漫谈移动银行安全性

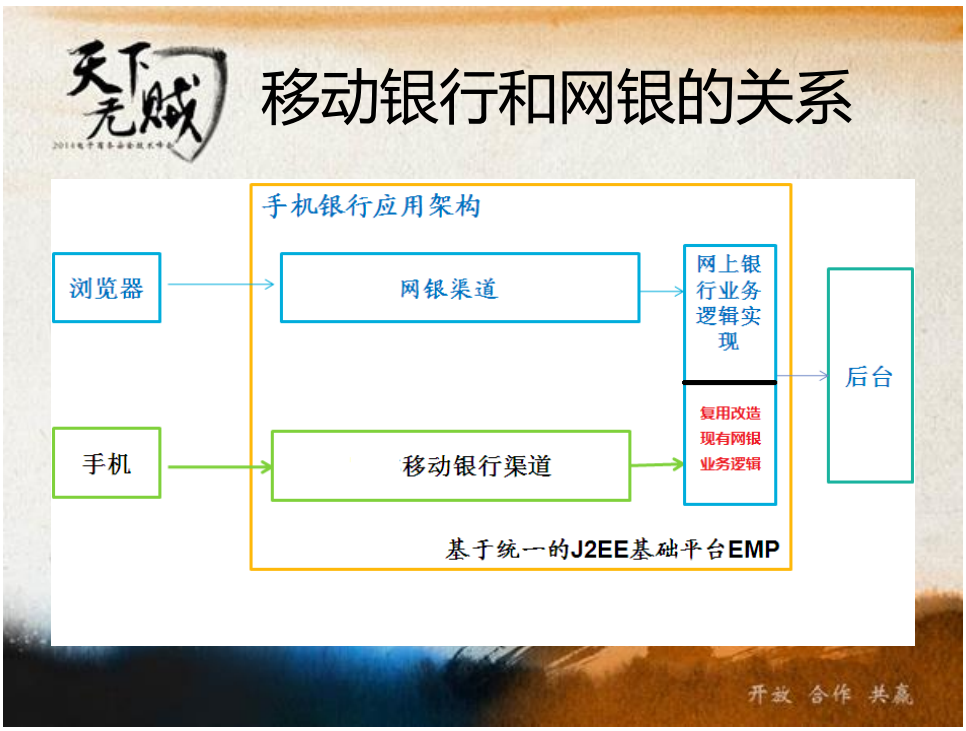
About me

- flyh4t
- PHP安全爱好者
- 绿盟科技安全技术部总监
- 六年应用安全测试、渗透测试、安全评估经验
- machuanlei@nsfocus.com

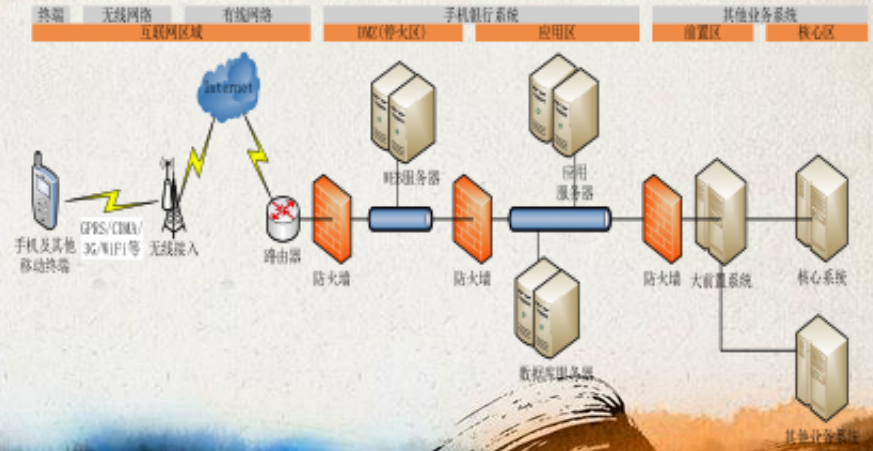

移动银行技术实现
开放 合作 共赢


移动银行分类
SMS
•利用短信办理业务
STK
•将银行服务的菜单写入特制的STK卡，为客户的菜单式操作
USSD
•USSD是一种基于GSM网络的新型交互式数据业务，可以用于开发各种业务
Kjava
•Kjava是专门用于嵌入式设备的Java应用，是Java技术在无线小终端设备上的延伸
BREW
•brew是一种基于CDMA网络的技术。用户可以通过下载应用软件到手机上运行，从而实现各种功能
WAP
•Wap是开发移动网络上类似互联网应用的一系列规范的组合
APP
•app
•微信 / 易信
开放 合作 共赢

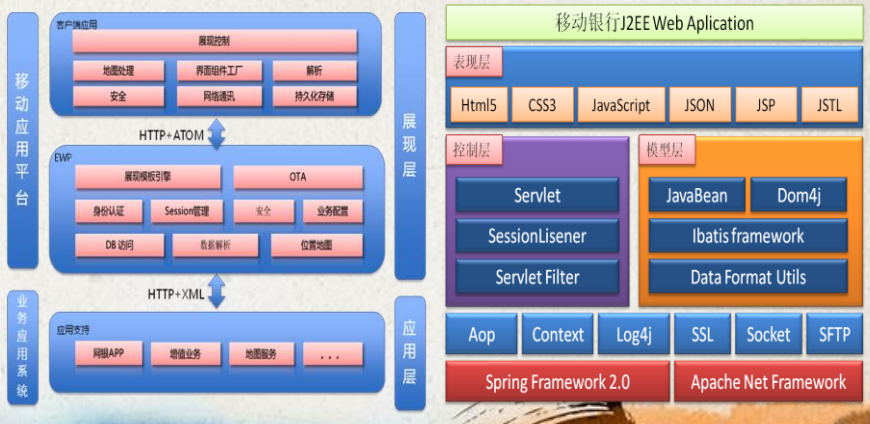

移动银行开发商
开发商
融易通
联龙博通
中科金财
恒生电子
上海屹通
科蓝软件
银行自主开发
.....
开放 合作 共赢


移动银行和网银的关系
手机银行应用架构
浏览器
网银渠道
网上银行业务逻辑实现
后台
手机
移动银行渠道
复用改造现有网银业务逻辑
基于统一的J2EE基础平台EMP
开放 合作 共赢

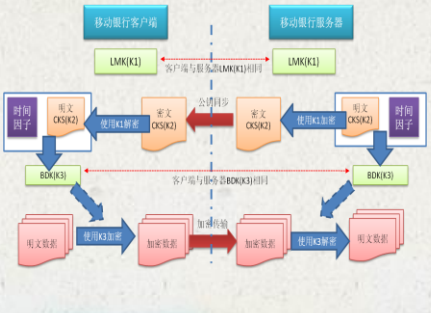
网络拓扑图

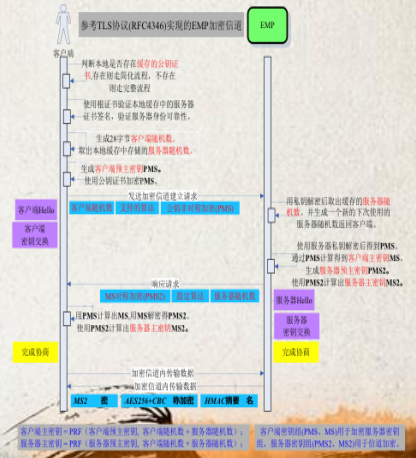


移动银行系统结构



二次加密技术方案





开放 合作 共赢

业务处理

```
Map<String, Object> execute(HttpServletRequest request) {
usinessContext busiCtx = null;
><String, Object> rst = new HashMap<String, Object>();
初始化数据总线
siCtx = busiCtxFactory.createBusiContext(request);

初始化页面请求
siCtx.initParamCover(jsonParamCover, true);
ring trans_code = busiCtx.getParam("TransId");
ger.info("TransId=" + trans_code);

检测重复提交
olean reCommit = controlReSubmit.isReSubmit(request,trans_code);
!reCommit){
 return TransUtil.buildResponseErrorMessage("E111111",MBankMessage.E111111);

发送执行交易
olean result = this.execute(busiCtx, request, trans_code,ebankNetTools);
result){
 logger.info("base context:\n"
  + StringUtil.replacePwdFormXml(busiCtx.getBaseContext().asXML()));
 busiCtx.setParam("RESULT_TYPE", "0");
 TransUtil.buildResponseMessage(AppConstants.RspCode_OK, "交易成功",rst);

将接口返回的数据组装给前台
sageTools.elementToMap(busiCtx.getResponseContext(trans_code), rst);
插入操作日志
nsLog.saveOprateLog(request, busiCtx);
插入交易日志
paramMap = buildTransParam(request,trans_code,busiCtx,rst);
```

```
<!--CS 手机银行所需字段-->
<pkgid>123456 </pkgid>
<!--出错码，无错误情况下为0-->
  <gc>0</gc>
<!--出错消息，无错误情况下为空-->
  <gm></gm>
<!--前端跳转目标页面，仅供 WEB2.0 前端使用-->
  <rp>/html/transfer/interRemit/b020903_interRemitDetail.htm</rp>
<!--数据内容-->
  <cd>
    <payAccount>6[redacted]9</payAccount>
    <currencyType>USD</currencyType>
    <cashFlag>0</cashFlag>
    <tranAmt>200.00</tranAmt>
    <deals>
      <deal>
        <tranDate>20131018</tranDate>
        <tranAmt>100.00</tranAmt>
      </ deal >
      <deal>
        <tranDate>20131019</tranDate>
```

开放 合作 共赢


移动银行面临的安全问题
开放 合作 共赢


移动银行面临的安全问题
移动银行
安全策略
安全实现
基础环境
合规性
业务安全
管理流程
技术架构缺陷
客户端/服务端漏洞
逻辑漏洞
木马病毒
重打包攻击
钓鱼攻击
企业内部
行业
开放 合作 共赢

移动银行面临的安全问题

- 安全策略

开放 合作 共赢

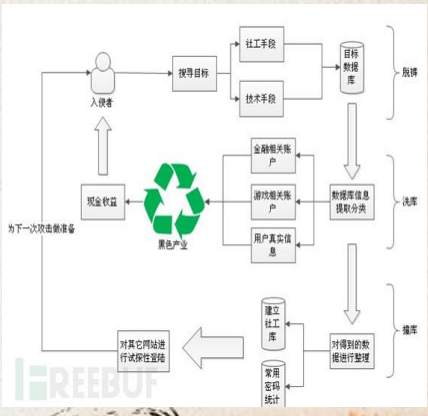
业务安全策略

冒名补办手机卡盗转存款“手机银行大盗”团伙落网

通过在异地补办他人手机卡登录手机银行转账，四川一盗窃团伙在1个月内盗转他人资金1600余万元，118名受害者遍及全国29个省份。湖南省公安厅23日召开新闻发布会，首次披露了这起案件。

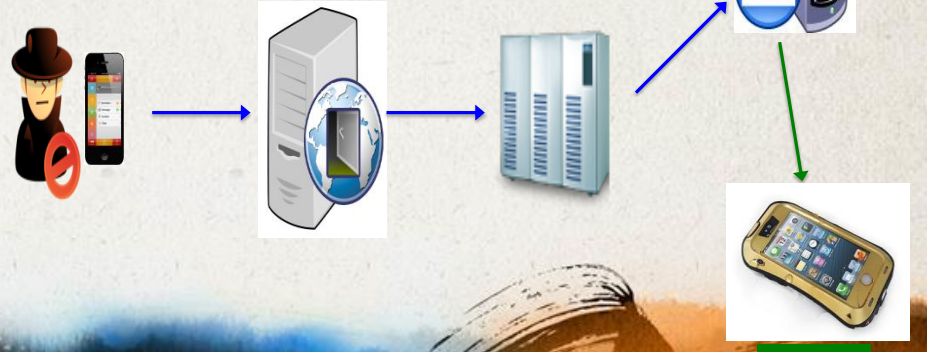
据介绍，2013年9月，湘潭市九华工业园发生一起盗窃案，受害人在银行卡和手机均未离身的情况下，手机银行账户被盗转40多万元。警方侦查发现，嫌犯是在非法取得受害人手机号、账户等信息后，在异地补办手机卡，并通过手机银行客户端进行转账。2013年9月至10月，湘川警方在川闽粤三地将12名犯罪嫌疑人抓获。

警方透露，嫌犯是凭借花费20万元从网上购得的3000万条公民个人信息“起家”的。目前，本案已进入起诉阶段，赃款已全部追回。据新华社



开放 合作 共赢

补卡攻击



开放 合作 共赢

管理流程

漏洞概要

缺陷编号： WooYun-2013-27759
漏洞标题： 中国银行移动客户端严重问题（可导致代码等信息泄露）
相关厂商： 中国银行
漏洞作者： 基佬库克
提交时间： 2013-07-04 21:45
公开时间： 2013-08-18 21:46
漏洞类型： 设计缺陷/逻辑错误
危害等级： 高
自评Rank： 20
漏洞状态： 已交由第三方厂商(cncert国家互联网应急中心)处理
漏洞来源： http://www.wooyun.org

5.1.3 动态图像验证码功能存在万能验证码

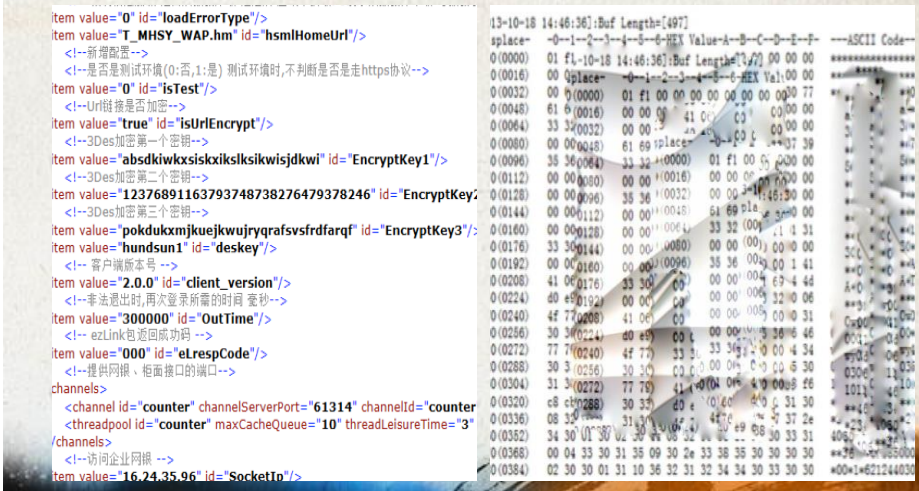
5.1.3.1 漏洞概述

漏洞描述	手机银行系统采用动态图片验证码防范客户使用机器人程序不断尝试获取用户密码。它在图片中加入肉眼可以识别而计算机不能自动识别的随机数字或符号，从而达到防范黑客攻击的目的。但是开发人员在实现图像验证码过程中存在一个逻辑缺陷，产生了一个万能验证码hccb，无论系统显示给客户的是什么图像验证码，只要用户输入hccb就可以绕过验证。
利用场景	黑客通过逆向分析客户端程序，可以发现该万能密码xxcb，进而可以固定图形验证码，使用自动化的程序批量破解手机银行的用户密码。虽然密码错误达到一定次数后会触发账号锁定策略，黑客可以选择固定密码批量跑账号的方式，而账号就是手机号，格式较为固定，很容易进行自动化的暴力破解攻击。
可利用性	容易。
风险等级	高风险，可造成大量账号密码被破解。

开放 合作 共赢

Debug接口未关闭

Debug接口未关闭



移动银行面临的安全问题

- 安全实现

技术架构设计缺陷

漏洞概要

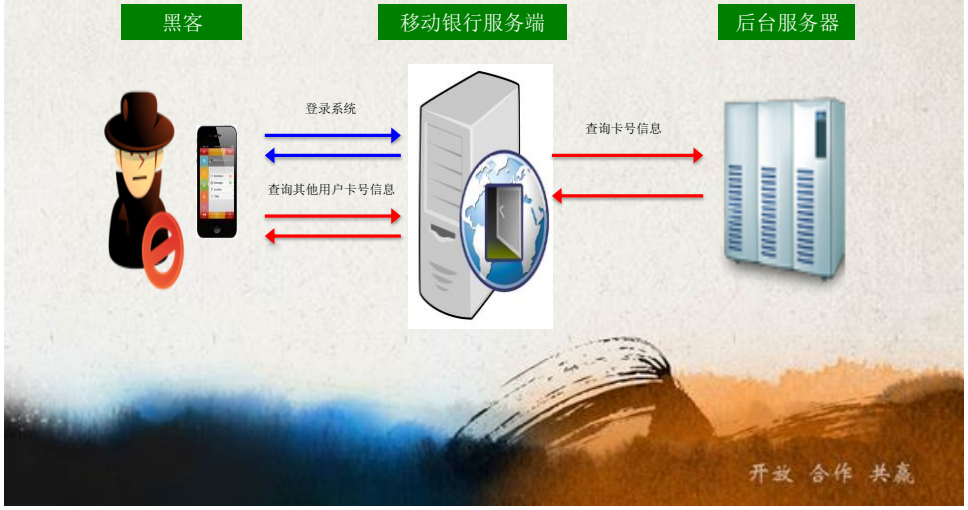
缺陷编号：WooYun-2013-41332
漏洞标题：民生银行android客户端可查询修改他人账号各种信息
相关厂商：中国民生银行
漏洞作者：lupin
提交时间：2013-10-29 10:02
漏洞类型：非授权访问/认证绕过
危害等级：高
漏洞状态：已交由第三方厂商(cncert国家互联网应急中心)处理
漏洞来源：http://www.wooyun.org
Tags标签：无
分享漏洞：分享到 0

5.1.11 转账交易手机短信验证机制可被绕过

5.1.11.1 漏洞概述

漏洞描述	手机银行进行转账交易时，由客户端发起getsms获取交易短信验证码的请求，通过手机银行服务端应用程序转发到后台网银应用程序。后台网银系统发送短信验证码到用户的手机中作为转账交易的第二验证渠道。后台网银程序返回到手机银行服务端应用程序的响应数据包中包含了该验证码，手机银行服务端应用程序未做处理直接将该响应数据包返回到了发起请求的客户端，数据包中包含本次交易的短信验证码。
利用场景	黑客在攻击目标用户时，如果无法控制目标用户的手机短信，可以通过截取客户端程序和服务端程序的交互数据，直接抓取到短信验证码。
可利用性	困难。
风险等级	高风险，此漏洞导致交易短信验证码机制形同虚设。

