parameters:nil success:^(NSURLSessionDataTask *task,
53.         NSLog(@"success---
54. %@", [[NSString alloc] initWithData:responseObject encoding:NSUTF8StringEncoding));
55.     } failure:^(NSURLSessionDataTask *task, NSError *error) {
56.         NSLog(@"failure---%@", error);
57.     }];
58. }
```



收藏到代码笔记



```

56.     }];
57. }
58.
59. // 不信任的HTTPS 12306官网
60. - (void)unBelieve {
61.     // 创建会话管理者
62.     AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];
63.
64.     // 设置反序列化方法
65.     manager.responseSerializer = [AFHTTPResponseSerializer serializer];
66.
67.     /**
68.      typedef NS_ENUM(NSUInteger, AFSSLPinningMode) {
69.      AFSSLPinningModeNone, // 这个模式表示不做SSL pinning, 只跟浏览器一样在系统的信任机构列表里验证
        服务端返回的证书。若证书是信任机构签发的就会通过, 若是自己服务器生成的证书, 这里是不会通过的
70.      AFSSLPinningModePublicKey, // 这个模式同样是用证书绑定方式验证, 客户端要有服务端的证书拷贝, 只
        是验证时只验证证书里的公钥, 不验证证书的有效期等信息
71.      AFSSLPinningModeCertificate, // 这个模式表示用证书绑定方式验证证书, 需要客户端保存有服务端的证
        书拷贝, 这里验证分两步, 第一步验证证书的域名/有效期等信息, 第二步是对比服务端返回的证书跟客户端返回的是
        否一致
72.      };
73.     */
74.     // AFNetwork中的AFSecurityPolicy模块主要是用来验证HTTPS请求时证书是否正确
75.     // 验证证书的模式
76.     AFSecurityPolicy *securityPolicy = [AFSecurityPolicy policyWithPinningMode:AFSSLPinningModeNone];
77.
78.     /**
79.      Whether or not to trust servers with an invalid or expired SSL certificates. Defaults to YES
80.      */
81.     // 设置允许不受信任的证书
82.     securityPolicy.allowInvalidCertificates = YES;
83.
84.     /**
85.      Whether or not to validate the domain name in the certificate's CN field. Defaults to YES
86.      */
87.     // 设置不验证域名
88.     securityPolicy.validatesDomainName = NO;
89.
90.     manager.securityPolicy = securityPolicy;
91.
92.     [manager GET:@"https://kyfw.12306.cn/otn" parameters:nil success:^(NSURLSessionDataTask *task,
93.      NSData *data) {
94.         NSLog(@"success---%@", [NSString alloc] initWithData:responseObject encoding:NSUTF8StringEncoding)];
95.     } failure:^(NSURLSessionDataTask *task, NSError *error) {
96.         NSLog(@"failure---%@", error);
97.     }];
98. }
99. @end

```

顶 2      踩 0

上一篇 [iOS 【使用Xcode和Instruments调试解决iOS内存泄露】](#)

下一篇 [Network 【SVN Mac端虚拟机端搭建步骤】](#)

我的同类文章

Network (10)

• Network 【TCP/IP 四层模... 2016-08-06 阅读 12341

• Network 【SVN 命令行操... 2016-07-21 阅读 125

• Network 【用 AFN 通过 P... 2016-03-14 阅读 316

• Network 【OAuth授权步骤 ... 2016-02-19 阅读 265

• Network 【HTTP通过程/... 2016-01-30 阅读 703

• Network 【CornerStone托... 2016-07-22 阅读 92

• Network 【SVN Mac端虚... 2016-07-21 阅读 100

• Network 【OAuth 认证流程... 2016-02-19 阅读 173

• Network 【Charles 抓包 (... 2016-01-31 阅读 788

• Network 【简述GET/POST... 2016-01-29 阅读 660

猜你在找

全网服务器数据备份解决方案案例实践

AFN框架 之同时兼容Http和Https通讯 配置

iOS8开发视频教程Swift语言版-Part 10:iOS的数据基

IOS平台ASI框架 和 AFN框架 之Https通讯

iOS开发高级专题—数据存储

AFN之HTTPS

iOS8开发视频教程-Part 4:iOS数据源协议、委托协议

tomcat7的https配置及ext.js在IE中的undefined错

iOS进阶课程-苹果的WebService

iOS AFN监听网络封装网络请求 HERO博客

广告

阿里大鱼

阿里巴巴集团旗下

三网合一短信通道

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap