2014乌云安全峰会文字版记录

转载

aezwm4109 于 2014-09-24 16:36:00 发布 1988 收藏
文章标签： 运维 java 人工智能
原文链接：http://www.cnblogs.com/miyeah/p/3991035.html

转自51CTO，先摘录一些我认为比较精彩的部分。全文在下面噢~~~

猪猪侠： 就如果研究用户密码设置规则的话就得上大数据，这里我们搜集了10亿条用户和用户密码，我们得出了这个东西，我们利用大数据，统计出了100多个使用的最多的弱口令，标红色的是Top10，中国是使用英语较少的国家，我们可以看到前10位全是数字，再看蓝色标记的，这个很有意思，这是一个多情的国家。我还对比一下，我的更具有中国特色，我这是Top数字使用数据，在整个10亿条密码中，使用123456有4千多万，使用123456789的有9千8百多万。

猪猪侠： 分析这10亿密码可以看到密码用纯数字的大概有3.7亿，就占了36%，密码全部用小写字母当密码的占了41.8%，密码全是大写的占了0.38%，用户名和密码相同的也有两千多多万，占了2.36%，密码中包含了一部分用户名的也有1.32%，使用Top10口令的大概占了17%左右，口令中含有特殊字符的占7.24%。

猪猪侠： **这是整个整个针对使用10亿密码计算出来的结果。当我们分析一些密码的时候，往往也能够得到一个规律，就是我根据那些10亿条密码得出的结论，就是大多数人设置密码的时候都会把生日加进去，还会把年份加进去，比如这个密码是去年设置的，会把2013年加进去。还有可能是名字加上生日，做技术员的会转换一下，还有设密码的时候，复杂度有要求，就要有数字和特殊符号，就会加上@，还有键盘布局，和公司相关的东西补充进去，还有些地方要求大小写，要么就是第一个大写，要么第一个单词大写这样，或者把自己姓名的第一个首字母大写。**

猪猪侠： **这是我们发现频率最高的密码设置方式，就可以总结成这样，要么是用户名加生日，用户名加特定字符，要么是用户名@加生日，中文名拼音加生日，中文名拼音@加拼音。**

猪猪侠： 根据这些规律，我们实现了生成密码的算法，可以根据你这个人生成一个和你有关的密码。大家可以看一下，这是一个分隔符，比如很多人会用乌云@123，要么乌云.123这样来连接后面一些特殊常用字符，就有一次我在微博上面看到李铁军，他教别人怎么设置密码，他说他的密码一般都是在后面会加上一堆@123Com，他把这个方法告诉更多人，更多人可能会采用这个方法，所以你只需要看他们怎么设置的，然后把它变成算法，自动生成出来，这是处理生日的一个小函数，我们知道日期这个东西会有前导0的，没有前导0的话是9，有前导0的话是0909。

猪猪侠： 就是你把能够考虑的细节考虑进去，就生成一个专门有针对性的密码脚本。有了生日密码算法，我们发现都是中文名，我们又有一个小函数，就是把中文名转成拼音，然后再把你拼音的名字，比如姓不转成拼音，把你的名取前面的两个开头字母，这里可以看到朱

猪猪侠： 整个生成别人密码的过程是动态生成的，有人会说你根本不知道我的名字，不知道我的生日是没办法知道我的密码的，在这里的话，因为互联网上已经泄漏了10亿多条用户信息，我们知道，所以这里可以做一个数据接口，然后给扫描器调用，给算法调用，这里是我做的一个案例，大家可以看一下，乌云上也介绍过，就是你只需要在这个系统上面输入一个域名，然后它就会返回所有与这个域名为结尾的所有人的信息。

张健： 可能有一些人已经知道这个事情，就是最后这一条，一个公司破产的时候，它宣称丢掉了75万个用户的比特币和10万个自己的比特币，加起来超过4亿美金，这是一个非常惊人的数字。

张健： 第二，黑客攻击。这个案例挺有意思，这个交易所在一年四个月的时间内被人黑了三次，每一次黑的数量都挺大，第一次是4.3万个，第二次是1万8千个，第三次是4万个，加起来差不多有10万个，而且每次黑的方式不同，第一次黑的托管商，第二次闯入帐户通过电子邮件，第三次稍候我会讲一下。

张健： 刚才我问大家比特币价格的时候，不知道有几个人查了，它价格现在人民币不到三千元。这个交易所很有意思，它第三次被盗的时候，在它被盗的前一天，这个交易所创始人在QQ群里大量出售他的LR和B CORN，而且说要多少有多少。

张健： 所以说很多人怀疑这是监守自盗，因为被黑了很多次，不在乎自黑一次，然后彻底关门。因为我一个朋友在这个事件中丢失了500个比特币，所以他要求我一定提下这个人，这个人叫周彤。

张健： 案例三，攻击服务托管商。刚才那个交易所第一次被黑是黑了它的托管商，这个托管商出的事太多，其中一个跟比特币有关，就是超级管理员密码泄露，因为不费工夫就可以更改用户的密码，因为他是一个相当于后台客户服务的帐号被泄露，这个黑客拿到了超级管理员权限之后就扫描所有程序，最后只选择了攻陷八台

张健： 就是因为这八台上运行了比特币程序，也就是说这个黑客就是冲着比特币来的，这八台中出现了这三个人，就是刚才的那家交易所，还有一个矿石，它应该是排第三、第四的，还有一个比特币当时的首席科学家，这个人居然也丢失了50个比特币。所以这个事件给我们的启示，千万不要把私钥放在云平台，即使你没有问题，他可能黑掉你的上级。

张健： 域名劫持，这也是一个挺典型的案件，黑客利用公开的信息，获得了这个拥有者的出生地点和母亲婚前姓名，说服注册商交出了域名和控制权，进而又控制了他的电子邮件，最后进入他在一个交易平台的帐户，然后把这个交易平台全部侵了。

张健： 虽然得逞的原因因为公开的这些情况，可能跟他并没有直接关系，但是他没有开启双边验证，这是一个安全上的漏洞。但是这个公司也挺有故事的，其实这里面每个公司都挺有故事的，这些公司的创始人也是在今年初，因为洗钱被FBI逮捕了，现在应该还在FBI那儿。

张健：　合适这个刚才我也提到了，这个造成的丢失不在少数，包括最大的一个丢失事件，门头沟那个事件，85万个比特币，有部分人怀疑私钥是管理不善，就是没有备份的后果是灾难性的。应该有三四个人找过我，其中一个人相当典型，他属于本地的私钥损坏了，把钱包发给我，问能不能恢复，说恢复的钱分我一半。

张健：　我经过很多次尝试后真的不能恢复。所以这样的情况下，不管你有多少资产，很快就丢掉了，就是说没人能花这笔钱，就是它永远的丢掉了。其实丢掉这些钱让比特币整体的数量稍微减少了一些。

张健：　第二种情况，就是私钥被盗，这个也是非常普遍的。可能每个月，甚至每周都在发生，这个对安全性要求非常高，包括企业，我们公司也会采用冷钱包的存储架构去做这样的安全防范。

张健：　也就是说，因为你要用比特币进行支付，所以你必须在业务的层面保证有一定的私钥或者有一定的逼债，所以你不能百分之百冷存储，所以现在很多交易所将大部分币冷存储，一旦黑客侵入，而且侵入的级别比较深，拿到这些私钥也不会损失在线下冷钱包的钱，这是一个现在普遍的一个架构。

张健：　其实交易安全风险，比特币还是做的非常好，这也是比特币安全性一方面，叫双发攻击，因为比特币是P2P的架构，什么意思呢？就是每花一笔钱，这笔交易是通过这个结点性质广播给全网，就是说全网不同节点接受这个交易的节点时间是不同的，所以有可能你可能同时广播两个交易，一个交易是发给交易的商家，另一个交易是发给自己，通过不同的节点广播，一旦出现了，假如说你又同时掌握了比较大的算例

张健：　你就可以通过自己的算例优先确认你广播发给自己的这笔交易，而推翻发给商家的那笔交易，所以商家会误认为收到这笔钱，但是后来事实上这笔钱消失了，这就是双发攻击。避免这个攻击的方法非常非常简单

张健：　因为这个攻击的成本非常之高，所以说只要在发送大额交易的时候，比如说几千个或者上万个，等待足够多的确认数量就可以避免这个攻击，因为攻击程度概率随确认数量的增加呈指数级下降的。

张健：　谈到比特币私钥容易被盗，它的脆弱性问题，其实所有人都认识到这个问题，包括比特币核心开发团队，所以在2012年采访中，比特币首席科学家说正在添加一个新的特性，这个特性可以降低比特币钱包受到攻击后的脆弱性，这个特性叫多重签名交易。

张健：　什么意思呢？简单讲一下，就是传统的私钥被盗，相当于这个地址商的钱全部被人拿走，但是通过多重签名交易生成，就是多个私钥共同生成这个地址，然后有M个私钥签名这个地址上的交易，然后M和N都可以自定义，只要M小于N就可以，这就造成了安全方面的特性

张健：　不仅仅对存储，还有交易，就是对不同的角色赋予不同的私钥，保证比特币本身的安全性。比特币第一次从技术层面上实现了真正的可执行的担保交易，比如说支付宝的担保交易，它是用支付宝信誉担保的，买卖双方都相信支付宝平台，但是通过多重签名技术，步入说三把私钥生成一个地址，只要其中两把私钥同意才能动这个私钥的钱

Gainover:　　就这样完了吗？当然不止这些危害，这个不仅仅是个漏洞，这是我2013年10月24号我在漏洞下面做的一个评论，我是这么说的，问题很隐蔽，这个开发人员肯定是注意不到的，只要是有这么用的，估计都会存在问题。这是当时我在漏洞下面做的评论。

Gainover:　　后来的话，我看还存在哪些问题？我们看一下乌云已经公开的案例，实际上这三个漏洞的话，与我多少都有点关系，不管是马甲还是间接报告，还是怎么样的，实际上我发了这个之后还是没有引起太多各大企业的注意。第一，全部博客可留后门。

Gainover:　　你只要你是新浪博客用户，不管你以后访问新浪博客任意文件，都会执行我这个恶意代码的。第二个，淘宝支付宝的，这个叫一个可大规模悄无声息窃取淘宝支付宝帐号的。这个标题看起来很大，但是实际上一点都不危言耸听的。

Gainover:　　这些企业不知道自查周期比较长还是怎么样的，同样的漏洞在阿里云服务上一样存在的，后来也被报告修复了。现在我看了一下，淘宝相关的页面上，同等类型漏洞都已经做了相应保护措施了，他们已经意识到这个漏洞危害的严重性了。

Gainover:　　其实我今天给大家讲的，就是说漏洞这块东西，就说到这里。我标题还有后半部分，叫XSS僵尸网络。这个东西敢这么说的话，是有一定漏洞基础的，是因为这个漏洞本身特点决定，它是可以用来做这个东西的。首先，隐蔽性。

Gainover:　　为什么说它隐蔽？这个漏洞本身特点，你想想，对于一个用户来说，可能很多用户对钓鱼网站网址都分不清楚的，你让他在页面来看原代码，看它上面是不是存在恶意代码，这是根本没可能的。

Gainover:　　第二个，传统的杀毒软件，它是很难去防这类攻击的，这个东西对于软件来说，至少很难有办法去区分，是XSS，还是这个网站本身的。所以对用户这一层的话，用户是很难去发觉他们是已经中招的了。

Gainover:　　哪怕我自己搞这方面的，如果不专门看的话，我自己很难察觉，所以我也自己也很担心这类漏洞。这个对厂商来说，这与传统的漏洞不太一样的，传统的是在页面里插一段代码，你接到报告说这个页面有问题，厂商查的话，发现这里黑客插了一段代码。

Gainover:　　但是对XSS来说，这个漏洞是存在用户电脑本地存储上的，用户觉得页面有问题，让厂商去查，厂商去看页面完全没有任何问题，然后就不再过去了。这样的话不管是最用户还是厂商来说，这种类型的漏洞隐蔽性是特别好的。

Gainover:　　第二个，持久性。这是一个长期存在用户电脑上的，这个漏洞原理也是发过的，大家可以看看，它用Flash本地存储，如果你不是自己去清Flash Cookies的话是很难清掉的。

Gainover:　　我记得360软件Flash Cookies是勾上的，以前我看安全卫士软件的话，Flash Cookies是不勾上的，就是用户清理垃圾什么的，不会清理掉这些垃圾，这样的话，这个东西就是在用户电脑上长期存在。

Gainover:　　另外的话，那些软件去来清理这些东西，它就是找到浏览器默认的Flash Cookies存放目录对它清除。我现在不知道第三方有没有清除这个，如果没有我建议以后可以做上。如果没有的话，这个后果很严重的，按照我前面演示，如果这次没有打开网页或者QQ空间不会被窃取，但总有一天你会被窃取。

Gainover:　　第三，流量大。大家可以看到这种类型的漏洞影响到了，基本上是全国所有视频网站，当然有些网站没有用那个代码的话就没有。

Gainover:　　第二个，电商类的也是受影响的，虽然现在都修了。第三个，新浪微博也是受影响的，这也是大流量的社交网站。所以说这个受影响的用户量是非常大的。

Gainover:　　其次，用户页面时间停留长，大家可以看到，特别是对于视频网站来说，它的用户基本上都在看视频，一个小时、两个小时，恶意代码就会在那里被执行一个小时、两个小时，这样黑客利用代码可以获得很长执行时间的，这为后面的攻击是能够提供很大的帮助的。

Gainover:　　最后，基于这些特点的话，这跟前面原理差不多的，我们可以看到，黑客首先做一个感染阶段，它首先找一些比较好入侵的，或者有漏洞的中小型的网站去入侵，入侵了之后就在里面插入这样一个XSS代码，因为这些中小型网站它在上面用户Cookies有什么用啊，一点用没有，所以黑客不会偷这个，它的目的是植入后续的XSS，接着这些FLash XSS在感染到足够用户数量之后，我们可以让FLash XSS执行黑客的操作，最简单的就是信息窃取，当感染数量达到一百万的时候，我假设里面只有1%的用户，比方说他先看QQ空间，再看网站的话，就会有一万人的数据窃取，这可能只是一天的数据，如果是一个月呢？一年呢？

Gainover:　　因为这个漏洞不宜被发现，不易被清理，所以这个积累到一年的话危害是非常大的。而且这些受害者，在看视频的时候，他的浏览器可以被黑客做一些事情的。当然，这个还会有其他的利用方式，这个就要发挥想象力了。所以我这里总结了一下，这种类型的做僵尸网络的话，首先感染阶段，然后是执行阶段，从危害上来说的话，我觉得影响几乎全国所有网站的。下面可以进行提问。

剑心:　　总结起来，我们知道国内安全最大的问题，不是说我们技术有多差或者什么，是环境问题。环境问题总结起来就是封闭的问题。所以在座的了解到，说这个技术我不能说，那个企业说这个事不能对外说，说了怎么样。这才是真正阻止我们的安全环境的一个根本问题。乌云一直以来，大概四年以来一直以开放的思路做安全，核心的法就是说让社区能够通过开放的方式成长，也很高兴今天看到了13岁的白帽子，得有一个好的社区让他学习、引导他。

剑心:　　我觉得是用开放的思路做社区。还有我觉得很多企业，企业是最让人头疼，以前大家认为第一时间不会说，为什么不会说？因为一个企业如果出了安全问题，一定是安全工程师最先知道，他不会反馈给领导，他的领导不会反馈给公司决策层，所以整个企业看起来一片太平，但是我们知道安全，包括今天这么多议题，我们可以看到我们的安全滥到什么程度。

剑心:　　对用户也一样，我们希望以开放的思路在安全上好好做一做。今天峰会也是这个思路，峰会上午是白帽子的专场，下午是企业。我们也看出这很符合实际，上午大家很自由，下午慢慢有点拘泥了，就是总是不止痒的感觉。如果乌云通过这个峰会有一个改变。看到下午有很多人在这儿挖人，我挺反感这个事情的，我看到很多好的人，去了阿里巴巴，在网上不说话了，这个ID不活跃了，一到企业里去可能被非技术人员就给限制了，我觉得这是很不好的，而且企业分享的时候会有一些，当然这不能怪企业本身，它的立场是对的，但是我们不希望这样的方式出现，即使出现大家也是买过票的。

剑心:　　核心的意思，希望大家一起，包括社区，白帽子，乌云企业，一起把这个事情做的更开放一些，长久一些，而不是把两边人挖走。

51CTO直播小组: 2014乌云安全峰会即将开始，敬请期待！
主持人: 我叫肉肉，我和大家一样，也是乌云的白帽子。我是第一次做主持，所以你们不要对我要求太高。首先，感谢大家对乌云峰会的支持，这是乌云第一次举办大型活动，之前有很多，我听多很多小伙伴问乌云怎么没有线下活动？因为太忙了嘛。
主持人: 这是第一次，以后会经常有的。因为是第一次活动，虽然很用心地去准备，但是没有经验，时间上面可能有些匆忙，所以准备的可能还有点不足，还望大家多谅解。这样的情况下，我知道白帽子是赠票活动，虽然要求高，但是很值，因为咱们门票在一周时间抢购一空。
主持人: 今天到场的应该是三百多人，有一百多个人是专门从外地赶过来的，经常有人给我晒机票、门票的。我在这里代表乌云对各位的到来表示热烈的欢迎。
主持人: 感谢大家，这次峰会没有安排任何商业化议题，这是对大家的尊重，我们这次的议题有十一个，可能你会听到很多干货，这次以有第一次登台演讲的嘉宾，我相信他们的表现不会让你们失望的。
主持人: 因为有很多人想要过来，但是因为没有票，所以很遗憾，不过我们也准备了互动屏，不管在场内还是场外都可以参与的，地址在屏幕上面，可以通过手机和电脑访问。
主持人: 在会议开始之前我得提醒大家一下，这是黑客的聚会，大家知道黑客的聚会可能有点危险的，我不知道有没有人注意到外面的易拉宝上面写着“熊出没”？估计你们都忽略了，所以今天上网要格外小心一点哦。
主持人: 会议开始前我们先看一个短片。
主持人: 这短片大家应该能看出来，是对乌云的介绍。之前我统计了一下数据，乌云从2012年7月12日有了第一个注册用户，现在白帽子数量是6214个。注册厂商1827个，这一路走来有很多艰辛，感谢大家一路的陪伴。
主持人: 这个短片可能有点太匆忙了，还不能让大家深入地了解乌云，如果想了解乌云，就是在网上活跃一点，还有今天，今天乌云第一运营官疯狗会在接下来的第一个议题中给大家介绍一下乌云，讲讲乌云的成长，还有它一些理念上的，还有从乌云上沉淀出来的东西分享给大家。
主持人: 请大家欢迎乌云主站负责人疯狗。
乌云主站负责人 : 因为肉肉平时跟我们不是这样的，面对我们几个想说什么说什么，但是今天到场嘉宾太多了，看起来非常非常紧张，所以希望大家理解。
乌云主站负责人 : 我也是看到这么多人，非常非常感谢大家。因为我也是参加过这样的会议，感到很正常了，但是我们自己乌云峰会的时候心里倒是很忐忑。
乌云主站负责人 : 今天到场的嘉宾主要是互联网企业，还有互联网媒体，再有就是平台上的白帽子。还有一个特殊的群体，我也是昨天才知道的，还有一个叫白帽子的家长们。
乌云主站负责人 : 现在计算机技术还有网络普及的非常快，很多的学生接触互联网非常非常早的，新一代做安全网络的非常小，还未成年，所以有的小白帽家里不放心，说你自己大老远跑帝都很混乱的地方不放心，所以爸爸妈妈陪着过来了。我想对各位家长说，你们的孩子很优秀，他们也会是中国未来互联网强大核心的力量。
乌云主站负责人 : 大家知道乌云是第三方机构，大家可以看到今天很应景，白帽子会分享安全研究的心得，企业也会分享一些防守防护的经验，也是双方的经验与智慧的碰撞。我也相信今天的白帽子身价也会飙升三倍，企业演讲之后也会吸引有场下白帽子的加入，一切皆有可能。
乌云主站负责人 : 首先我还是自我介绍一下，我的网络ID，也就是乌云上的昵称叫疯狗。这起了很久了，也没什么意义，就是一个代号。我研究安全也是蛮久的了，我有十余年经验。
乌云主站负责人 : 这十余年并不是向大家炫耀我的技术多么多么，而是我真的很热爱这个行业，我相信今天在座的人把网络安全，信息安全目前，或者曾经是自己非常重要的一部分。
乌云主站负责人 : 插一句，除了上面的介绍外，我不知道是不是只有我一个人，我是唯一把乌云公开漏洞的都看了的人。
乌云主站负责人 : 四年前的夏天，几个小伙伴对外正式地公布了乌云漏洞报告平台，黑客圈或者中国安全网络圈有一些老前辈，后来也是找到我们说，其实我们也想做这样的平台，但是被你们先走了一步。这样也就是说乌云或者说第三方漏洞报告平台就是由我们这一帮年轻人营运，担负了很多期望和责任在肩上。
乌云主站负责人 : 今天也希望用你好，路人甲做话题。因为开始登录乌云的时候都没有帐号，在没有获得帐号之前我们都统称为路人甲。所以很多新加入我们都当做路人甲，所以我希望以这样的文化开始今天的简单的乌云的介绍。
乌云主站负责人 : 很多加入乌云的都会误会我们，因为我们是没有甲方工作经验的，就办了这个平台，其实我想说的是我们甲方经验还是很多的，我们在甲方工作了很多年，发现了种种弊端才出来做这个平台的。
乌云主站负责人 : 我们做的过程中发现了什么问题呢？我以前在新浪网工作，除了日常的发掘自己的漏洞，做防御体系等等外，还有一个重要的事情，我相信在座的很多，比如新参与工作的一些安全从业者们是没有经历过