越权查询漏洞

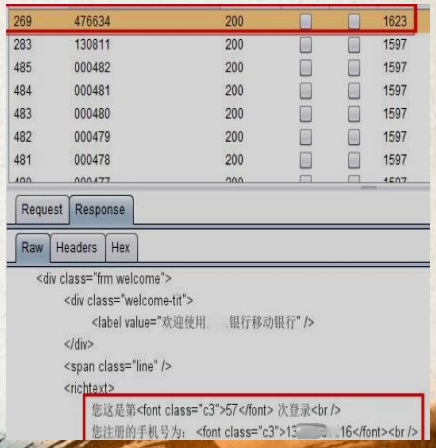


短信验证码泄漏漏洞

短信验证码暴力破解

```
if(System.currentTimeMillis()-180000>Long.parseLong(sms_time)){//三分钟
        priDataCache.setParam("respcode", "m2020");
        priDataCache.setParam("respmsg", "短信验证码已经超时，请重新获取");
        return -1;
}

if(sms_input!=null&&sms_input.equals(sms_yzm)){
        ///add by lancan 2013/10/10 用过的短信验证码不再使用
        priDataCache.setParam("sms_yzm", "");
        priDataCache.setParam("sms_yzm_time", "");
}else{
        priDataCache.setParam("respcode", "m2021");
        priDataCache.setParam("respmsg", errMsg);
        return -1;
}
```



客户端漏洞：未验证ssl证书

```
public EasyX509TrustManager(KeyStore keystore)
        throws NoSuchAlgorithmException, KeyStoreException {
    super();
    TrustManagerFactory factory = TrustManagerFactory
            .getInstance(TrustManagerFactory.getDefaultAlgorithm());
    factory.init(keystore);
    TrustManager[] trustmanagers = factory.getTrustManagers();
    if (trustmanagers.length == 0) {
        throw new NoSuchAlgorithmException("no trust manager found");
    }
    this.standardTrustManager = (X509TrustManager) trustmanagers[0];
}

public void checkClientTrusted(X509Certificate[] certificates,
        String authType) throws CertificateException {
    standardTrustManager.checkClientTrusted(certificates, authType);
}

public void checkServerTrusted(X509Certificate[] certificates,
        String authType) throws CertificateException {
    if ((certificates != null) && (certificates.length == 1)) {
        certificates[0].checkValidity();
    } else {
        standardTrustManager.checkServerTrusted(certificates, authType);
    }
}

public X509Certificate[] getAcceptedIssuers() {
    return this.standardTrustManager.getAcceptedIssuers();
```

客户端漏洞：未验证ssl证书

```
public MySSLSocketFactory(KeyStore truststore) throws NoSuchAlgorithmException, KeyManagementException, KeyStoreException, UnrecoverableKeyException {
    super(truststore);
    TrustManager tm = new X509TrustManager() {
        public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {
        }

        public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {
        }

        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    };
    sslContext.init(null, new TrustManager[] { tm }, null);
}
```

开放 合作 共赢

服务端漏洞

5.1.1 公共信息模块存在任意文件读取漏洞

5.1.1.1 漏洞概述

漏洞描述	手机银行系统公告信息模块允许客户端远程读取服务中存储的图片并展示到客户端。但是服务端程序代码没检测客户端提交的文件名是否为图片文件，也未过滤路径跳转的字符串../，导致用户可以遍历读取服务器上的重要文件，如手机银行服务端配置文件、服务端程序、操作系统重要文件、数据库配置文件等。
利用场景	攻击者直接提交想读取的文件名，服务器即发送相关文件到客户端，也可以通过浏览器直接利用。
可利用性	容易。
风险等级	高风险，可造成手机银行服务端程序信息泄露并引发后续攻击。

```
db2inst1:oiLu00aMfzZMU:108:105:,,,:/home/db2inst1:/usr/bin/ksh
tuxedo:1ur7jC/iY7PU:109:105:,,,:/home/tuxedo:/usr/bin/ksh
star:2vR59q[illegible]Kfn6:110:105:,,,:/home/star:/usr/bin/ksh
ebank:PWuQ0iA.7i0wo:111:105:,,,:/home/ebank:/usr/bin/ksh
chntst1:/qN[illegible]GInTG/o:112:105:,,,:/home/chntst1:/usr/bin/ksh
termtst1:Zq/[illegible]/DOU4kSQ:113:105:,,,:/home/termtst1:/usr/bin/ksh
chninst1:E[illegible]PiWaw:114:105:,,,:/home/chninst1:/usr/bin/ksh
chntst2:KBo[illegible]2VnaA:115:105:,,,:/home/chntst2:/usr/bin/ksh
chninst2:xB[illegible]MSiaQes:116:105:,,,:/home/chninst2:/usr/bin/ksh
termtst2:f[illegible]M9ZsqrSuQ:117:105:,,,:/home/termtst2:/usr/bin/ksh
termtst3:UT[illegible]kLcpew:118:105:,,,:/home/termtst3:/usr/bin/ksh
[illegible]b:QdE[illegible]623N6.ZPw:119:105:,,,:/home/[illegible]sb:/usr/bin/ksh
s[illegible]32:45Mr[illegible]f1qp2:120:105:/home/st[illegible]2:/sbin/sh
m[illegible]:EQKIU[illegible]02:121:105:/home/m[illegible]s:/sbin/sh
```

开放 合作 共赢

服务端漏洞

·5.1.6 预约转账查询存在 SQL 注入漏洞

·5.1.6.1 漏洞概述

漏洞描述	手机银行预约转账查询模块存在SQL注入漏洞，攻击者可以通过提交精心构造的参数查询服务器端DB2数据库中所有内容，包括其他用户的交易记录、登录密码hash、客户端MAC_ID等敏感信息，也可以借助数据库的扩展功能获取服务器的控制权。
利用场景	黑客可以通过SQL注入工具直接利用。
可利用性	容易。
风险等级	高风险，可造成所有用户的交易记录、登录密码 hash、客户端 MAC_ID 等敏感信息泄露，也可能造成服务器被完全控制。

```
<isNotEmpty prepend="and" property="ENDMOMEY">
    <![CDATA[ PST_TRANAMT <= #ENDMOMEY# ]]>
</isNotEmpty>
<isNotEmpty prepend="and" property="STARTDATE">
    <![CDATA[ PST_ENDDATE >=#STARTDATE#]]>
</isNotEmpty>
<isNotEmpty prepend="and" property="ENDDATE">
    <![CDATA[ PST_ENDDATE <= #ENDDATE# ]]>
</isNotEmpty>
<isNotEmpty prepend="and" property="TYPE">
    <![CDATA[ PST_STT in ($TYPE$) ]]>
```

