联系方式(更新时间：2018-11-3)

- 手机：188740548 （长沙号码）
- Email：zhang_kn@icloud.com
- 微信号：188...

个人信息

- 张坤楠/男/1992
- 怀化学院 本科 信息与计算科学 2010.09-2014.06
- 工作年限：5年
- 技术博客：https://zhangkn.github.io/tags/ 、https://kunnan.github.io/tags/#security
- Github：https://github.com/zhangkn
- 户籍：泉州

求职意向

- 求职状态:在职，寻新工作
- 工作性质：全职
- 期望城市：泉州、厦门
- 期望职业：iOS&macOS应用逆向与安全工程师，高级iOS工程师
- 期望薪资：税前月薪30k~50k,特别喜欢的公司可例外

工作经验

2017年07月-至今

湖南微流网络科技有限公司

> - 主要负责逆向分析，开发iOS加固产品；越狱环境下Tweak、守护进程的开发与维护，非越狱环境下dylib注入重签；负责aso 项目。逆向系统框架，逆向系统应用。开发iOS逛有钱app、阿布云隧道 app 。
> - iOS&macOS应用逆向与安全工程师,高级iOS工程师:
>   - 正向开发： iOS逛有钱app、阿布云隧道 app 、疯狂笔划微信小程序。
>   - 逆向开发：利用Hopper、class-dump、ios-ssl-kill-switch、Keychain-Dumper、KNParseClangLib(MachOParser)进行静态分析；使用CycriptTricks（Powerful_private_methods）、hookClass_knhook_hookClassLog、UIButton_sendActionsForControlEvents、DerekSelander_LLDB(Python scripts to aid in your debugging sessions)、install_frida_on_device_and_mac(codeshare.frida.re) 进行动态调试分；采用Theos、MonkeyDev_Tricks进行开发调试iphone/tool、iphone/tweak
>   - CocoaAsyncSocket tweak
>   - WebScoket_tweak
>   - Daemon tool
>   - knaso
>   - 安全保护

    - 静态混淆: Static_obfuscation
    - 动态保护：反调试(`PT_DENY_ATTACH = 31`参数用于告诉系统阻止调试器的依附;利用`sysctl`函数查看当前进程信息，判断是否有此标志位来检测是否处理调试状态)、反反调试、反注入(`通过_dyld_get_image_name()获取加载的模块名，判断是否都在白名单中`)、hook检测（通过`dladdr`函数得到imp地址所在的模块`info.dli_fname`；遍历符号表中的每一个指针，然后判断指针是不是指向`__stub_helper`或者系统模块； 分析函数的内存前几条指令中有没有跳转）、完整性校验（检查文件load command 的修改，获取内存中运行代码的md5值；是否重签名，从可执行文件的LC_CODE_SIGNATURE 读取信息，拿到当前的签名文件的签名信息和编译前的签名信息比对；bid）
    - 代码混淆：
        - 静态库混淆：混淆带有bitcode的静态库 Confusing_static_libraries_with_Bitcode_Sectname
        - 采用LLVM针对源代码混淆

2013年09月-2017年07月

湖南高阳通联信息技术有限公司

IOS开发工程师、Java程序员

- 负责iOS的app包括：和包商户版、和包客户端、和聚宝、和包刷卡、商户接入和包支付能力项目 以及为和飞信app 提供安全支付组件
    - 对iOS商户和包支付插件项目进行代码重构，进行底层的网络框架进行封装。独立设计iOS应用架构。
- java 服务端开发
    - 对高阳ICS 金融平台有较深刻的理解，能熟练开发与第三方系统的接入接出接口
- 主要职责：完成app的设计、开发、测试、修改bug等工作，包括业务需求的沟通，功能模块详细设计，业务功能实现与单元测试。
    - 除了担任iOS工程师角色外，还担任过项目经理，以及维护支付平台的UI后台管理系统。UI对Portal、IPOS、SMS、Wap、CAS、IVR一共6个渠道提供了规则的配置。

项目经验

2017年07月-至今

ASO(机刷)

- 项目描述 : AppStore Search optimize ， 更多具体细节信息可参考@houshuai0816在GitHub开源的项目knaso
    - 下载流程：清理进程和数据（包括修改设备信息）、切换IP、登录appID、打开App Store、在App Store搜索应用、下载并安装app（打码）、注销app ID、关闭App Store、卸载app

    - 评论流程：在下载流程的基础上进行评论
- 使用的技术(举个其中一点，这些在GitHub上都有开源的code,这里只是点到为止)
    - Wi-Fi的切换和控制
        - wifiutil
        - code forked from qpiu/wifiutil
    - 使用制作 cydia repo 进行部署
        - how_to_host_cydia_repo
        - cydiarepo
    - 进程通信：librocketbootstrap,更具体的请看这里
    - 设备信息的修改：`使用capstone 对二进制文件进行反汇编来定位方法的地址，以便于MSHookFunction`
        - 使用capstone disassemble MGCopyAnswer and locate the subroutine for hooking，具体可看这里、
        - ChangeCode.m;
        - 当然有些app是直接使用`sysctlbyname`进行获取的设备信息此时就要直接`MSHookFunction(&sysctlbyname, &new_sysctlbyname, &old_sysctlbyname);// 可以考虑注释 具体的可以看` 这里
- 职责描述：
    - 负责开发、维护、运维；其中包括适配最新的操作系统版本iOS11
- 项目业绩：
    - 下载任务的成功率由40%提升到60%；
    - 优化打码流程，节约成本；
    - 优化进程间的通信；
    - 更换 cydia repo 的方式进行部署，降低运维成本；
    - 优化VPN的切换IP方式；
    - 新增网络异常自动处理流程。

2017年10月-至今

大型app的逆向开发(非越狱App集成、tweak)

- 项目描述： ios weChat、moon; macOS weChat;逆向系统框架，逆向系统应用；实现一些特殊的功能
- 职责描述：基于特殊app的现有功能，进行功能的优化延展。
- 项目业绩：快速按时实现需求功能，集成tweak的App支持非越狱环境以及越狱环境，支持多架构（arm_v7、arm64; 采用`lipo -create`解决 dumpdecrypted 之后的架构不匹配的问题）。以及支持系统版本范围：V8-V12
- 使用的技术
    - 动态库的注入原理：
        - 一个是基于修改Mach-O 的Load Commands，即通过修改可执行文件的Load Commands来实现的. 在Load Commands中增加一个LC_LOAD_DYLIB , 写入dylib路径。`Usage: insert_dylib dylib_path binary_path [new_binary_path]`
        - 一个是利用环境变量DYLD_INSERT_LIBRARIES,例如使用它进行dumpdecrypted（补

充：**Clutch** 通过`posix_spawnp`生成一个新的进程，然后暂停进程并dump内存）

- 另一个是在挂载的进程上创建一个挂起的线程, 然后在这个线程里申请一片用于加载动态库的内存,然后恢复线程,动态库就被注入(通过 taskfor_pid函数获取目标进程句柄, 然后通过在进程内创建新线程并执行自己的代码。) cycript 就是以这种方式执行脚本代码。

- hook 的方式： 一个是通过修改内存中懒加载和非懒加载符号表指针所指向的地址来达到修改方法的目的，作用于主模块懒加载和非懒加载表的符号，在越狱和非越狱环境都可以使用,例如fishhook（符号表替换）。一个是 cydia substrate: `通过inline hook的方式修改目标函数内存中的汇编指令，使其调转到自己的代码块，以达到修改程序的目的；主要是针对c,c++ 函数。同时支持针对oc的method swizzle（替换imp）`

  - hooking-swift-methods 利用MSHookFunction、MSFindSymbol进行实现。

    - Using nm , we can dump the Swift symbols
    - Using MSFindSymbol we can find the function pointer to the Swift method, and call it

  - `hook MGCopyAnswer 使用了 libcapstone + dlopen+ MSHookFunction`

    - 使用capstone进行MGCopyAnswer方法地址获取，然后使用MSHookFunction对方法进行hook
    - `gestalt = dlopen("/usr/lib/libMobileGestalt.dylib", RTLD_GLOBAL | RTLD_LAZY);`
    - `size_t CAPSTONE_API cs_disasm(csh handle, const uint8_t *code, size_t code_size, uint64_t address, size_t count, cs_insn **insn);`

  - facebook/*fishhook*符号表替换:: `struct rebinding { const char *name; void *replacement; void **replaced;};`

    - （`rebind_symbols((struct rebinding[1]){{"ptrace", my_ptrace, (void*)&orig_ptrace}},1);` 第一个参数为需要替换的符号，第二个参数为自己实现的函数名称，第三个参数为原函数地址，因为他是基于地址进行替换的）+ `__attribute__((constructor))`实现注入

  - `void MSHookMessageEx(Class _class, SEL sel, IMP imp, IMP *result);`

    - `MSHookMessageEx(_logos_class$wxHook$CMessageMgr, @selector(UpdateVoiceMessage:MsgWrap:), (IMP)&_logos_method$wxHook$CMessageMgr$UpdateVoiceMessage$MsgWrap$, (IMP*)&_logos_orig$wxHook$CMessageMgr$UpdateVoiceMessage$MsgWrap$);`

  - `void MSHookFunction(void *symbol, void *replace, void **result);` 注入方式和fishhook 一样，都是基于地址进行hook的

    - `MSHookFunction(&CNCopyCurrentNetworkInfo, &newCNCopyCurrentNetworkInfo, &oldCNCopyCurrentNetworkInfo);`
    - `MSHookFunction((void *)MSFindSymbol(NULL,"_ptrace"), (void *)newptrace, (void **)&oldptrace);`

- object-c 的运行时API： 动态新增属性(objc_setAssociatedObject、objc_getAssociatedObject)；修改和获取属性（class_getInstanceVariable、object_setIvar、object_getIvar）；swizzling交换替换方法的实现（class_getInstanceMethod、class_addMethod、class_replaceMethod、method_exchangeImplementations,想要执行原来的方法就直接调用replaceMethod，因为方法的实现IMP已经被换了）
- LLDB+chisel
  - 给/usr/bin/debugserver添加task_for_pid权限, 并开启`debugserver host:port --attach=<process_name>`(有五种启动方式：auto、posix、fork、backboard 、frontboard)，常用backboard 从头开始调试，例如进行anti ptrace。
  - 使用symbolic breakpoint 进行条件断点。
- 开发的ide和分析工具： theos、monkeydev、lipo、cycript、hopper 进行查看交叉引用，修改汇编、frida-ios-dump、mach-oView、KNtoggle-pie：changes the MH_PIE flag of the MACH-O header on iOS applications to disable ASLR on applications，只支持app,不支持动态库，因为动态库需要ASLR特性的来保证模块基地址不冲突、AntiAntiDebug.m

2017年07月-2017年10月

vpn类 iOS app、导购类iOS app、微信小程序等项目

> - 项目描述：
>   - 阿布云隧道 app、逛有钱app、疯狂笔划微信小程序；
> - 使用的技术：
>   - Swift、lua、iOS逆向开发、iOS正向开发、`git "https://github.com/zhuhaow/NEKit" "master"`(A toolkit for Network Extension Framework)、gzip格式传输数据
>   - ips 文件的分析： 主要分析两块，一块是Triggered by Thread线程的调用栈信息，一块是Binary Images信息。通过命令行工具 symbolicatecrash 来手动符号化 crash log(`symbolicatecrash --dsym=/Users/devzkn/dylib.dSYM /Users/devzkn/SpringBoard-2018-03-23-153316.ips | open -f`)
>   - 把函数名隐藏在结构体里,以函数指针成员的形式存储，编译后，只留了下地址。
> - 职责描述： 开发、维护
>   - AppStore应用协助审核，马甲包制作
>   - App代码加密和混淆工具开发；
>   - AppStore排行榜提升，ASO关键词优化；
>   - 阿里百川SDK 的集成
>   - iOS app 功能开发
> - 项目业绩： 完成app上线，在优化时期，对应的关键词排名前三

2016年02月-2017年07月

和飞信（iOS APP）

- 项目描述: 和飞信是中国移动提供的基于4G网络和2G、3G、WLAN、4G四网协同环境下的基础通信服务，是语音、消息和通讯录等基础通信业务在4G下的升级，是力求满足客户基础通信升级和社交需求的新产品。产品形态包括Native终端、手机APP客户端、PC客户端三种
- 职责描述：
  - 担任iOS工程师角色:负责和飞信中的扫码付、支付密码管理、绑卡、充流量、发红包功能，以静态库的形式提供
  - 负责书写接入文档，维护技术方案文档，解决联调问题
- 技术
  - 支付插件中截图反馈功能的实现 `pod 'KNPodlib'`
  - 设备信息的获取
  - 除了使用UIDevice， 还可以使用`sysctl` 获取cpu、macaddress信息，以及使用`sysctlbyname`获取设备型号等信息、 使用`CNCopyCurrentNetworkInfo`获取ssid、bssid
  - 关于设备ID的心得：
    - 逆向研究一些app的设备ID都是基于CFUUIDCreate、CFUUIDCreateString 进行创建，包括其中的开源的OpenUDID 也是采用CFUUIDCreate、CFUUIDCreateString 进行创建。
    - `_idfa = [ASIdentifierManager sharedManager].advertisingIdentifier.UUIDString;`
    - `_idfv = [UIDevice currentDevice].identifierForVendor.UUIDString;`

2015年05月-2015年07月

搭建一个提高开发效率的iOS静态库工程

- 项目描述： 由于经常要以静态库的形式提供支付能力，因此搭建一个通用的效率调试静态库代码的工程模板是很有必要的
- 职责描述：
  - 负责完善此模板
- 技术
  - 搭建一个提高开发效率的iOS静态库工程
    - code
  - 这段期间的技术提升:对自定义控件、消息推送、通知、键盘处理、屏幕适配、iPad的开发等实用技术进行锻炼；加强自身对UIKit以及Map Kit的理解以及谓词NSPredicate技术的应用。
  - 体会了真正的MVC思想：视图不依赖于具体的数据类型，而是依赖于遵守特定协议的数据源。
    - M 和V 是不存在依赖关系：就像UIKit中的UItableview一样，什么样的数据M，UItableview都可以展示，只要M遵守实现了UITableViewDataSource协议。

    - 最后发现如果使用采用 `pod lib` 开发并打包静态库也是很方便的,并从此学会了how_to_Using_CocoaPods
        - https://kunnan.github.io/tags/#CocoaPods
        - https://kunnan.github.io/tags/#CocoaTouchStaticLibrary

2015年03月-至今

和包支付能力SDK（iOS CocoaTouchStaticLibrary）

- 项目描述 :主要用来向其它的应用程序提供便捷、安全以及可靠的支付服务。
- 职责描述
    - 担任iOS工程师角色。负责整个静态库工程的搭建，以及demo工程的创建
    - 负责和包商户接入支付能力组件的问题解决。
    - 维护接入文档，以及技术方案文档
- 技术心得体会
    - 体会了采用xcworkspace 进行对xcodeproj的管理的方便。
    - 使用`pod lib`进行静态库的开发和打包