的，是什么呢？
乌云主站负责人：我们每天还有项重要的工作，就是去看这些白帽子，可能当时叫黑客或者研究者的博客，因为他们会公开一些漏洞信息，而且是在没有修复的情况下，人人看到之后都会去利用。
乌云主站负责人：所以我们天天还要扫各种RS，看看大家有没有报告我们的漏洞，报告了赶紧应急。我们觉得这种模式或者现状体现了一个问题，我们深入研究了这些人心理，他们有时候也并不是很恶意的，他们也想说我发现一个问题，我希望反馈给厂商，得到厂商的重视，同时我的技术也在业界得到反响，交更好的朋友，技术得到更好的提升。
乌云主站负责人：但是那种环境没有乌云的平台，所以大家选择了这种不太友善的方式，所以我们觉得做乌云的平台，漏洞的信息在它有影响的时候，我们报告给厂商，厂商解决后，我们会公开细节，让你在圈子里能分享你的想法，结交更多的朋友。
乌云主站负责人：还有一个，安全群太封闭，大家也知道，安全技术或者说黑客技术稍微走偏一点，可能通过非法手段牟取暴利的方式，一种手段，其实这个不得不承认，现在国内特别特别的普遍。因为它是牟利的手段，攻击者就不希望让其他人跟自己争利，也不希望让企业知道有哪些缺陷，封堵掉，而是长期循环。
乌云主站负责人：所以攻击的手段永远走在我们前面，而我们对黑产一无所知。乌云一直强调说我们要把细节公开，其实也是希望让大家看到系统安全到底是什么样子。我们做着做着就发现了很层次的意义，我们发现受益越来越大，我让安全研究人员通过这个平台学到其他人的技术和技能，让安全研究人员互相认识，也能获得更高荣誉的机会，也让企业了解到自己一个真实的安全的现状，同时我们也让普通用户如何才能保护自己，现在的问题都会出现在哪里。
乌云主站负责人：所以我们是帮企业完成了正像循环，白帽子提供安全漏洞，企业披露安全问题，用户了解到这些安全问题信息，对企业提出安全需求，企业加强自身的安全建设和重视白帽子价值，更多的白帽子会加入到这么一个流程，让这个循环越走越大，创造更多的价值。
乌云主站负责人：刚才肉肉也提到了乌云上有很多很多的企业，我这里简单截取了一些，说明我们乌云为哪些企业提供漏洞信息报告服务的。
乌云主站负责人：厂商我只是随便取了一下，乌云第一页、第二页的企业，其实我们覆盖非常非常广，有互联网媒体、网络的金融、电子商务、社交平台、视频分享、移动营运，还有软件开发商，甚至说安全服务商，等等等等，可以说覆盖了整个中国互联网网络，我们就是为这样的企业提供漏洞信息报告的服务。
乌云主站负责人：刚才肉肉也说了乌云上有很多厂商，大家现在可以看到的大概有570多家，这些厂商都是我刚才提到的这些性质的企业，还有一些看不到的，它们是地方的信息站点，传统的行业，这些我们就没有去显示。我说这五百家的企业注册是这个数字的三倍。
乌云主站负责人：不能说几乎，现在也是已经覆盖了整个中国互联网企业。这是一个活跃的白帽子数量，还有一些白帽子活跃在乌云的其他平台，像内容分享平台上等，实际上注册数量是这个数字的六倍多。
乌云主站负责人：而且这些白帽子来源，他们的角色，个人身份都千奇百怪，首先有专业的安全人员，还有学生、医生、博士、教师、安防人员、企业策划等等，他们来自各个领域，但是他们有一个共同点，对安全充满了兴趣，非常非常喜欢，就是这样一批人集中在一起形成了现在乌云白帽子的群体。
乌云主站负责人：目前，乌云已经接到了七万五千家的漏洞报告的数量，但是这个数量还在飞速地上涨中，我只是取了一下前阵子我统计出的结果。其实这些漏洞对乌云，对企业，对白帽子来说都是无形的财富，几乎所有都有覆盖，我想不到还有什么没有报告提交的。
乌云主站负责人：所以乌云接下来要做的也是积极分析这些漏洞报告中有价值的信息，反馈给各个渠道，像企业、用户，甚至白帽子，媒体，等等等等。
乌云主站负责人：大家也会看到过各种各样的漏洞风险的统计图，其实这种也确实很有意义，大家可以从中得到一些辅助的参考作用，乌云的风险统计图是非常非常有价值，它的价值在哪里？它的统计是以实际发生案例统计出来的，并不是我们拍脑袋共享出来的，或者通过数据计算出来，我们就是通过切实发生的案例来统计的。
乌云主站负责人：这个风险统计图我们认为是当今互联网一个真正的现状，可以看到前五条基本都是安全运维还有管理上的，最终撞库太多太多，很好地体现，像好莱坞的艳照门，这都能体现出来，这是现在最广也是我们迫切解决的问题。而且安全的影响也从传统的开发者慢慢转移到了安全的运维者和管理者上面，有这么一个转变。
乌云主站负责人：乌云在这四年的运营当中也是见过了很多的漏洞，我们也产生了漏洞的 非常中肯的评估方案，这个都在乌云的安全中心，大家可以去查看，这也会是我们今后向外界输出的重点。
乌云主站负责人：我们首先认为漏洞有两个方面具体的影响，第一，威胁的角色是谁，一个，难度怎么样，是不是很容易被人重现？说这个世界上一个人能利用，还是世界上一万个人能利用。还有一个，就是数据的影响。
乌云主站负责人：就是这个漏洞具体影响到我们业务，比较敏感信息，就是这个数据的范围多大，还有这个系统

乌云主站负责人：就是这个漏洞具体影响到我们业务，比较敏感信息，就是这个数据的范围多大，还有这个系统的重要性，看似很简单，这都是乌云这四年来慢慢沉淀分析下来的，也跟曾经的安全机构、组织所说的理念有不同的地方。
乌云主站负责人：我们乌云的想法，我们想把安全圈子变成什么样子？大家都知道有SOHO工作的模式，就是在家里工作，我想我们乌云叫乌HO，我觉得大理非常安逸，空气都很安逸，你稍微积极一些都不被人接受，这是非常舒服的地方，像我真辞去一切，去那儿待一阵子。大家知道这个在国内很不现实，在国内国民生活压力很大的，我们不能放弃一些东西去追寻我们理想国这种状态，所以我们就希望能把乌云，起码在白帽子领域做到什么呢？
乌云主站负责人：我们可以游山玩水，去哪里玩儿就去哪里玩儿，没钱了，打开电脑只用不到一个小时的时间，参与了一些项目，可能拿到一个月的开销，我就可以轻轻松松去做一些想要的生活，很理想，但这说明了我们的理想和目的。
乌云主站负责人：很多企业跟我们说我们报告平台是被动的搜集平台，但有时候我想主动地杜绝我的信息安全风险，比如一个重要的产品上线，我想主动地知道我面临的风险，并且告知我怎么修复，这就是我们的理念，主动为企业提供安全的服务。
乌云主站负责人：另外，也希望给白帽子一个合法收入的机会。因为现在做黑色产业的人非常非常多，他们真想这么做吗？也是被社会，被生活所逼迫。我敢打赌，如果有一个合理的收入，没有人愿意赚这个黑钱。所以我希望这样能让白帽子从一个合理的方向去赚取自己应得的报酬。
乌云主站负责人：很多白帽子还会问很多新来或者老白帽，或者还不是白帽子想成为白帽子的，来到乌云能得到什么？
乌云主站负责人：我觉得大家可以在上面用一个合法的渠道，能谋取一些合法的收入，同时还能以一个正确的渠道展现自己，也能得到思想比较开放、前卫的企业认可，还可以认识朋友，通过交流得到提升。我们的目标想法就是这么简单，这都是说的很实际的一些。
乌云主站负责人：上面我介绍了一下乌云过去、现在、将来的大致方向，接下来我还是把时间交给顶尖的白帽子和企业，让他们来与大家分享。
乌云主站负责人：刚才肉肉提到场外有“熊出没注意”的牌子。这个想说，在现在的环境，手机等通信都是不安全的，但是我们通过报道和比较虚的东西让大家看到这个，我们觉得很不可靠，所以我们想在现场做一个互动，让大家亲身参与其中，是怎样的惊心动魄的体验，这样我想大家会更刻骨铭心一些。
乌云主站负责人：大家知道海底捞是很著名的火锅饭店，其实一个火锅你一筷子下去也不知道捞出什么，我们“场里捞”其实能捞到什么我也不知道。乌云向大家保证，如果你对测试的行为敏感或者反感，请注意你们的平板电脑、手机之类的不要传递敏感信息。大部分数据我们还是会马赛克给大家盖住的，重在参与。
乌云主站负责人：最后，如果在场的嘉宾需要帮助的话，我可以在现场对大家进行支援，也可以在屏幕上反馈一些问题，我们工作人员会帮助大家处理，之后，我们的平时工作交流上的可以通过QQ和邮箱，我主要从事乌云报告平台运维工作。谢谢大家。
主持人: 谢狗哥介绍，听完之后，我相信对乌云不怎么了解的也有了一定的认识，狗哥在上面总结乌云2014风险统计图，现在风险标准挺多，但是最适合咱们的可能还得是从乌云上面沉淀下来的，所以在座的企业朋友也可以回去思考一下，自己什么地方没做好的，还需要再改进一下的。
主持人: 接下来进入正式议题，第一个是关于智能家居的。智能家居我记得提出的时候，就已经在两年前开始提出了，但是真正火起来的话也就是这一段时间，一个东西火起来的时候很多人会好奇，有些人好奇它怎么用，有些人可能想着怎么做，还有的人好奇点就是这个东西安不安全？我们乌云一个小伙伴对智能家居方面很着迷，他就是乌云的白帽子Livers。
Livers: 大家好。第一次和大家见面心情非常紧张，我先说一下我自己。我之前主要专业还是搞Web的，我一直做乙方的工作，大约是两年前，业余的时候开始研究一些可编程的路由器，就一发而不可收拾，特别喜欢智能上的设备，但是研究的也不够多，大家知道后来替补上来的，有一些东西没准备好，时间太匆忙，如果后面有讲的不好的地方，请大家多多谅解。
Livers: 这里先给大家看一个视频，这是一款智能插座，它有一个手机控制单可以控制开关，但是我发现了它的协议，发现里面是有问题的，所以我就用就简单地模拟了一下它的收发，这两个是在同一个网络上的，就是家里远程可以控制我的插座。
Livers: 我先介绍一下背景，可能有些人对智能家居还不是很了解，智能家居很早就进入中国市场了，大约有十年的时间了，但一直不温不火，直到这两年才突然一下子迸发出来，而且出现了不少比较好玩儿的事，以前都是智能家居成套的系统，现在有一些智能家居的单体，像之前出了一些智能路由器，还有一些智能的插座，深受极客的喜爱，销量非常火。

Livers: 但是纵观全局来看，传统的智能家居占有非常大的市场份额，我相信未来肯定会有越来越多的智能家居的产品出现，它形态也会越来越丰富，我也很期待。
Livers: 我简单介绍一下智能家居整体框架，大家可以看到它中间有一个中心的部分，叫中控梯，它相当于枢纽，它控制了视频开关，还有红外的一些装置，还有基于Wifi的装置，它会和路由器进行交互，再通过手机终端或者电脑来控制。
Livers: 但是按照目前的形式，渐渐的觉得中控梯太臃肿，就把中控梯和路由器做到一块，小米马上就上一些，把网关和路由器结合卖的路由器。
Livers: 但是我是觉得，它们太过于追求创新，技术上的一些创新，对于产品的安全性考虑的不太周到，出现了一些安全问题。按照剑心的要求，他让我写一个技术框架，就整理了一套，把之前做的经验整理出来。
Livers: 智能家居，我这边研究的主要是两大块：一块，智能路由器这部分；另一块，智能家居的配件。路由器这部分我主要是从它的固件，Web接口和云端的交互等去研究的，配件我主要分析了它的通信协议，还有客户端，包括它的云端控制，当然，很普通的例子就不讲了，我讲一些我碰到的坑儿，比较好玩儿的事。
Livers: 可能有些智能设备的厂商，防止代码不被外漏，或者保护的作用，它会把自己的固件做加密。这里我看到了360智能路由器，我下载了它的固件，它是加密的，后来找到了它一个非常老的版本，就是未加密的，我把这个版本提取出来，搜索它里面可能很固件更新有关的东西，我发现在它的设计页面里有一个（阿不路）的函数，我反汇编它的程序，它里面硬编码了一个BUS文件，这个BUS文件调用了另外一个，这个BUS调用了解压固件的。
Livers: 我就反汇编这个，大家可以看到图上，有一个非常明显的加密的密钥，其实就是被类似解密出来，可以直接当做AESK传进来，我又跟着程序往下走，发现它又调用了RSA的加密，上面可以看到一个非常大的整数串，这就是RSA的密钥，就是科普一下小知识，片入式的系统，为了更精细化没有采用OPELL，如果对照里面加密库的话，就是参照了IPI。
Livers: 我又顺着程序往下走，把程序完全理出来，它整体就是这样，先读取了128字节，用RSA解密这128字节，剩下的就是用AS，每次循环4096次节，进行解密，前面它会有一个版本限期，我图上标注了，上面13，然后是长度，把这个去掉，然后做RMB5验证，如果相同就刷入固件，如果不相同，就把解密出来的东西删除，这样的话我们就可以完全模拟程序，写一套自己的解密程序，来发现它固件里面的各种问题。
Livers: 另一种比较好玩儿的，就是变形压缩的东西，这里小路路由，我后来又看了一下小路路由，它是内部有一些不常规的压缩，你用普通工具是无法把它踢出来，这就需要用精彩文件分析的功力。当你解到这三层，你会发现它有一个（插队）的压缩包，它以XZ开头，以YZ结尾，你去掉多余的代码才能把它顺利地解压出来。
Livers: 前面我是从软件的部分来讲，像我这样的无业游民，对一些设备不可能花很多的财力去购买，更多的是自己从网上下载一个固件去分析，但是如果你有能力购买一些硬件的话，你会发现硬件的（漏洞）更多。
Livers: 像智能路由器经常用的OTW系统，它本身的串口会够你一个ROOT终端，你直接用TTL转USB，就可以直接控制了你整个的路由器。除了这个串口，还有像GTAK口，这个完全可以调节GPU，像有些不需要加密的，你就可以用GTAK口去破解它的加密，也可以读写它的里面的Flash内容，把文件直接提取出来。当然还有更暴力的，可以直接把Flash拆除下来。
Livers: 说完了固件我再说说接口，之前提路由，在乌云也测过，它里面的版本就是被自身变异了，变的没法儿看，相当于提高了一个难度。从网上搜索也搜索不到任何这种反变异程序。我自己尝试了一下，发现了可以把变异过的程序反变异过来。
Livers: 我看了一下官方的PACH，官方的PACH打的LNO，嵌入式对浮点数的运算非常处理的，所以它本身会把浮点数转换成整形。LUWA本身不到200K，这时候我就把π值π出来，可以反变异出LUWA的代码。后面我会把工具分享给大家。
Livers: 还有比较特殊的情况，有些不是把代码加密，它把代码给隐藏了，这里我看了一下华为的智能路由器，它本身的Web目录是空的，我就觉得很神奇，它怎么没有Web的接口？
Livers: 我提取了它的固件，找到它的Web调用相关的程序，我反变异了它Web，发现的一个很神奇的配置，它里面保存了所有的文件的名称、大小还有偏移，直接从Web值里给提取出来，这样我们就能进行Web上的审计。
Livers: 说到审计，尤其像LUWac这样的程序，可能之前不是经常用，很多程序员在用的时候不太熟悉，往往会犯一些非常低级的错误，像直接读取，加到字符串中直接执行。
Livers: 这里我就把它全部整理了一下，这里就说一下接口认证，它的接口认证比较奇特的，Luwas节点分多层次的，上层节点如果设置了就需要登录验证了，如果没有设置就没有登录验证的环节。
Livers: 如果上层节点设置，下层节点也不会登录验证。还有从弹出声明，官方是四个参数，但是后面加了一