开放 合作 共赢

服务端漏洞

Overview | Request | Response | Summary | Chart | Notes

```
--- The error occurred while applying a parameter map.
--- Check the transfer.queryPreList-InlineParameterMap.
--- Check the statement &#40;query failed&#41;.
--- Cause: com.ibm.db2.jcc.b.SqlException: DB2 SQL error: SQLCODE: -10, SQLSTATE: 42603, SQLERRMC: &#39;aaa&#39;&#39;&
        at org.springframework.jdbc.support.SQLStateSQLExceptionTranslator.doTranslate(SQLStateSQLExceptionTranslator.java:9
        at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTrar
        at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTrar
        at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTrar
        at org.springframework.orm.ibatis.SqlMapClientTemplate.execute(SqlMapClientTemplate.java:203)
        at org.springframework.orm.ibatis.SqlMapClientTemplate.queryForList(SqlMapClientTemplate.java:293)
        at com.yitong.commons.dao.BaselbatisDao.findList(BaselbatisDao.java:67)
        at com.yitong.mbank.controller.transfer.Transfer.getPreList(Transfer.java:739)
        at sun.reflect.GeneratedMethodAccessor131.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:600)
        at org.springframework.web.bind.annotation.support.HandlerMethodInvoker.invokeHandlerMethod(HandlerMethodInvoker
```

Headers | Text | Hex | HTML | Raw

开放 合作 共赢

业务逻辑漏洞

5.1.8 超级网银转账功能可自定义手续费

5.1.8.1 漏洞概述

漏洞描述	手机银行用户使用超级网银转账功能时，可以自己定义转账手续费，可以将每次转账手续费修改为0.01元。也可以定义为浮点数等，可造成系统异常。
利用场景	登录手机银行后使用超级网银转账功能，提交数据的时候修改数据包。
可利用性	容易。
风险等级	高风险，可以对银行造成直接经济损失，也可能造成系统接口异常。

开放 合作 共赢

移动银行面临的安全问题

- 基础环境

开放 合作 共赢

基础环境漏洞

- 5.1.13 Android 客户端存在远程代码执行漏洞
- 5.1.13.1 漏洞概述

漏洞描述	手机银行Android客户端调用Webview控件的addJavascriptInterface方法使JS与本地JAVA对象交互。该方法存在安全隐患，从Android SDK4.2(API17)开始已经被JavascriptInterface方法取代，并且客户端程序没有对传入的URL进行合法性判断，如果访问了恶意的JS代码，利用该漏洞可能导致恶意代码执行。
利用场景	攻击者通过中间人劫持等方式向客户端程序发送了恶意的JS代码，可能导致用户Android手机被植入后门。
可利用性	一般。
风险等级	高风险。

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
        OSStatus        err;
        ...

        if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
                goto fail;
        if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
                goto fail;
                goto fail;
        if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
                goto fail;
        ...