2015年01月-2017年07月

和包商户版（iOS app）

- 项目描述 :中国移动面向手机用户及合作商户提供的一项综合性移动支付管理服务.
- 职责描述 :负责整体app开发和维护
    - 主要功能有消息中心，订单查询，订单消息推送功能、收款。
- 业务上的收获
    - 商户收款的流程：主要有扫码付、动态密码支付、客户端支付方式进行收款。 此项目嵌入的“我要开店”、“商户资料维护”功能模块 使用H5 实现，方便以后的功能更新;
- 技术心得
    - 使用coreData 实现消息中心的功能
    - 使用了谓词技术，实现按照日期分组订单列表
    - 使用PushMeBaby工具调试消息推送
    - 采用Charles工具进行问题分析，进行报文分析，修改报文进行开发以及单元测试
    - 总结出app嵌入H5页面的通用模版
        - `pod 'KNBaseWebViewController'`,主要适配webView使用相册的功能。

2014年01月-2017年-07月

和包支付（和包官方iOS客户端）

- 项目描述 :和包客户端是中国移动和包业务针对手机推出的综合性生活服务平台.和包已发展成为融合了支付、生活服务、购物等多个场景的开放性平台。用户可通过和包客户端在手机上不仅能使用中国移动提供的移动支付服务，如缴话费、收付款、生活缴费、订单支付等，还能在电子券商城购买商品和服务，在带给用户随时随地随身的移动支付体验的同时，还可确保用户交易的安全性和便捷性。 主要功能包括：

    - 1、话费缴纳、流量充值；
    - 2、生活缴费（水费/电费/燃气费）；
    - 3、收付款、订单支付、账户充值、提现、账户及交易明细查询、注册和更新；
    - 4、在电子券商城购买商品
- 职责描述: 担任iOS工程师角色，并担任V3.6-3.9 的项目经理。负责整个项目的进度、风险包括，协调厂商以及技术设计
- 业务收获
    - 收获了支付产品的业务知识：支付方式主要有有账户和无账户体系。这两个体系下有借记卡和贷记卡支付，以及快捷和无磁有密支付方式。
- 技术
    - 采用静态库的形式提供安全支付组件给主端集成。优化静态库工程的网络框架的代码，解决了由于更换第三方开源框架导致业务逻辑代码大范围修改的问题。
    - 总结出一套，集成方唤起支付插件，以及支付插件自己退出界面的方法。

2013年11月-2014年01月

UI后台管理系统

- UI对Portal、IPOS、SMS、Wap、CAS、IVR一共6个渠道提供了规则的配置。包括商户的接入流程审核，用户的身份证审核，清结算、风控、银行入账规则的维护等等。
- 职责描述 :
    - 负责基础（用户和商户）模块的规则维护
    - 包括以下需求的开发:
        - 平台的报文验签
        - 商户结算打款成功短信通知优化
        - UI系统银行入账配置
        - UI增加商户信用级别商户清单查询
        - UI商户清单信息查询
        - 2014现场商户线上签约协议优化
        - UI系统客户端侧的CCA接口权限配置的快照改造
        - 安卓3.9客户端启动页面配置
        - 客户端公众服务号管理
        - 公众服务消息推送列表查询。
- 业务收获
    - 熟悉并掌握了支付平台的支付流程、基础模块以及客户端模块的业务
- 技术收获
    - 对JS、JSP、SQL、Linux技术的进一步巩固.

2013年09月-2014年02月

和包支付平台 (ICS)

- 项目描述 : 中国移动面向个人和企业客户提供的一项综合性移动支付业务
- 职责描述 : 担任java工程师。

- 在无线渠道部门，设计和开发移动客户端的接口，来满足app需求的开发。
- 通过此角色，能熟练使用高阳的ICS金融平台,对职责链模式有更深刻的理解。掌握了支付密码、登陆密码的加密、转加密流程。

技能清单

1、逆向开发技能

1.1 熟悉的领域

- iOS 逆向开发,熟悉`iphone/tweak`、`iphone/tool`、`cydia的repo 制作`、`cocoapods的Specs repo制作`(using-pod-lib-create、private-cocoapods)、使用 capstone 对二进制文件进行反汇编来定位方法的地址，以便于MSHookFunction 。
  - 结合lldb、Cycript、hopper、KNHook、以及AFLEXLoader进行分析快速找到入口。
  - 利用lldb、hopper、theos、MonkeyDev、运行时常用的API 进行开发。
  - 具体的通用技能

```
Theos (iphone/tool、iphone/tweak) 、socat、class dump、
Cycript、lua、AFlexLoader、Hopper、usbMuxd、
CocoaAsyncSocket、tcpdump、Wireshark(rvictl)、nm、
symbolicatecrash、host cydia repo 、jshook、debugserver
Making CocoaPods、OC底层原理：Runtime, Runloop, Mach-O,
```