个，当有第五个的时候就可以直接访问。
Livers: 后面我又研究了一下其他的，就是前一段时间刚出来的华为荣耀立方，大约这个固件是八月二十几号的版本，我把它提取出来，然后查找里面的问题，我这里碰见了一个非常神奇的地方，在它ETC目录下有一个PPT文件，这个文件可以任意文件上传，我又看了它LAT符，这样的话如果能访问到的话，就可以上传一个，通过系统的命令安装一些APK，可以做一些非常好玩儿的事。
Livers: 对于智能路由器，业内人士的话比较喜欢它的各种APP的插件，也是它的一个比较大的卖点，但是由于这些插件可能是第三方开发的，或者自己又重新做的，所以在一些配置上没有做好处理的。
Livers: 像这个非常常用的一个离线下载的插件，它本身没有登录验证，很多人直接拿来用了，包括一些大的厂商，这样的话如果对外网开放的话，你可以直接远程查勘它下载了什么，你也可以帮它下载一些东西。有的是硬编码直接在里面，也是非常奇葩。
Livers: 前面说了这些路由器可能有密钥执行或者什么，大家可能只知道我能拿到控制权限，但到底我们拿到了它的权限之后，我们能做什么好玩儿的事或者邪恶的事？
Livers: 按照我的经验想了一些，比如你拿到它的权限之后，你可以交叉变异，留一个后门，可以是免登录的SSH，还可以是后门的Web的万能密码，或者是像之前一些路由器的漏洞，把DNS的服务导向一些钓鱼网站，甚至流量去分析偷取一些帐号密码等。
Livers: 前面都是一些传统的想法，当智能路由器和智能家居设备结合的时候，我们就不只可以控制它的路由器，我们还可以控制它的家用设备，比如说攻破对方的安防系统，偷拍声音、图像，或者我们自己搞一些非常有趣的视频，我这里就做了一个视频。
Livers: 前面可以看到我一些东西用了不少的反汇编，现在嵌入式设备基本上都是MIPS构架，这个在网上的资料比较少，所以我总结了一些比较有特点的地方，比如它的寻址，它会有一个GK的计算器，这个是保持静态变量的寻址，本来我也想去搞一些溢出方面的，我发现它的溢出比X86下还要麻烦，X86本身参数可以直接压入对站，MIPS是完全通过计算器完成的，所以调整它的对站相对麻烦一些。
Livers: 前面讲了我对路由器的小小的研究。下面我讲一些智能家居配件协议上的分析。很多传统的智能家居配件会基于315还有433频段，通过262的协议进行传输。如果淘宝去搜一下，几乎80%都是这样的配件，但是这个安全性非常差，算是上时代的产物了。
Livers: 这里我为了让大家看一下，我也做了类似的视频。这里通过手机控制端控制一个遥控门铃，通过控制我的电灯的开关，我自己DIY了一些工具，用了它的收发器，还有发射装置，用PRS的单片写了一些262解码程序，我可以直接SNIF它的信号，然后点击。
Livers: 当然这些东西完全可以用工具直接按，非常简单的完成，但是这个价格不菲，我自己搞了一套，成本大概也就十块钱，就可以直接进行收发，由于低成本的话，这样的话我们就可以做更大范围的事，因为它的传送范围是非常远的，大约有十米左右，所以我可以做很多，放在每个楼层，通过SNIF把抓取的数据存储到单片机里，最终传到我的电脑上，这样的话可以远程控制一栋大楼的智能设备，就是这种比较简单的智能开关。
Livers: 大部分315、433的设备，基本上都是用的262的芯片，它发的基本上都是固定码和硬编码的一些东西。除了315还有433智能家居还常用一些红外，红外就是非常简单的，像控制电视机，空调，但是红外是多信号，无法穿墙，所以它经常做转发，像前面讲的315、433等进行转发，这样的话我们就可以间接地再来控制315或者Wifi的东西来控制红外，我可以远程地帮他调高空调，或者远程地帮他打开电视换频道。
Livers: 前面都是讲的一些比较传统的，可能稍微有些科普了，这里我讲一些最近比较火的一些，主要是Wifi上的一些东西。传统的Wifi，就是物联网主要基于AP模式去，那些是各种包括传统的破解密码就可以了。
Livers: 但是后面一些单品延用了一些新的技术，叫做闪联技术。我之前买过一些闪联设备的插座，我第一次配置的时候完全震惊到了，我直接输入Wifi就可以直接连入互联网，我当时还奇怪，我用的苹果，它怎么可以控制我切换网络呢？
Livers: 所以我就觉得这个东西非常好玩儿，然后去研究它。等我理清了它的传输方法之后，我完全被它震撼到了。
Livers: 它的过程其实比较简单，它的芯片刚开始的时候，它把它的网卡设为混杂模式，然后监听网络中所有的报文，像手机端配置帐号还有密码，它可以通过自定义算法把它编到数据包中，它接收了这些直接通过长度的算法，可以把传送的帐号还有密码给解码出来。
Livers: 所以这个就非常好玩儿了，攻击者可以同样的也开一个网卡监听，掌握编码规则，就是你怎么解码的我也怎么解码，我可以在旁边SNIF解密到你加的Wifi帐号密码，它的编码稍微麻烦，都是易位操作，来减少CPU的预算。
Livers: 这个问题之前老外研究者也提到过，他们做了一些收复方案，再传输的时候用UES加密然后再传输，但是AESK是直接硬编码到芯片里面的，所以用户输的时候每次都要输比较长的AESK，所以比以前更麻烦了。

是ALCK是直接硬编码到芯片里面的，所以用户输的时候每次都要输比较长的ALCK。所以比以前更麻烦了。
Livers: 前面是传输密钥，到了后面有一些其他的东西，就是通过Wifi进行传递。我又研究了一下它的插座，它的插座协议做的太不安全了，我可以直接伪造它的协议进行重放。
Livers: 我分析了一下它的协议，它发了ulink数据报，然后指令，后面是它的六个字节，再到后面是Link空格，最后是操作指令，然后是开关，它也会进行验证，就是发送一个本身的密码，是默认的888，这样的话攻击者完全可以远程控制它的插座，就是理解了它的流程。
Livers: 攻击者伪造它的MIKE，加密码发送到云端，云端再回击攻击者一个包，攻击者再发射到云端，就可以控制你的设备。
Livers: 后面是蓝牙快速融入到物联网当中，它也推出了4.0，就是低速率的传输协议，但是因为安卓好像是4.3支持了这个协议，但是对协议站的实现却存在着一些缺陷，大家可以看到蓝牙连接设备往主机发送通知包的时候没有验证LES长度，结果导致了溢出。
Livers: 我之前写过类似这样的程序，我之前接收一些游戏的数据包，大部分游戏数据包是四个字体，但是突然发一个空的过来，我的整个对站都跑飞了，调了一天才调好，所以这就非常感同身受。
Livers: 再者，讲一些国外经常用的频段协议。在美国可能就是有80%都用的ZWAVE，它在密钥传输的时候有问题，它分了三个层：路由层、安全层和应用层。路由层在密钥传输的时候有缺陷，就是中间人可以直接劫持，大家可以看这里，这相当于协议来回收发的过程包。
Livers: 它的设备为了组网方便，会在内部初始化一个16个0的初始密钥，但这个密钥，每个设备都一样，所以我在和设备建立加密通信的时候，中间人可以劫持两端，然后用初始化密钥把它发送的K给解密出来，然后通过这个K解密到它的一些通信。
Livers: 之前Blackset上也有人发现，就是去年发现了一个实例，它和前面不太一样，它的设备在恢复出厂模式的情况下，它没有检查当前的密钥就直接恢复了，导致攻击者可以伪造密钥建立连接。
Livers: 最后讲一下云端方面。这个我在乌云上也提交了，就是云端和客户端两个协议相结合，综合利用可能会造成更大规模的事件。它的云端存在一些Web上的漏洞，可以直接控制到权限，而且它的权限比较大，是直接Root权限。
Livers:
Livers: 我前面把协议分析清楚，有很多东西就可以远程控制它的设备。它的云端几乎就把设备上所有的信息全部存储，而且云端上还有DIBAT模式，它可以直接升级固件，甚至黑客可以直接编一个固件，然后升级到你的设备当中，而一直远程监控你。
Livers: 可能这次题目选的不好，选的太大，很多细节的东西没法儿仔细去讲，前面的演讲就大致这些。这里我总结一下，传统的类似315、433的设备，安全性非常差的，越来越会出现这个设备的单体，它的安全问题会日益改善，像闪联的技术更多的采用到新的设备当中，我希望相关的一些厂商在追求新技术的同时，也要兼顾下安全。
Livers: 最后，向大家宣传一下，希望大家多多关注乌云等等。谢谢。
主持人: 谢谢。我看他的短片里直接有劫持监控的录像，那个大家应该可以看懂吧，这个危害还是挺大的。我还听到他说他可以控制整栋大楼的智能设备，所以议题对大家来说有点生疏。下面进入茶歇时间。
乌云主站负责人 : 我中间跟一位朋友沟通，他新买的房子，家居进入的时候已经智能化了，所以我们在峰会上抛出的一个隐患，家庭安全上的风险。因为有些人没懂“场里捞”发的意义是什么？它由两部分构成：一个是Wifi信息的劫持，一个伪造的Wifi，另一部分是GSM协议的缺陷，导致短信的嗅探。
乌云主站负责人 : 大家可以在屏幕上看到现场互动的数据，比如登录微博一些网页的帐号密码，还有一些我们刷微博或者看网页时传出的信息。还有大家看到了一些短信，这个短信，大家也知道，十一快到了，买票，这个可能会看到大家购票的短信。
主持人: 大家老是听Wifi不安全什么，所以今天就是带大家感受一下。
主持人: 接下来进入下一个议题，由乌云的白帽子，就是大家熟悉的二哥来跟大家分享。
Gainover: 我今天讲的东西，XSS是一个非常古老的漏洞，但是在互联网里还是普遍存在的。我今天给大家看一看，为什么说它普遍存在，是存在什么样的程度？大家看到这个标题，一个被忽略的漏洞，到XSS僵尸网络。看似很严肃的标题。
Gainover: 我实际上是一个不严肃的人，我是一个喜欢卖萌的人。我来过两次北京，上次路过北京天安门，但是我是在天安门地铁下面，所以我天安门也没看到就回去了，我来是为了一个会议，这个会议叫做中国细胞生物学大会。
Gainover: 大家认识我的以为我是搞计算机的，实际上我是一个生物研究工作者。当然，你不要问我，转基因到底有没有危害？这个其实我也不是很好明确地回答你们。但是你们可以问我，XSS漏洞是什么？