fail:
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
}
```

开放 合作 共赢

底层安全机制较弱

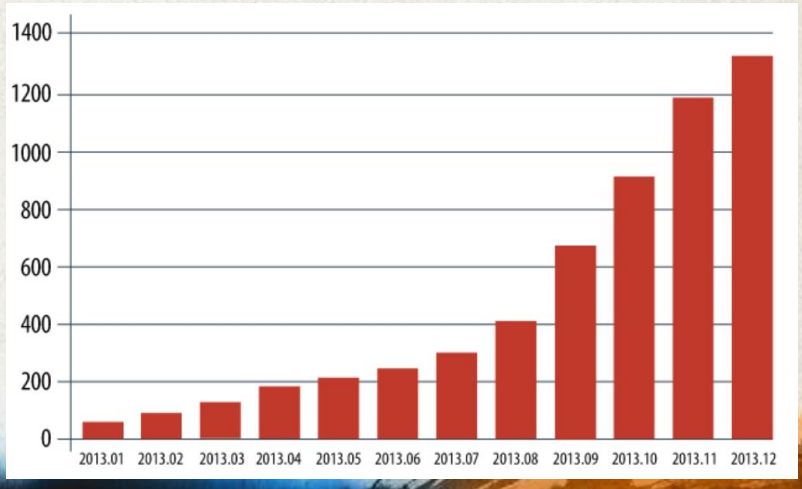
开放 合作 共赢

病毒木马

- 2013年全年，截获手机病毒总量超过79万个，其中截获Android平台病毒包达到76万个；
- 与2012年相比,2013年截获的病毒包总数是2012的4.47倍，手机病毒呈现疯狂增长态势。
- 从2012年到2013年，是截至目前手机染毒用户数激增暴涨的2年。2012年1月~2013年12月，手机染毒总用户数突破1.4亿，接近俄罗斯总人口。；

开放 合作 共赢

移动手机病毒及恶意攻击



开放 合作 共赢

媒体报道

近日，央视曝光了一款名为"银行悍匪"的手机银行木马。报道称，该病毒可窃取用户银行存款，危害极大。记者就此联系了360安全中心，据安全专家介绍，360手机卫士可有效拦防该木马，请360用户放心；同时建议非360用户尽快安装360手机卫士进行手机杀毒。

图：央视报道截图

手机新木马隐身大盗专窃网银验证短信_银行信息港
2013年11月10日 - 手机新木马隐身大盗专窃网银验证短信 据新华社电 360手机卫士6日宣布截获一款名为隐身大盗的手机木马变种,该木马启动后会自动隐藏图标,并伪装成系统...
www.yinhang... 2013-11-10 - 百度快照 - 评价

手机新木马"隐身大盗"专窃网银验证短信
2013年11月7日 - 据新华社电 360手机卫士6日宣布截获一款名为"隐身大盗"的手机木马变种...对此,该专家建议,如果手机绑定银行卡不可避免,尽量不要在银行卡内放过多...
wbnews.sxrb... 2013-11-07 - 百度快照 - 评价

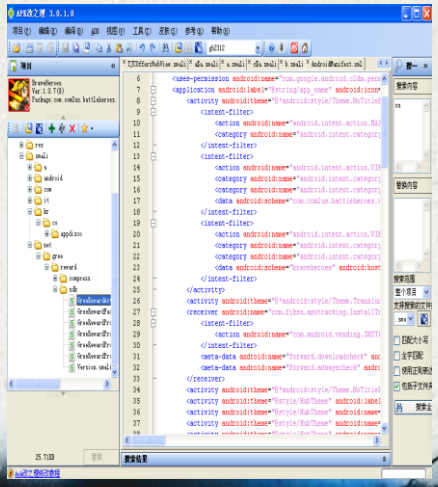
手机木马"隐身大盗"再升级 偷短信还偷身份照号-国内-BTV在线-...
2013年12月1日 - "隐身大盗"二代行为更为隐蔽,采取连网上传短信到黑客服务器的方式,同时,还会判断中招手机接收的短信号码和内容,对普通短信放行,对银行、第三方支付...
www.btv.org... 2013-12-01 - 百度快照 - 评价

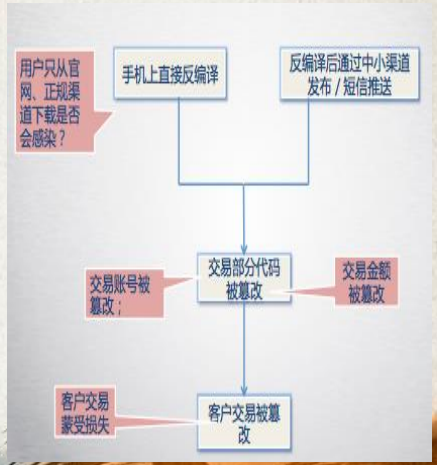
开放 合作 共赢

病毒木马

开放 合作 共赢

重打包攻击





开放 合作 共赢

钓鱼攻击

近日，百度安全实验室发现一款“伪中国移动客户端”病毒，犯罪分子通过伪基站方式大量发送伪10086的短信，诱导用户点击钓鱼链接；并在钓鱼页面诱导用户输入网银账号、网银密码、下载安装“伪中国移动客户端”病毒；该病毒会在后台监控用户短信内容，获取网银验证码。黑客通过以上方式获取网银账号、网银密码和网银短信验证码后，完成窃取网银资金。

伪基站发送的伪10086短信

图1. 伪10086短信

诱导用户领取现金红包界面

新闻晨报V：【二维码扫一扫 一万多元就没了】扫一扫，就能打开网页；扫扫，就能快捷支付……只需用手机对着二维码扫一扫，就能让生活变得更简单时髦。但市民王先生近日在扫描二维码后，其支付宝和余额宝竟被悄无声息地11000多元。专家分析，这背后可能是一种手机木马病毒在作祟http://t.cn/8kqrXGB本报

12月3日11:36 来自皮皮时光机 (39) | 转发(255) | 收藏 |

热门微博

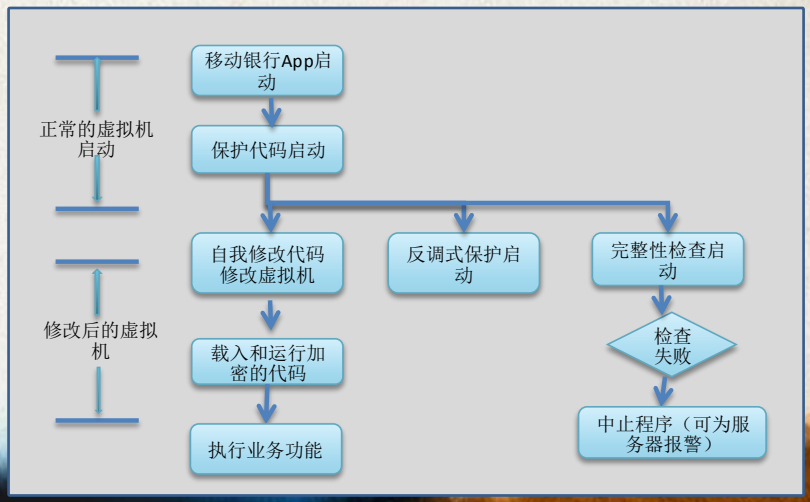
中国新闻周刊V：【二维码千万别轻易扫 有的带病毒！！】11月13日，开网羊毛衫的汪女士，扫描一个“顾客”发来的二维码，打开链接，手机就中毒了。着，对方通过支付宝转走她账户上18万元。据警方调查，仅半个多月，就有2汪女士一样中招，涉案总金额已达26万多。警方提醒，不要随便扫二维码。晚报）

开放 合作 共赢

一种新的黑色产业链

开放 合作 共赢

App加固保护



开放 合作 共赢

Android APP 加固保护

腾讯 | 360 | 娜迦 | 梆梆 | 爱加密 | 通付盾

开放 合作 共赢

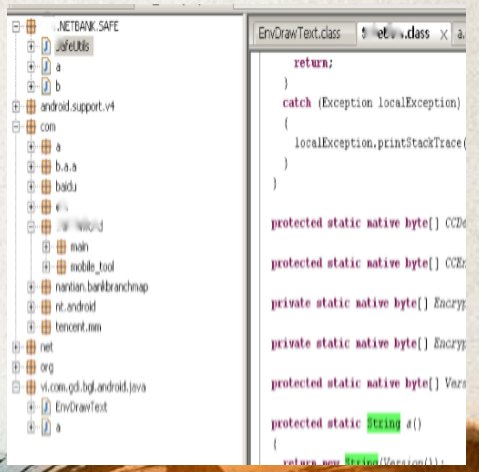
某大行APP破解案例