1.2 例子

- iphone/tool 项目例子：
  - MobileWiFi.framework的使用
  - knDaemonDemo
- iphone/tweak类型举例
  - tweak中使用WebSocket 的例子
  - tweak集成CocoaAsyncSocket
  - 使用capstone进行MGCopyAnswer方法地址获取，然后使用MSHookFunction 对方法进行hook
- The zhangkn CocoaPods Specs
  - https://kunnan.github.io/tags/#CocoaPods
  - https://kunnan.github.io/tags/#CocoaTouchStaticLibrary
- 常见的自动化的功能的实现例子

> 自动设置永不锁屏、自动处理进程通信的问题、网络异常的时候自动关闭VPN（防止VPN有效性过期），
> 并且自动断开Wi-Fi，再次重新连接（防止IP有效期超时）、重启sb、在sb 进行搭建界面

1.3 熟悉的iOS模块

- <Foundation/NSTask.h>： task launch
- iOS PrivateFrameworks(iPhone:~ root# ls -lrt /System/Library/PrivateFrameworks)
  - 1) MobileWiFi.framework: 启动处理Wi-Fi的连接断开
  - 2） SpringBoardServices.framework
  - 3)/System/Library/PrivateFrameworks/AppStoreDaemon.framework/appstored.bundle/appstored
  - 4）PrivateFrameworks/StoreServices.framework/ SSAccountStore
- <mach-o/dyld.h>
  - _dyld_get_image_name
  - https://kunnan.github.io/tags/#dyld
- <SystemConfiguration/CaptiveNetwork.h>
  - CNCopySupportedInterfaces
  - CNCopyCurrentNetworkInfo
- <sys/sysctl.h>
  - sysctlbyname
- ios CoreServices
  - /System/Library/CoreServices/SpringBoard.app/SpringBoard
- MacOSX.sdk/usr/include: /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk/usr/include
  - 1) spawn.h:
  - 2)stdio.h: popen(const char *, const char *)
- jailbreak toolkit
  - Electra iOS 11.0 - 11.1.2 jailbreak toolkit:定制一些自己的功能
  - yalu102 iOS 10.0.0 -> iOS 10.2 : 修改jailbreak.m、dropbear，适配Xcode9；
    - 替换掉一些过期的API

      > 用peopen替换system
      > 使用posix_spawn替代system

2、 正向技能

- 个人技术主页
- 熟悉Swift/C/C++/Objective C/Java，cocoapods的Specs repo制作(using-pod-lib-create、private-cocoapods)
  - 1） 消息推送机制、KVC、KVO、GCD、block、RAC(ReactiveCocoa)、CocoaPod、