Gainover: 这个东西我不会多讲，我可以简单地说一下，当你浏览一个正常页面的时候，如果这个正常页面能够被攻击者在里面插入恶意代码，来执行他们目的的话，我们可以说这个页面存在XSS漏洞。
Gainover: 这个可以是网站本身的代码问题，也可以是用来渲染这个浏览器本身存在的问题，都可以导致XSS漏洞。什么是XSS漏洞挖掘呢？就是你想办法把网页插到别人的网页中去。今天的关注点在XSS漏洞利用上，这个在乌云上有很多案例了，大家平时用的最多的，用XSS来窃取客户的Cookies，这个在乌云推出来一个XSS漏洞利用平台，这个攻击实际上被推到了顶峰。
Gainover: 后来有一些厂商，百度后来把这个机制加上，用来缓解XSS漏洞利用方式。乌云还有一个导致XSS漏洞利用，叫XSS盲导。这点由于简单粗暴的方式，也导致大家把这个发挥到极致了。还有三类，我用红色图标给标出来的，也是将XSS危害发挥到极致的方式了。
Gainover: 大家听说过XSS是蠕虫，它会放大的，有了XSS蠕虫，像微博，百度贴吧，都发生过这个蠕虫案例的。我在这里说一下，黑客可能发一条微博，这个链接里会有恶意代码，当客户点击黑客发布的内容或者链接之后的话，这个代码会让用户自己也发条微博，这里同样也会发这个恶意代码。
Gainover: 这样的话，含有恶意代码的微博会呈现几何级增长的，所以可以在很短时间内得到很大的爆发量，这对社交网络来说是危害很大的攻击方式。
Gainover: 另外，XSS分布式服务攻击，这个一直是停留在概念上的东西，这段时间有一个实际案例，就是利用搜狐视频漏洞进行攻击，这是老外先报出来的，国外媒体也纷纷报道了。
Gainover: 这个实际上，黑客在大的网站，像搜狐视频网站里去插入这样一个恶意代码，当用户访问大流量网站，比如在观看视频的时候，用户浏览器会自动执行黑客命令，黑客命令可能会向他所要攻击的目标网站，一秒钟发出请求，搜狐有十万客户观看这个视频的话，就呈几何级的增长，这样使目标网站停止服务了。
Gainover: 这个案例也是我们团队发在乌云上的一个案例，这叫水坑攻击，这个在XSS，我们现在把这个概念结合起来了，恩为什么叫水坑攻击？就是有时候我们要去攻击某个目标的话，实际上是你很难接触到它的，在这个时间里，比如说李开复，我们想去给李开复发一个链接，都很难直接发的，我们不知道他的联系方式什么的，想直接攻击他是不可能的，但是我们是可以知道李开复会去上36k的，我们可以在这里植入XSS，在我们等四天之后，李开复中了我们XSS，我们当时结合QQ的漏洞，就是让李开复关注了乌云漏洞报告平台，在这个过程中不光李开复受到影响了，还有其他的，只要经常上36K的，36K只是其中一个网站，我们还会在很多网站上插入这个XSS。为什么叫水坑攻击？
Gainover: 比如你在大草原的话，比如说我是狮子的话，我想找猎物，但是大草原太大了，我哪里去找呢？但是动物都喝水，我就蹲在水坑等着，等动物来喝水我就把它捉到了。
Gainover: 大家要注意这个漏洞有一个特点，第一个，实际上我们在36K插入XSS的话，实际上并不是36K本身网站的漏洞，它本身网站我们也没去找，是因为36K在当时用了第三方的评论插件，我忘了叫什么名字了，这个插件本身是有漏洞，我们可以往里插恶意代码，这样的话在36K在第三方评论插件劫持了QQ漏洞，然后使李开复关注了乌云漏洞平台。
Gainover: 实际上36K对我们一点用没有用，我们要劫持的是QQ的信息。
Gainover: 虽然说我前面讲了XSS利用方式有这些，可能还有其他方式，我这里没有提及。厂商和攻击者对XSS分别是什么样的态度呢？实际上XSS漏洞相比传统的，它的危害确实要小很多，对于一些厂商来说的话，特别是国内的一些厂商，这种类型的漏洞就像牛皮癣一样，清都清不完，它一直存在。厂商有的时候要么修，要么有的觉得域名不重要就不修了。
Gainover: 对攻击者来说，实际上XSS可以干很多事情，首先，可以获取个人信息，这个待会儿我要给大家演示的。
Gainover: 第二个，XSS是用来伪造钓鱼页面，实际上我平时身边也有人在淘宝上买东西，别人告诉他退款，让他输入银行帐号密码什么的，可能前面装的很像，但是容易露馅的就是发来乱七八糟的域名，就让人警觉。如果配合XSS漏洞的话，在域名或者页面上很难判断是页面自己的还是黑客构造出来的，这对用户来说很难区分出来的。
Gainover: 这个黑色产业已经在这样做了，在乌云上有一个实际案例，就是利用搜狗拼音来钓鱼的案例。
Gainover: 现在是应题了，叫一个被忽略的漏洞。这个漏洞是在2013年发布出来的，叫优库分站一个存储型XSS漏洞。这个漏洞是被优酷忽略的，这个漏洞不光危害是高，而且很高。这个是发漏洞的一个截图证明，在这上面确实很难看出来危害，无法就是优酷的一个视频里面有一个XSS，可以弹出这样一个框框，就是不能直接看出它的危害。
Gainover: 后来这个危害还被忽略了，也公开了，包括我现在还在讲，这个漏洞还没被修。为什么会被忽略，这个是我不能理解的。首先我猜，难道是因为存在缺陷的域名不是优酷自己的域名。

Gainover: 实际上，XSS影响域名的话，大家可以看到，它确实是优酷.com，至少从影响的域名来看的话是应该被修的，而不是被忽略的漏洞。第二，哪怕你是第三方文件导致自身的域名，不是你自己文件下的，为什么没有通知第三方来修，这也是我不能理解的。漏洞被忽略之后会有什么后果？
Gainover: 现在我们就一起来看一下，这个忽略了一年多的漏洞到底是什么后果呢？大家看到我打开优酷，这个说明首页上就可以执行恶意代码，不光是优酷，土豆也一样，这是我测试的时候的提示，酷6也是一样，这是曾经出过漏洞的搜狐视频，也是一样，也就是说，用户只要曾经访问过我们恶意构造的页面，之后不管他看哪个页面都会执行我们恶意代码的。也就是说，这些视频网站都受影响的。
Gainover: 问题一，到底有多少个网站会受到这个漏洞的影响？我们看一下，我们在某些网站上，就像我刚才在36K评论一样，在上面插一些XSS，用户在路过36K的时候就会被我们感染了刚才那个漏洞代码，接下来他去看优酷视频或者其他视频网站的时候，就会执行我们这个恶意代码。
Gainover: 大家知道，有很多网站会调用优酷网或者其他网站，这样它们也是很大的受害群体。为了研究到底有多少网站受到漏洞影响？我们是做这样的实验，我和我们团队的人一起来做这样的测试，我们找了一些网站，这个网站被我们拿来做测试，表示抱歉，但这个也没办法。
Gainover: 我们在这里插入XSS五之后，我们开始看这个，它们被我们感染恶意代码，它们访问用户的时候或者其他网站的时候，我们恶意代码会让它们去调用我们一个我们指定的网站，在访问日志里会记录这些用户在哪些网站访问到我们的。我们统计结果是什么样的？
Gainover: 这是在一段时间内统计的，我们一共得到了2.39G的访问日志，所有的受害者向我们指定的发送了9513830次请求，我们在请求里访问来源以后一共有2831个域名受影响。我们取了前20名，第一名是优酷，第二个是车的网站，SCAR，包括360也受牵连了在里面，还有其他的，还有我没见过域名的网站，这些都是被统计出来的。
Gainover: 问题二，对XSS不了解或者不理解它的危害，可能看我之前的视频，弹个OK，弹个OK，到底有什么用呢？
Gainover: 我给大家看一下，怎么从OK，到真实数据的泄漏，会泄漏什么数据，黑客会获得什么数据，我们看一个视频，大家在生活中，对普通网民来说用的就是QQ，大家也比较关心QQ上面的数据，而且了解我的，知道我喜欢黑腾讯，所以我来拿它做演示。
Gainover: 现在黑客把代码改成真实的代码了，我就像刚才那样打开页面，我正在看QQ空间，这个是大家正常浏览，但是没有想到的是内容更新了，然后我去上优酷网，大家可以看到这个时候恶意代码已经被执行了，这都是模拟大家在做的操作，最后我们来看看黑客这边得到了什么？
Gainover: 大家可以看到，我用自己QQ做的测试，这里访问了优酷之后，你QQ所有好友被发送到黑客这边，这个只是拿好友数据做掩饰，可能其他数据也会因此而泄漏。
Gainover: 我们拿优酷做演示，就是你曾经访问了我恶意构造的网页后，你去看某些视频网站，你的信息都是会被发送到我这边的，而且更关键的是，这个东西只要你不清理它，它一直存在在你电脑上的，总有一天你会上钩的。
Gainover: 就这样完了吗？当然不止这些危害，这个不仅仅是个漏洞，这是我2013年10月24号我在漏洞下面做的一个评论，我是这么说的，问题很隐蔽，这个开发人员肯定是注意不到的，只要是有这么用的，估计都会存在问题。这是当时我在漏洞下面做的评论。
Gainover: 后来的话，我看还存在哪些问题？我们看一下乌云已经公开的案例，实际上这三个漏洞的话，与我多少都有点关系，不管是马甲还是间接报告，还是怎么样的，实际上我发了这个之后还是没有引起太多各大企业的注意。第一，全部博客可留后门。
Gainover: 你只要你是新浪博客用户，不管你以后访问新浪博客任意文件，都会执行我这个恶意代码的。第二个，淘宝支付宝的，这个叫一个可大规模悄无声息窃取淘宝支付宝帐号的。这个标题看起来很大，但是实际上一点都不危言耸听的。
Gainover: 这些企业不知道自查周期比较长还是怎么样的，同样的漏洞在阿里云服务上一样存在的，后来也被报告修复了。现在我看了一下，淘宝相关的页面上，同等类型漏洞都已经做了相应保护措施了，他们已经意识到这个漏洞危害的严重性了。
Gainover: 其实我今天给大家讲的，就是说漏洞这块东西，就说到这里。我标题还有后半部分，叫XSS僵尸网络。这个东西敢这么说的话，是有一定漏洞基础的，是因为这个漏洞本身特点决定，它是可以用来做这个东西的。首先，隐蔽性。
Gainover: 为什么说它隐蔽？这个漏洞本身特点，你想想，对于一个用户来说，可能很多用户对钓鱼网站网址都分不清楚的，你让他在页面来看原代码，看它上面是不是存在恶意代码，这是根本没可能的。
Gainover: 第二个，传统的杀毒软件，它是很难去防这类攻击的，这个东西对于软件来说，至少很难有办法去区

