iOS - 网络数据安全加密(MD5)

原创

极客学伟 于 2015-08-05 21:25:57 发布 810 收藏
分类专栏： ❶ iOS开发 ❻ 网络 文章标签： 数据安全 ios 网络

本文链接：https://blog.csdn.net/qxuewei/article/details/47304121


❶ iOS开发 同时被 2 个专栏收录
257 篇文章 2 订阅
订阅专栏


❻ 网络
30 篇文章 0 订阅
订阅专栏

提交用户的隐私数据

一定要使用POST请求提交用户的隐私数据
GET请求的所有参数都直接暴露在URL中
请求的URL一般会记录在服务器的访问日志中
服务器的访问日志是黑客攻击的重点对象之一

用户的隐私数据
登录密码
银行账号
… …

数据安全

仅仅用POST请求提交用户的隐私数据，还是不能完全解决安全问题
可以利用软件（比如Charles）设置代理服务器，拦截查看手机的请求数据
因此：提交用户的隐私数据时，一定不要明文提交，要加密处理后再提交

常见的加密算法

MD5 \ SHA \ DES \ 3DES \ RC2和RC4 \ RSA \ IDEA \ DSA \ AES
1
加密算法的选择
一般公司都会有一套自己的加密方案，按照公司接口文档的规定去加密

MD5加密

什么是MD5
全称是Message Digest Algorithm 5，译为“消息摘要算法第5版”
效果：对输入信息生成唯一的128位散列值（32个字符）

MD5的特点
输入两个不同的明文不会得到相同的输出值
根据输出值，不能得到原始的明文，即其过程不可逆

MD5的应用
由于MD5加密算法具有较好的安全性，而且免费，因此该加密算法被广泛使用
主要运用在数字签名、文件完整性验证以及口令加密等方面

MD5解密网站：http://www.cmd5.com

MD5改进

现在的MD5已不再是绝对安全，对此，可以对MD5稍作改进，以增加解密的难度
加盐（Salt）：在明文的固定位置插入随机串，然后再进行MD5
先加密，后乱序：先对明文进行MD5，然后对加密得到的MD5串的字符进行乱序
… …
总之宗旨就是：黑客就算攻破了数据库，也无法解密出正确的明文

网络数据加密方案

1> 加密对象：隐私数据，比如密码、银行信息
2> 加密方案
* 提交隐私数据，必须用POST请求
* 使用加密算法对隐私数据进行加密，比如MD5
3> 加密增强：为了加大破解的难度
* 对明文进行2次MD5 ： MD5(MD5(pass))*先对明文撒盐，再进行MD5：MD5(pass.$salt)

2.本地存储加密
1> 加密对象：重要的数据，比如游戏数据

3.代码安全问题
1> 现在已经有工具和技术能反编译出源代码：逆向工程
* 反编译出来的都是纯C语言的，可读性不高
* 最起码能知道源代码里面用的是哪些框架

2> 参考书籍：《iOS逆向工程》

3> 解决方案：发布之前对代码进行混淆

```
* 混淆之前

@interface HMPerson :NSObject
- (void)run;
- (void)eat;
@end

混淆之后
@interface A :NSObject
- (void)a;
- (void)b;
@end

MD5加密实例

导入加密文件
```

```objc
#import "ViewController.h"
#import "MBProgressHUD.h"
#import "NSString+Hash.h"

@interface ViewController ()
@property (weak, nonatomic) IBOutlet UITextField *username;
@property (weak, nonatomic) IBOutlet UITextField *pwd;
- (IBAction)login;
@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    [self.view endEditing:YES];
}

- (IBAction)login {
    // 1.用户名
    NSString *usernameText = self.username.text;
    if (usernameText.length == 0) {
        [MBProgressHUD showError:@"请输入用户名"];
        return;
    }

    // 2.密码
    NSString *pwdText = self.pwd.text;
    if (pwdText.length == 0) {
        [MBProgressHUD showError:@"请输入密码"];
        return;
    }

    // 增加蒙板
    [MBProgressHUD showMessage:@"正在拼命登录中...."];

    // 3.发送用户名和密码给服务器(走HTTP协议)
    // 创建一个URL ： 请求路径
    NSURL *url = [NSURL URLWithString:@"http://218.83.161.124:8080/job/login"];

    // 创建一个请求
    NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:url];

    // 5秒后算请求超时（默认60s超时）
    request.timeoutInterval = 15;

    request.HTTPMethod = @"POST";

#warning 对pwdText进行加密
    pwdText = [self MD5Reorder:pwdText];

    // 设置请求体
    NSString *param = [NSString stringWithFormat:@"username=%@&pwd=%@", usernameText, pwdText];

    NSLog(@"%@", param);
```

```
    // NSString --> NSData
    request.HTTPBody = [param dataUsingEncoding:NSUTF8StringEncoding];

    // 设置请求头信息
    [request setValue:@"iPhone 6" forHTTPHeaderField:@"User-Agent"];

    // 发送一个同步请求(在主线程发送请求)
    // queue : 存放completionHandler这个任务
    NSOperationQueue *queue = [NSOperationQueue mainQueue];
    [NSURLConnection sendAsynchronousRequest:request queue:queue completionHandler:
     ^(NSURLResponse *response, NSData *data, NSError *connectionError) {
         // 隐藏蒙板
         [MBProgressHUD hideHUD];

        // 这个block会在请求完毕的时候自动调用
        if (connectionError || data == nil) { // 一般请求超时就会来到这
            [MBProgressHUD showError:@"请求失败"];
            return;
        }

        // 解析服务器返回的JSON数据
        NSDictionary *dict = [NSJSONSerialization JSONObjectWithData:data options:NSJSONReadingMutableL
        NSString *error = dict[@"error"];
        if (error) {
            [MBProgressHUD showError:error];
        } else {
            NSString *success = dict[@"success"];
            [MBProgressHUD showSuccess:success];
        }
     }];
}

/**
 *  MD5($pass.$salt)
 *
 *  @param text 明文
 *
 *  @return 加密后的密文
 */
- (NSString *)MD5Salt:(NSString *)text
{
    // 撒盐：随机地往明文中插入任意字符串
    NSString *salt = [text stringByAppendingString:@"aaa"];
    return [salt md5String];
}

/**
 *  MD5(MD5($pass))
 *
 *  @param text 明文
 *
 *  @return 加密后的密文
 */
- (NSString *)doubleMD5:(NSString *)text
{
    return [[text md5String] md5String];
}
```

```objc
/**
 *  先加密，后乱序
 *
 *  @param text 明文
 *
 *  @return 加密后的密文
 */
- (NSString *)MD5Reorder:(NSString *)text
{
    NSString *pwd = [text md5String];

    // 加密后pwd == 3f853778a951fd2cdf34dfd16504c5d8
    NSString *prefix = [pwd substringFromIndex:2];
    NSString *subfix = [pwd substringToIndex:2];

    // 乱序后 result == 853778a951fd2cdf34dfd16504c5d83f
    NSString *result = [prefix stringByAppendingString:subfix];

    NSLog(@"\ntext=%@\npwd=%@\nresult=%@", text, pwd, result);

    return result;
}
@end
```