IAP内购、Method Swizzling、lldb、UIBackgroundModes、NSPredicate；
    - 2） iOS CocoaTouchStaticLibrary: 探索了出了一个提高静态库工程开发的搭建方法；目前比较喜欢using-pod-lib 进行静态库的开发、打包以及维护
- 深入理解MRC和ARC内存管理机制；
    - 1) 需要释放的资源：imageCache、queue、operations、view、通知监听者的移除、销毁soundID
    - 2） 释放的方法：dealloc 、applicationDidReceiveMemoryWarning、didReceiveMemoryWarning
    - 3）凡是函数名中带有create、copy、new、retain等字眼的，都应该在不需要这个数据的时候进行release。 GCD的数据类型在ARC环境下不需要进行release；[而CF的数据类型在ARC、MRC环境下都需要做release的
    - 4） 内存管理的补充：foundation框架（OC语言）、core foundation 框架（C语言，例如通讯录就是基于这个框架）
- 熟悉iOS中多线程和GCD的使用
    - 0）queue： dispatch_get_global_queue、dispatch_get_main_queue、dispatch_queue_create
    - 1）dispatch：dispatch_after、dispatch_async、dispatch_sync、dispatch_barrier_async（ 场景：在写的过程中不能被读：要等到当前所有并发的block都执行完毕，才会单独执行这个barrier block代码块，等到这个barrier block执行完毕，再继续正常处理其他并发bloc）
    - 2）dispatch_group：dispatch_group_t、dispatch_group_async、dispatch_group_wait、dispatch_group_notify、dispatch_group_enter、dispatch_group_leave-- 简单的场景：请求多个接口，再进行View的渲染
    - 3) dispatch信号量（dispatch semaphore）：

```
//    dispatch_semaphore_t  semaphore = dispatch_semaphore_create(0);
//
dispatch_semaphore_wait(semaphore,dispatch_time(DISPATCH_TIME_NOW, (int64_t)( 16 * NSEC_PER_SEC)));//保证是同步的
dispatch_semaphore_signal(semaphore);//在特定的条件激活信号，往下执行code；有些场景使用递归实现更好
```

- 熟练使用AFNetworking, MBProgressHUD, MJRefresh, Masonry, SDWebImage等第三方库；熟悉Making CocoaPods；熟练使用git和svn等版本管理工具。
    - 尤其喜欢git_subtree进行以子目录的形式引用外部项目进行管理依赖关系
- 具有良好的编码习惯, 掌握常用的数据结构和算法；
    - 良好的英文开发文档阅读能力、面向对象编程、链式编程以及常用的设计模式:MVVM、职责链模式
- unix
    - 1） unix（文件目录结构、df、dm、sort、find、grep、cat、vi、ssh、scp）
    - 2） 自己写的shell工具：https://github.com/zhangkn/KNBin
- 对前端后台开发技术也有一定的了解；

> - jQuery、JSP、css、html、数据库(sql基本操作)、j2ee(Spring + Struts +Hibernate)、tomcat

技术文章

- 安全相关的文章
  - 混淆带有bitcode sectname 的静态库
- 逆向相关的文章
  - 搭建私有Cydia源
  - Common hook methods in iOS
  - 设备信息的修改：capstoneHook64_capstoneHook32 for libMobileGestalt
- 正向相关的文章
  - making_private_cocoapods
  - how_to_Using_CocoaPods](https://kunnan.github.io/2017/04/13/how_to_Using_CocoaPods/)
  - 搭建一个针对iOS静态库开发的主项目，提高开发效率，方便调试
- 调试相关的文章
  - CycriptUsefulCommand
  - symbolicatecrash

培训经历

2013.07-2014.01 中信软件教育中心 培训课程: java高级工程师

语言能力

- 英语 : 听说\读写能力
- 闽南语

自我评价

- 充满激情的iOS逆向与安全工程师、高级iOS工程师
  - 1、熟悉Object-C、Swift、lua、iOS&macOS逆向与安全、java、js开发。 常逛的技术论坛是http://iosre.com/。
  - 2、有前后端开发经验者(包括微信小程序开发);
    - 1 年的java开发经验，3年的iOS app 开发经验，2年的iOS&macOS 逆向开发经验
  - 3、对待工作主动积极，责任心强，对代码规范有轻微强迫症，能良好处理人际关系。 爱好游泳、羽毛球、篮球。
  - 4、近期的学习计划：x86和arm的指令集转换，参考Xcode的模拟器的系统库。

- 更多具体信息
  - https://github.com/zhangkn
  - 实时更新的简历： https://kunnan.github.io/2018/08/25/Resume/
  - 整体的附件简历： https://kunnan.github.io/resume/2018-08-25-Resume.pdf

致谢

感谢您花时间阅读我的简历，期待能有机会和您共事。