Gainover: 第二个，传统的杀毒软件，它是很难去防这类攻击的，这个东西对于软件来说，至少很难有办法去区分，是XSS，还是这个网站本身的。所以对用户这一层的话，用户是很难去发觉他们是已经中招的了。
Gainover: 哪怕我自己搞这方面的，如果不专门看的话，我自己很难察觉，所以我也自己也很担心这类漏洞。这个对厂商来说，这与传统的漏洞不太一样的，传统的是在页面里插一段代码，你接到报告说这个页面有问题，厂商查的话，发现这里黑客插了一段代码。
Gainover: 但是对XSS来说，这个漏洞是存在用户电脑本地存储上的，用户觉得页面有问题，让厂商去查，厂商去看页面完全没有任何问题，然后就不再过去了。这样的话不管是最用户还是厂商来说，这种类型的漏洞隐蔽性是特别好的。
Gainover: 第二个，持久性。这是一个长期存在用户电脑上的，这个漏洞原理也是发过的，大家可以看看，它用Flash本地存储，如果你不是自己去清Flash Cookies的话是很难清掉的。
Gainover: 我记得360软件Flash Cookies是勾上的，以前我看安全卫士软件的话，Flash Cookies是不勾上的，就是用户清理垃圾什么的，不会清理掉这些垃圾，这样的话，这个东西就是在用户电脑上长期存在。
Gainover: 另外的话，那些软件去来清理这些东西，它就是找到浏览器默认的Flash Cookies存放目录对它清除。我现在不知道第三方有没有清除这个，如果没有我建议以后可以做上。如果没有的话，这个后果很严重的，按照我前面演示，如果这次没有打开网页或者QQ空间不会被窃取，但总有一天你会被窃取。
Gainover: 第三，流量大。大家可以看到这种类型的漏洞影响到了，基本上是全国所有视频网站，当然有些网站没有用那个代码的话就没有。
Gainover: 第二个，电商类的也是受影响的，虽然现在都修了。第三个，新浪微博也是受影响的，这也是大流量的社交网站。所以说这个受影响的用户量是非常大的。
Gainover: 其次，用户页面时间停留长，大家可以看到，特别是对于视频网站来说，它的用户基本上都在看视频，一个小时、两个小时，恶意代码就会在那里被执行一个小时、两个小时，这样黑客利用代码可以获得很长执行时间的，这为后面的攻击是能够提供很大的帮助的。
Gainover: 最后，基于这些特点的话，这跟前面原理差不多的，我们可以看到，黑客首先做一个感染阶段，它首先找一些比较好入侵的，或者有漏洞的中小型的网站去入侵，入侵了之后就在里面插入这样一个XSS代码，因为这些中小型网站它在上面用户Cookies有什么用啊，一点用没有，所以黑客不会偷这个，它的目的是植入后续的XSS，接着这些FLash XSS在感染到足够用户数量之后，我们可以让FLash XSS执行黑客的操作，最简单的就是信息窃取，当感染数量达到一百万的时候，我假设里面只有1%的用户，比方说他先看QQ空间，再看网站的话，就会有一万人的数据窃取，这可能只是一天的数据，如果是一个月呢？一年呢？
Gainover: 因为这个漏洞不宜被发现，不易被清理，所以这个积累到一年的话危害是非常大的。而且这些受害者，在看视频的时候，他的浏览器可以被黑客做一些事情的。当然，这个还会有其他的利用方式，这个就要发挥想象力了。所以我这里总结了一下，这种类型的做僵尸网络的话，首先感染阶段，然后是执行阶段，从危害上来说的话，我觉得影响几乎全国所有网站的。下面可以进行提问。
主持人: 二哥讲的XSS很好，由浅入深，看视频的时候，我看大家都很激动，说XSS的话我也算跟XSS挺有缘，也一年也是PPTV团队让我配合做一个视频，其实说真的，他当时其实告诉我了，如果他不告诉我，我不会觉得那个有问题，是一个QQ的，那会儿就会有很多人嘲笑我，你又被钓鱼了怎么了。
主持人: XSS可能你自己有些厂商对不怎么熟悉，自己看的话看不到什么影响，觉得它就是一个小问题，不值得修补什么的，但是实际上二哥讲的，他讲的一个案例，就是之前在乌云上被忽略的一个漏洞，因为乌云上漏洞都是会先通知厂商，让厂商去修补，但是厂商可能没注意，导致一大批视频网站也跟着被连累，所以以后还是得注意一下，不管什么样的漏洞，漏洞的价值不是它的技术点或者一些其他东西，而是怎么去利用，感谢二哥的分享。
乌云主站负责人 : 今天上午的议题是很巧妙的安排，我们知道互联网有两大入口：第一是PC，PC上的浏览器，这个就是二哥讲的浏览器上种种的问题。
乌云主站负责人 : 第二个是手机移动端。其实我们接下来的议题就是茄子给我们贡献的移动端种种的安全隐患，特别是现在正在用安卓手机的各位你们要仔细听。
茄子: 非常感谢乌云给我们这么一个平台，分享这些，对大家有益的攻防的东西。我今天讲的议题是细数安卓WebView的那些神洞。WebView是安全中常用的组件，这个组件的功能是导入一些网页的浏览，像常用的微博、微信，还有手机来装的所有的APP和核心的浏览器，都是用的这个组件。
茄子: 我自我介绍一下，我和剑心认识有七年了，还有疯狗，我们是比较好的朋友。我是2007年在360工作，也有七年的安全从业经验，之前一直做的是Web安全。
茄子: Web安全做了三年之后，开始做浏览器安全，曾经也获得过微软的致谢。从2013年我开始做安卓安全，就是移动安全这块。

茄子: 首先我有一个引子，就是2012年9月12号Google官方公布的一些漏洞公告。这修复了很多APP的，它只修复了APP漏洞，但是并没修复安卓的问题，这个漏洞可能有一些大家背后不知道的秘密，两年后，正好今天是一个非常巧合的日子，9月12号，也是两年这个时间我们来看Google对它安卓系统做了什么样的修复。
茄子: 第一各问题是，Google在APP中内置了XSS漏洞，它是通用XSS漏洞。安卓系统版本4.4中都有通用的XSS漏洞，我这里列出了五到六个，实际上全部公开的至少有几百个。
茄子: 我了一个BUG的号是37383，这个漏洞是前段时间乌云网站上炒作的漏洞，应该说这个漏洞实际上影响了安卓4.4以上所有的手机，包括4.3，2.3，就是大家常用的安卓的系统。
茄子: 实际上这个漏洞就是我说的两年前9月12号的时候，Google在自己项目中把yp的漏洞修复了，修复了大家可以看到我这个红框的修复，它是把字符串做了相过滤。
茄子: 如果大家是企业商品人员，上线前肯定要做一些测试，实际上这个项目是内置于一个QA的测试用秘，大家可以看到这个字符串是空白的。
茄子: 我这里有一个演示，来看这个能在大家的安卓手机中导致什么样的危害。我点我的QQ空间说说，在57分的时候没有进行任何说说，但是进入UX页面，在这里出了一个脚本运行，然后我再进入我的说说，会我说说全自动发了一条留言。
茄子: 打开这个链接会进行测试页面，我的私信会自动发送。 刚才可以看到我们常用的QQ和微信，实际上是社交化的软件，如果朋友给你发链接你点击后会被劫持。这个漏洞我给大家的安全建议，就是大家如果是4.4以下的手机，赶紧把手机升级到4.4。给QQ或者微博这种使用Web的建议在APP中不相关业务的分离。
茄子: 下一个我要讲的是Google在APP中内设的远程代码漏洞，刚才案例可以看到Google是非常不负责任的企业。第二个案例更可恶，刚才我第一页说的，两年前的9月12号Google修复了自己其中的一个漏洞，这个漏洞价值500美元，这个漏洞是日本的一个安全研究者发布的。
茄子: Google为了修复这个漏洞，因为这个漏洞分节点是安卓的4.1版本，这个版本中有内置的加速化切口，也就是你的APP没开发这个接口，但是你用了WebView这个组件，在安全4.1及其以下的都会接入这个。
茄子: 在4.1这个版本也是4.2开始，Google在安卓系统中增加一个特性，可以防止这个漏洞反射注入的行为，实际上就是屏蔽掉了代码输入的漏洞，但是这个东西是用安卓的非常严重，每个APP编辑开发者编辑的时候会选择你的兼容性，比如我当前编辑的是大于等于17，Ok，你加入的话，你的APP在4.2以及以上是不存在漏洞的，但是这会存在很严重的兼容性的问题，也就是你的安卓在4.2及其以下运行，也会存在远程代码执行漏洞。
茄子: 到底Google对这两年前系统漏洞怎么修复的？它太不负责任了，Google到4.4这个版本才彻底屏蔽了这个问题，就是不管你APP编程是多少，到这个系统上运行你的远程代码漏洞没有，否则的话在4.4以下任意机器上去有这么一个开发的话，都会存在远程执行漏洞。WebView漏洞是做什么呢？
茄子: 实际上它就像PC端流行的那样，它可以做到和PC端上挂码的效果是一样的，但是安卓分很多传感器，分很多功能，我们可以操作安卓应用层代码去执行命令，发短信，打电话，获取GPS，任意手机上手工操作的功能都能实现，在实现这些功能之前，如果下面有安卓的开发者应该知道，要操作这个功能是要获取到APP的CONTAX。
茄子: 获取这个是导入LEI，然后获取到APP的CONTAX，然后调取APP任意层的功能，就是自己编一些代码，但是这里会存在一些坑儿。
茄子: 下面演示一下这个在小米三上是怎样的。这台机器是小米三的最新版，实际上之前报告这个漏洞，现在应该不是最新版了。
茄子: 我在远程监听的端口，我用小米内置浏览器打开一个网页，网页会发送一个意图，实际上是打开了一个漏洞的APP组件，它执行代码成功反弹的一个，当然这个是在APP上呈现，当我下载一个给它权限去执行，我们知道，拿到ROOT权限想做什么就做什么了。
茄子: 实际上，我说了，第二个漏洞希望企业的一些开发者有自己的一些想法，就是不要在滥用一些英特尔加FACE。除了这两个漏洞之外，Google在两年前9月12号还修复了安卓一个漏洞，如果你使用了WebView这个组件，它在默认设置时会使你本地隐私信息泄露的漏洞。
茄子: 这有一个实际案例，像淘宝，其实很简单，我把刚才的过程的代码换了。
茄子: 我的建议是你加上本地协议地址的时候，必须重定空白页。我今天要讲的WebView最后一个漏洞，它是安卓签名漏洞。
茄子: 我给大家演示一下，用这个漏洞去攻击微信的APP，现在我打开微信的APP，假设我现在点击了微信中的一个消息的网页，这个网页会显示我没有安装的插件，现在我给我的手机安装了一个Flash插件，我截取这个信息，我会往微信APP注入了一个软件，大家看戏Flash认出来了，我现在拿到的权限实际上是微信APP的权限，我现在有一个漏洞可以绕过权限去拨打10086的电话，现在我已经远程控制手机拨打10086成功。实际上这个漏