```
public static void run(Context paramContext)
{
  File localFile = new File("/data/data/" + paramContext.getPackageName() + "/.cache/");
  if (!localFile.exists()) {
    localFile.mkdir();
  }
  String str1 = "/data/data/" + paramContext.getPackageName() + "/.cache/" + "encode.dex";
  String str2 = "/data/data/" + paramContext.getPackageName() + "/lib/" + "libe      p.so";
  String str3 = "/data/data/" + paramContext.getPackageName() + "/lib/" + "libS      ls.so";
  setLog();
  checkX86(paramContext);
  copyFile(str1, "encode.dex", paramContext);
  String str4 = getMetaData(paramContext, "app_package");
  if (new File(str3).exists()) {
    dl(paramContext, str1, str2 + ";" + str3, str4);
  }
  for (;;)
  {
    init = true;
    return;
    dl(paramContext, str1, str2, str4);
  }
}
```

开放 合作 共赢

某大行APP破解案例

```
root@generic:/data/local/tmp # ls
ls
encode.dex
libSa      s.so
libcl      p.so
runme
root@generic:/data/local/tmp # ./runme
./runme
load ok
init ok[+]      real key is 3D D7721DBC73913B0E55
rs:4062064 p: 0xb6259008
write: 4062064
```

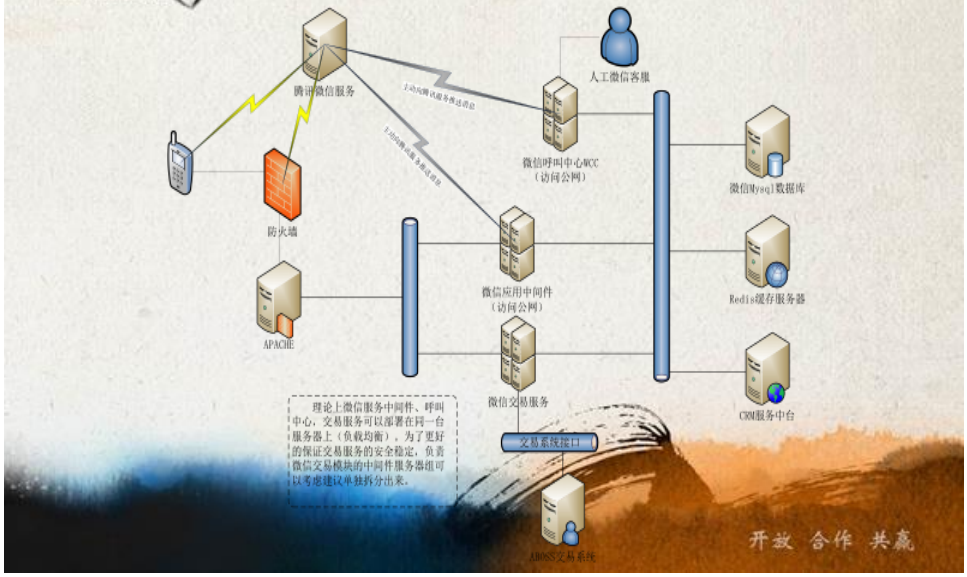


开放 合作 共赢

还可以怎么破

开放 合作 共赢

微信银行安全性



移动银行面临的安全问题

- 合规性

合规性

- 内部的监管
- 外部的监管
  - 人行
  - 银监

开放 合作 共赢

绿盟科技移动银行安全解决方案

- NSFocus MB-SDLC

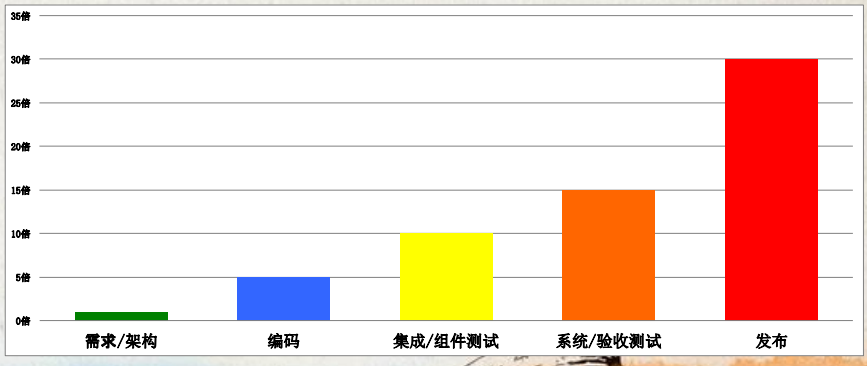
开放 合作 共赢

SDLC

需求阶段		
安全目标识别	安全需求分析	安全需求报告
架构设计阶段		
安全设计原则	威胁建模	安全方案设计
功能实现阶段		
安全编码	安全测试	
运维阶段		
安全保障	应急响应	

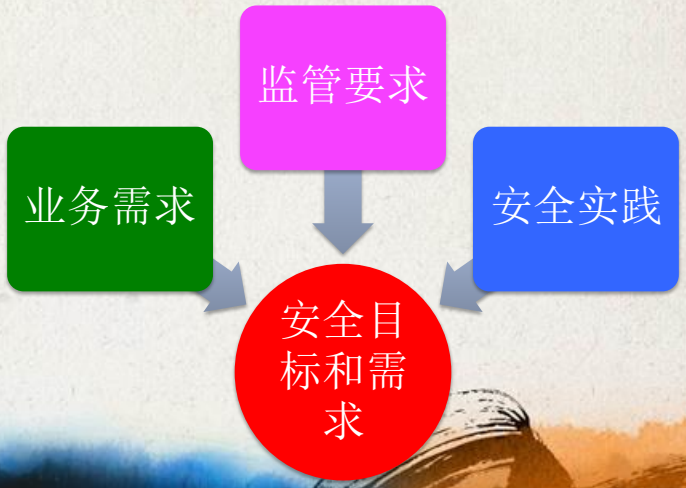
开放 合作 共赢

修复安全漏洞所需要的成本



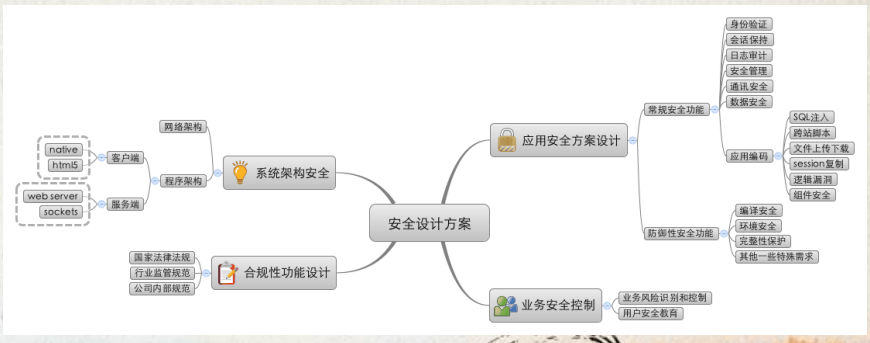
开放 合作 共赢

安全需求



开放 合作 共赢

移动应用安全控制模型

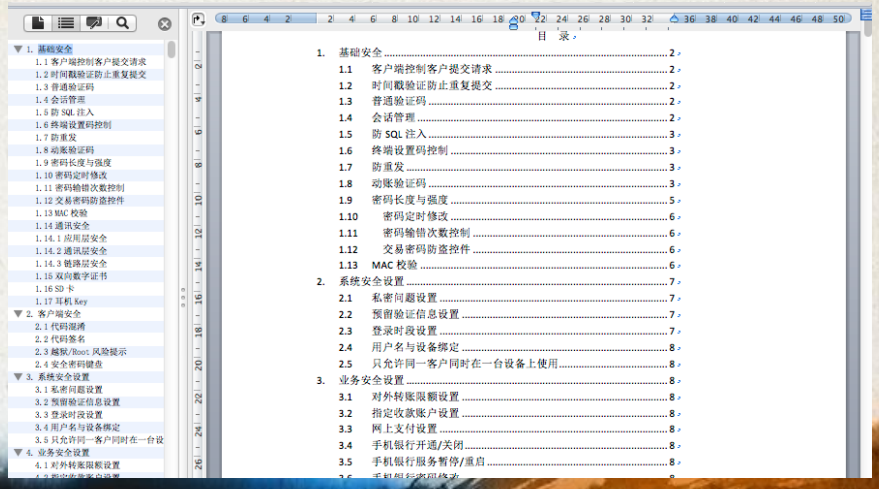


开放 合作 共赢

输出物：安全架构和设计

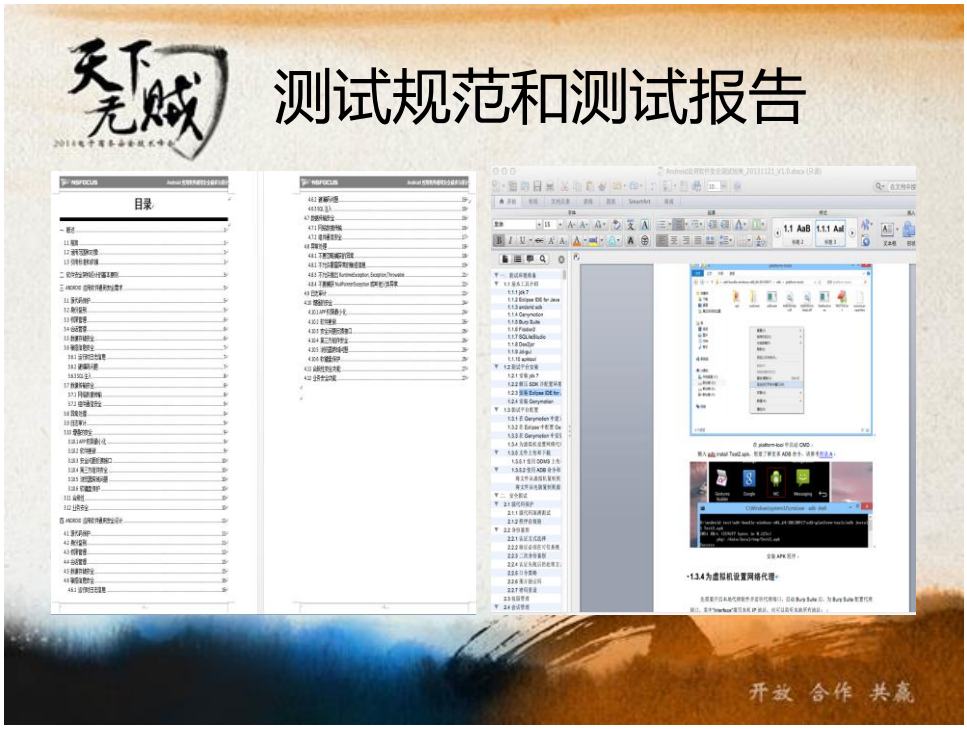
开放 合作 共赢

输出物：安全功能设计

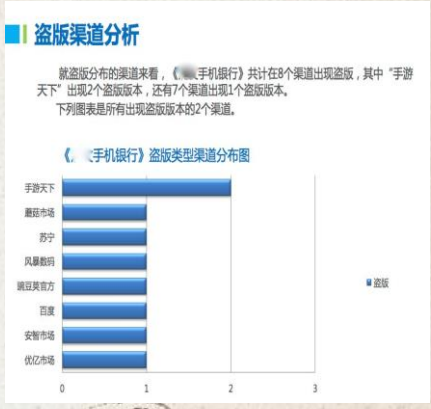


开放 合作 共赢


编码规范和代码审计报告
开放 合作 共赢


测试规范和测试报告
目录
开放 合作 共赢

运维监控



总结

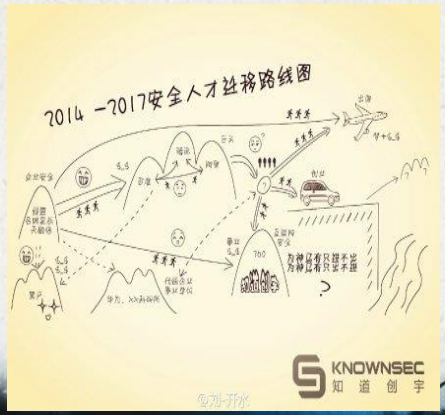
服务	客户端安全	培训
• 架构咨询 • 代码审计 • 安全测试 • 安全评估 • 应急响应	• 安全监控 • app保护	• 安全意识 • 技术能力

乙方工程师那点事

开放 合作 共赢

向坚守在乙方的安全工程师致敬

我们的幸福不是来自于掌握高精尖的技术，而是来自于通过自己的努力解决他人的痛苦。



开放 合作 共赢


天下无贼
2014电子商务安全技术峰会
END
Q&A 谢 谢!
开放 合作 共赢