洞要怎么修复？
茄子: 也很简单，开发者在开发APP时候强制去掉插件支持，如果是2.3以上系统可以设Off状态以后，就不会随意支持恶意插件或者Flash插件。
主持人: 感谢分享。今天早上的议题就到这里，下面是午餐时间。
51CTO直播小组: 上午的议题全部结束，下午14:0051CTO将继续为您奉上更为精彩的内容！
主持人: 下午的议题开始了。下面的议题是我们的神秘嘉宾，很多人可能猜到了神秘嘉宾是谁，他今天带来的议题，我先不说，咱们看他讲完了你自己去领会吧。我之前在写主持稿的时候，我要不要把他写成是乌云最帅的人，我没见过他。
主持人: 下午的议题开始了。下面的议题是我们的神秘嘉宾，很多人可能猜到了神秘嘉宾是谁，他今天带来的议题，我先不说，咱们看他讲完了你自己去领会吧。我之前在写主持稿的时候，我要不要把他写成是乌云最帅的人，我没见过他。
主持人: 我当时那个帅字不是对外貌的描述，而是对能力的描述，我见过他后，我觉得他真的是乌云最帅的人。你们先听一下他的议题，有不服气的再跟他挑战，好吗？现在有请神秘嘉宾猪猪侠。
猪猪侠: 这个其实还是有点紧张的，大家下午好。非常高兴有机会在这里和大家分享一个关于自动化攻击的议题。这个议题主要会从过去、现在、未来三个不同的时间跨度来介绍各自背景下的攻击的发展状况。
猪猪侠: 议题一，会引用到我经常用的做实例，附带讲几个小研究，还会披露安全现状，也希望使大家的安全意识提升那么一丁点儿。我先做一下调查，在座的有谁是在甲方的？有在乙方做安全研究的？有多少人是做开发的？这么少人，其他的人是干吗的？
猪猪侠: 我先介绍一下，在乌云ID上的猪猪侠是我的昵称，有八年的从业经历，擅长Web安全，数据挖掘。我的微博就是这个，会后有什么问题的话可以发私信给我。
猪猪侠: 开始议题之前，我想讲一下我的安全观。今天我们大谈信息安全的时候肯定都离不开互联网，互联网到底是什么呢？我理解的互联网就是信息。信息做什么用？信息的一个用处就是交换。所以可以用八个字总结互联网，就是连接彼此，交换信息。
猪猪侠: 而互联网下的信息安全就是保护信息在交换、传递过程中的完整性、可用性和保密性。因为互联网的特性是无时无刻都在发起连接和请求，所以黑客在选择攻击对象的时候不会在意你是谁，而是你有没有与目标系统进行信息交换，或者间接对目标系统发起连接的请求。
猪猪侠: 也就是说，如果耦合度越高的系统，也就是关联程度越高的系统，被攻击的可能性越大。有时候可能某个公司由于单个系统存在某个安全缺陷或者是用户帐号存在一个微弱的问题，都会导致整个网络被入侵。
猪猪侠: 更深入一点去了解互联网的话我们会发现，信息传递的客体对象，社会信息背后的属性都和人有关。说到人，因为在中国是没有良好的信息安全培训，就是安全培训机制的，就是普通网民安全意识还是比较低的。更重要的是有些人的愚蠢根本就没有补丁可打。这个是近年来很典型的APT案例，会发现，其中有一半，70%以上最终问题都出在人的身上。
猪猪侠: 2003年、2004年那个时候我刚如行，当时是IS和ASP的天下，记得那个时候，当时只要手上有每小时旁注的工具有可以做到想黑哪里黑哪里。那个时候比较流行的WEB程序是动议和动网，就是有天动网报出一个信息的时候，你黑动网的话来找到攻击目标，找到论坛就找出管理员的密码。
猪猪侠: 从现在看过去的玖，以前的工具来来去去就那么几个，还都是图形界面的，如果想用脚本调用的话根本没法儿调用。所以那个时候的工具流程都是很简单的一步一步，没有什么新意。过去可以总结成一个《三字经》，无非就是：进谷歌，找记录，没记录，找旁注，没旁注，猜目录，没目录，就嗅探，找后台，穷枚举，传小马，放大马，偷密码，挂页面，提权限，扫内网。个人感觉现在要攻击一个网站是越来越容易，我一直都想实现一个扫描器，只要你点一下鼠标就会黑下一堆服务器，直到我用上了一哥一整套分布式扫描框架之后，再套上一点脚本，这个想法就实现了。
猪猪侠: 就开始可以点一下鼠标黑一堆服务器。先讲一下这个扫描器的特点，先用模块化高可扩展，系统、服务、组件指纹标准化入库，高覆盖率，重新定义网络边界。
猪猪侠: 这里是整个扫描器的控制台，被红色框起来的是信息搜集模块，右边是扫描主机的管理模块，就是你只要把主机添加，设置好它的IP，然后设置好进程点保存，扫描任务就会分发下去。
猪猪侠: 黄色那块是任务管理模块，主要是用来添加扫描任务的，然后还可以查看扫描节点的完成状态。完成框里面主要是漏洞利用模块，可以用来配置一些Web下的识别规则或者管理一些漏洞管理代码或者测试脚本。大致上整个扫描器的结构就是这样，接下来我会逐个介绍里面的一个一个功能。
猪猪侠: 先讲一下流程，就是你添加一个目标，然后会有脚本自动分析这个目标，目标可以是IP段也可以是一个域名，目标分解后会有信息分析脚本去做全面扫描，把系统、网络或者网站的信息全部放到数据库里，接下来漏洞利用脚本选择适合自己条件的目标进行测试，验证后是否存在，如果存在的话会返回结果到数据库

漏洞利用脚本选择适合自己条件的目标进行测试，验证后是否存在，如果存在的话会返回结果到数据库。
猪猪侠: 使用扫描器之前需要给扫描器设置一个目标，这里只要填写一个你想扫描的IP，然后截出IP，设置好扫描参数以后，看这个参数和MIKE很像，因为后台就用的MIKE，设置好这几个信息之后点保存，整个扫描任务就会开始，然后你要做的就是等待扫描器返回漏洞结果。
猪猪侠: 这里就是把域名当做扫描目标，如果你想黑乌云的话，你只要把乌云域名打上，点保存，它自动就会去黑乌云了，后台的大致工作流程，第一步，它会检查一下这个域名有没有存在DNS预漏洞，如果存在的话就会自动利用那些信息。
猪猪侠: 如果不存在，就会利用定义好的去穷检域名，还会有一个脚本去在Google上爬，就采集乌云底下的二级域名，搜集完这些域名之后，接着会有对应的域名把这些写成一个一个IP，当获得这些之后把整个C段一下加到扫描堆里了，会不某个域名或者某个公司的MX加到C段里。当得到IP后，引擎里会查出每个IP对应的域名传到数据库，然后就会开始整个漏洞扫描，之后要做的结果就是等待扫描完成，然后看有哪些漏洞。这里是信息搜集查询界面所要保持IP所对应的类型，所使用的什么版本。
猪猪侠: 这里是更细化了一点，可以看到它的搜集能力还是非常强的，细化到每个小版本。这里可以查询每个主机开放哪些端口，端口开放哪些服务，用的什么信息，什么版本。这是刚刚讲的反查的东西，就是当把某个IP上面所绑定的信息，找到上面首页的东西，一起存到数据库。
猪猪侠: 这个界面是用Web指纹职别规则的，就是说你只要设置好规则它就会检测某个域名，某个IP或者某个系统使用了什么中间件或者什么Web程序。
猪猪侠: 讲一个简单的例子，如果我们要识别一个网站中间件是否用的金Boss，就可以在这个编辑框里设置一个条件，判断HPPT里头是否包含金Boss这几个关键字，如果包含的话就会判断出它使用的是金Boss。
猪猪侠: 如果你要识别某个网站或者域名使用的是不是Opres，第一步可以先对比一下它后台存在的某个文件，然后再看Opres登录里面是否存在某些内容，比如W1里面，你要找回密码就会有密码这样的关键字，然后判断这个网页里是否包含content路径，当所有条件都满足以后，就可以确定这个网站是不是真的是Opres。
猪猪侠: 我们可以看一下漏洞库配置的细节，反正只要你设置好参数，然后写好测试方法，测试规则，验证问题，是否存在的规则，点保存就可以扫描了，就好像这是一个最新出的Cast7.2的侏儒漏洞，就可以分析一下，然后把相应的规则，怎么测试，怎么识别填到里面点保存，就可以批量地扫整个世界上用Discars是否存在漏洞。
猪猪侠: 脚本管理模块，就每个漏洞你可以专门为它写一个脚本，只要你设置好每个漏洞的传参，你只要脚本指定两个参数，一个IP，一个端口。对应的脚本漏洞都可以手工下发，然后我们可以看到图片任务条件那里，是从端口表里查询出所有端口为43的IP，配这个心脏流血这个脚本，点保存之后就会开始全网扫描，全网扫描的目标就是从前面类似钟爱信息搜集里面读取的。
猪猪侠: 这里还是刚刚Discas侏儒漏洞里面，这还是从应用表里查询出Discas的脚本，保存之后就可以进行全网扫描。这里是一些已经执行过的脚本任务记录，任务条件是是否查询，还是简单使用。
猪猪侠: 讲完细节之后再回顾一下扫描流程，如果你要黑乌云的话，只需要在控制台那里输入乌云，点保存，它就会自动分解乌云目标，自动搜集所有与乌云的相关信息，然后会自动存到数据库，漏洞设置脚本会自动生成队列下发，最后你只需要等待漏洞结果。这里就是真的实现了全自动的攻击。
猪猪侠: 整个框架还有可以强化的地方，比如更强的信息搜集能力，也就是要加入更多的信息搜集规则，比如Web指纹，还可以加强资源整合能力，还有更高的覆盖率，更少的误保率，更多的应用规则。
猪猪侠: 在强化信息搜集这方面，未来将，这是1467项Google扫描器加进来。这是规则漏洞库。如果你想扫描整个互联网，就是全世界的话，对网络的资源要求还是挺高的，这是我们的扫描成本，如果你有10台5兆宽的机器扫描A段的话，大概需要7天，就是你要扫描全世界的话需要16万人民币，这个成本非常高的，考虑大成本的问题，我们又加入一个新功能，就是你只要搞到一个能执行系统命令，填上一个密码，点保存，它就会成为一个攻击节点，成本立马降下来了。
猪猪侠: 讲完了扫描器整个结构和工作流程之后，做分享一些我们做的小研究。当我们优化一些细节的时候，发现成功率是很可怕的，这里是优化的两个细节，一个是用户常用口令设置规则，一个是常用特定URL目录字典库的积累。
猪猪侠: 这两个越大的话，效果也会越好，能得出的结果就是稍微大一点的互联网公司就淡定不能不了，等一下会有一个漏洞扫描结果给大家看一下。
猪猪侠: 就如果研究用户密码设置规则的话就得上大数据，这里我们搜集了10亿条用户和用户密码，我们得出了这个东西，我们利用大数据，统计出了100多个使用的最多的弱口令，标红色的是Top10，中国是使用英语较少的国家，我们可以看到前10位全是数字，再看蓝色标记的，这个很有意思，这是一个多情的国家。我还对比一下，我的更具有中国特色，我这是Top数字使用数据，在整个10亿条密码中，使用123456有4千多万，使用

123456789的有9千8百多万。
猪猪侠: 分析这10亿密码可以看到密码用纯数字的大概有3.7亿，就占了36%，密码全部用小写字母当密码的占了41.8%，密码全是大写的占了0.38%，用户名和密码相同的也有两千多多万，占了2.36%，密码中包含了一部分用户名的也有1.32%，使用Top10口令的大概占了17%左右，口令中含有特殊字符的占7.24%。
猪猪侠: 这是整个整个针对使用10亿密码计算出来的结果。当我们分析一些密码的时候，往往也能够得到一个规律，就是我根据那些10亿条密码得出的结论，就是大多数人设置密码的时候都会把生日加进去，还会把年份加进去，比如这个密码是去年设置的，会把2013年加进去。还有可能是名字加上生日，做技术员的会转换一下，还有设密码的时候，复杂度有要求，就要有数字和特殊符号，就会加上@，还有键盘布局，和公司相关的东西补充进去，还有些地方要求大小写，要么就是第一个大写，要么第一个单词大写这样，或者把自己姓名的第一个首字母大写。
猪猪侠: 这是我们发现频率最高的密码设置方式，就可以总结成这样，要么是用户名加生日，用户名加特定字符，要么是用户名@加生日，中文名拼音加生日，中文名拼音@加拼音。
猪猪侠: 根据这些规律，我们实现了生成密码的算法，可以根据你这个人生成一个和你有关的密码。大家可以看一下，这是一个分隔符，比如很多人会用乌云@123，要么乌云.123这样来连接后面一些特殊常用字符，就有一次我在微博上面看到李铁军，他教别人怎么设置密码，他说他的密码一般都是在后面会加上一堆@123Com，他把这个方法告诉更多人，更多人可能会采用这个方法，所以你只需要看他们怎么设置的，然后把它变成算法，自动生成出来，这是处理生日的一个小函数，我们知道日期这个东西会有前导0的，没有前导0的话是9，有前导0的话是0909。
猪猪侠: 就是你把能够考虑的细节考虑进去，就生成一个专门有针对性的密码脚本。有了生日密码算法，我们发现都是中文名，我们又有一个小函数，就是把中文名转成拼音，然后再把你拼音的名字，比如姓不转成拼音，把你的名取前面的两个开头字母，这里可以看到朱
猪猪侠: 整个生成别人密码的过程是动态生成的，有人会说你根本不知道我的名字，不知道我的生日是没办法知道我的密码的，在这里的话，因为互联网上已经泄漏了10亿多条用户信息，我们知道，所以这里可以做一个数据接口，然后给扫描器调用，给算法调用，这里是我做的一个案例，大家可以看一下，乌云上也介绍过，就是你只需要在这个系统上面输入一个域名，然后它就会返回所有与这个域名为结尾的所有人的信息。
猪猪侠: 以京东为例，你比如360buy.com，点击，它就会把所有的信息返回过来。我们只需要知道一个人叫什么，他的人生日是哪天，或者不知道通过什么渠道得到了他的身份证号，就可以很简单地利用一些常用的密码生成算法，算出他的密码，比如前几天泄漏的QQ群数据，备注里就有很多人的真实姓名，那个拿过来的话就可以生成密码，效率大大提高。
猪猪侠: 这里再介绍一下众测刷钱的方法，我得到最终的信息，一个是用户名和密码。指令好，美杜莎支持协议，举个例子，在蘑菇街上，直接用美杜莎暴力截取他们的邮箱服务器，整个过程都是自动化的。
猪猪侠: 我们知道每个互联网多多少少都有应用或服务，比如邮箱、博客或者企业应用，如果当某个员工的邮箱或者密码泄露之后，都会直接导致网络边界被突破，可以看乌云上面的一些精彩案例，第一个是，我是如何进入中石油网络的，还有阿里巴巴信息泄露，直接进入内网，右边是优酷和土豆，由于敏感信息泄露就被内网漫游。
猪猪侠: 阿里巴巴的问题是由于员工安全意识不足，内网帐号挟持就变成了秘密访问，后来你都懂的。中石油是内部文档被员工上传到百度文库，这个文档里面还包含了用户名和密码，直接被人进了内网，就买了那么多防火墙，其实有时候是没用的。这是第三个小研究，就是有时候你用心思考，一些开发和运维的小习惯，把他们的习惯写成检测规则，你就能够在互联网上下载到各种各样的敏感信息，惨不忍睹。
猪猪侠: 比如说上面的灰色是备份文件的，比如你用VI编辑文件的时候，如果忘记保存，或者你在管理服务器的时候HS断了，它会有一个交换文件放在那里，这是我能想到的一些东西填在这里。下面这个就是网页浏览包，就是管理或者运维的要发布一些新的版本，图方便会把原码打成一个包扔上去再解压，有时候忘记删除了原码包就会造成更严重的影响。反正大家上乌云去搜各种案例吧。
猪猪侠: 还是回到正题，未来扫描器会更加智能，会有更多的规则，会有真针对性的扫描，总之未来攻击会越来越野蛮。
猪猪侠: 这是在未来要做的事，就是在传统安全公司还在讲防火墙这些东西应该联动起来阻拦攻击的时候，我们已经实现了实时扫描，就是里只需要在浏览器上设置一个代理，你在浏览器上每次点击链接的流量都会生成一个拷贝，服务商就会分析流量拷贝中的端口信息，会自动传递给扫描器，后台就会自动扫描服务器存在哪些漏洞，这里是扫描器插件，你可以用扩展写成插件，去分析当前网页中存在哪些URL资源，端口的信息，再回传给扫描器，后台会下载一个扫描任务，然后扫描个人存在的漏洞。

猪猪侠: 如果这个东西也可以用在移动客户端，如果你要找移动客户端的漏洞，你在上面播上微片，微片会自动分析出你每次点击的流量的URL资源，后台会把这些当做扫描目标，自动扫描它存在的漏洞。
猪猪侠: 这些用到的代理服务端都是修改过的，主要目的就是用来分析出流量中包含的URL，域名和端口给扫描器用。我们这里流量控制台得到的分析URL，主机，域名，端口都会在这里，然后会分发给扫描引擎，就是这样的话未来你找漏洞，就变得很简单，你只需要在浏览器设一个代理，然后点点就可以找到漏洞。
猪猪侠: 然后在移动APP上放一个微片，然后在移动APP上面各种点，使用各种业务有能自动发现可能存在的漏洞，这个工具近期会在乌云上上架。然后利用这个代理，你可以很轻易地找到产生的漏洞，这是扫描到的腾讯的漏洞，这是百度的，这是微博的，就是说你找漏洞不需要再像以前那样开一个分析工具，然后还要各种复制粘贴，不停地来回地切换，很繁琐，现在的话找漏洞主要点点点就行了。
猪猪侠: 有一天不小心扫到了阿里云的服务器，Agent又忘记删除，然后又不小心发现这里连接了ACE的数据库。所以说未来扫描工具并不可怕，至少有规则是可以检测到扫描行为，但是有些突如其来的攻击是完全没有办法防御的。
猪猪侠: 之前剑心跟我说，筹办本次峰会的目的，是想白帽子和厂商的攻防实践对安全的认知，让谈谈云计算，大数据的看法，顺带披露一些安全现状。现在讲一下安全现状，一个是互联网的攻击根基在动摇。
猪猪侠: 看到这个图，不知道有没有人知道这个事情，就是说2009年1月8号的时候，由于Discuz和Phpwind第三方接口问题，在DNS被劫持。这是新网的，当时安全做的很差，只是证明一下有这么一个事。
猪猪侠: 再看这个图，这个就更久远了，旺旺，06年11月份的时候，因为旺旺某个子目录下面使用了动议的4S版，后来注入了一个成功在它们网页上排窗。也就是说，旺旺，新网，国内最大的厂商都没办法保障自己的安全。
猪猪侠: 所以我们知道世界上可以总结出有三种人：一种是被黑过，一种是不知道自己被黑过，还有一种是不承认自己不黑过，后来我又想到第四种，就是正在被黑。
猪猪侠: 我们的基础运营商其实比想象中要脆弱很多的，就是从乌云上可以看到无数的技术运营商都在乌云上被人报过漏洞，比如说第一张图，这好像是中国铁通的信息系统，就可以入口进去。这是中国电信某台服务器上保留了大量核心路由的配置文件，看配置文件里是什么？
猪猪侠: 配置文件里基本上是某个城市的核心路由的IP，管理帐号密码或者是Root的密码。这个不知道是哪家的，广告系统自带DNS劫持功能。
猪猪侠: 有一天我无意间发现某个运营商系统是可以查询所有手机用户短信发送记录，可以看到左边倒数第二个，它还可以帮助用户发短信。有了这个系统后发现手机二次验证安全怎么办？就是你发一个短信，然后别人就能从运营商系统查询到短信的内容，就是你要找回支付宝密码。
猪猪侠: 输一个验证码，就给你重置了，QQ、微信等的密码更简单了，因为我发现微信用户体验最好的，就是它会自动识别出你手机号码是什么，按一个按纽，找回密码，然后查询出验证码，就能重置别人的微信密码。
猪猪侠: 中间的红框是我在测试的时候发现，我只是修改一下手机的服务密码，系统把我的服务密码也记录在里面了。根基在动摇的一点，就是你的身份证已不再是秘密。
猪猪侠: 就是你的出行记录，你的身份证不再是秘密了，因为整个国家的基础系统其实没有想象中那么安全，然后这是我在群里找的图，就是说别人要知道你的身份证其实很容易的，只需要知道你叫什么名字，你的生日，就可以推算出你的身份证号，有了身分证号，你在互联网要怎么玩儿，其实真是想怎么玩儿就怎么玩儿。
猪猪侠: 其实还有一点，信息泄露比大家在互联网上看到的要严重的多，远远不止十亿条那么简单，我们可以看到这张图上面，整个社会媒体的格局图上面，大家知道有多少家被黑过吗？这就发一个图，我就不讲了。
猪猪侠: 再讲一个小实例，就是王思聪在网站上暴露某电商送货太慢，就把自己的订单号推出来了，你看到他的定单号可以查到他的电话、地址、邮箱是什么，然后可以间接地搜集到大量的关于他的个人信息。
猪猪侠: 比如说有一天，他又晒图片，这是晒他的电脑桌，然后那里有一个小优盘，其实价值一万多，这是土豪的生活理解不了的。我就去看了一下，这个是真的，他真的买了，不是山寨的。
猪猪侠: 所以到现在，我越来越能理解，就是那些明星照片泄露事件，其实说根本的都是因为他们自己的不小心，自己也不知道怎么泄露了。
猪猪侠: 就像今天你去注册微博或者注册QQ空间，人人网，就各种网站，当你填下注册资料的时候，你永远不知道QQ会给你公开哪些信息在互联网上，微博会公开哪些信息在互联网上，有一天你只需要把所有涉及到媒体上面的那些公开信息抓下来，再做一个差异化对比，发现这些不对称可以完全还原出关于那个人的所有信息。比如说微博有时候会显示一个人的生日，人人网上不会，但是它会显示某个人是哪个学校的，你只要抓住这些差异，就可以还原出你想要的某些人的某些信息。
猪猪侠: 现在很多订单会泄露你在哪里吃过饭，睡过觉，这是某团可以查到任意用户信息，你的订单全部存在网上的，比你想象的要严重。

上的，比你想象的要严重。
猪猪侠: 随着移动互联网这几年的爆炸性发展，就来自移动端，就是手机端创造的数据是传统PC时代的上百倍，就在这样的一种大数据时代里面，越来越多的移动用户，移动设备无时无刻不在工作，随时随地都在连接互联网，前所未有的用户数据都被上传到云端，用户变得越来越透明，用户的安全也和互联网公司的安全捆绑在了一起，未来安全问题不在于黑客攻击那个电脑，很可能把某个互联网的服务器攻破，就可以得到上亿人的个人隐私信息。
猪猪侠: 未来我们面对的对手都是信息发掘资源整合的高手，我们不知道他们现在到底掌握了多少资源，但是从我个人了解的情况来看，结果很不乐观的，反正只要是一旦遭受攻击，结果都是灾难性的。
猪猪侠: 对于处于防御场景的甲方来看，有时候你只要被黑一次只要被黑客带走的信息足够多，下次他依然能够接触那些以往获取到的信息再次黑进来。大家有什么问题可以提。
听众: 请问刚才扫描器的时候，它对CPN的处理是怎么处理的？
猪猪侠:
猪猪侠: 如果有预存储的话就看有没有CN代码。就好像你使用安全堡服务的时候，它必须指定你要加入一台，就是必须指定一个，就是ND5加密的域名给它，就像前面讲的DNS已经不安全，你就想办法在DNS厂商那里能不能搞到记录，如果不行的话，就是乌云上的案例，你在上面找一些注册服务，看它的头，找它的信息。他刚刚提的问题，更重要的是你可以去它的M差记录入手，就是邮件解析切入。
听众: 有提到目录测试方式，还有高危的会自扫，我之前做过类似的东西，发现因为有些网站，当互联网站点特别多的时候会有误报的问题，就是我可能让它访问一个404，自己的数提出来，然后做一个对比，但是还有好多很大的误报程度，比方不同的后缀扫的结果不一样，还有一些文件，还有一些目录下，你访问什么都是403，就是类似这样的误报情况，你都有什么处理方法？
猪猪侠: 这样把判断规则写深一点，我用分布式引擎，就是说一哥写的，如果关于扫描引擎上有什么问题可以问一哥，我主要是用里面，就是把它的核心功能拿过来，然后再脚本套脚本，扩展，扩展，再扩展。你刚刚那个问题，就是说如果它出现的404，就是同类的访问太多的话，你尽量把判断规则多写几层，就是出现什么问题写一层，有时候真的太难识别的话放弃就好了，因为还有其他方法可以那个。
听众: 您对数据挖掘这个东西具体是怎么定义的？你是如何去做的？数据挖掘跟信息搜集差不多，就像通过搜索引擎以及各大门户网站搜集个人的信息，然后把信息整合在一起，这就是所谓的数据挖掘？还是说你有自己的看法，自己的其他方法之类的，可以分享一下吗？
猪猪侠: 其实也没什么方法，就是你看哪里提供了数据服务，比如QQ群上，你可以查看到一些信息，你就想办法把它全部扒下来，还有微博个人资料里，很多时候会泄露某个人的个人网站或者邮箱，那里是泄露特别多的，你就想办法把它扒下来，就是哪里泄露信息你把它扒下来就完了，最重要的是有数据可分析。
听众: 刚才讲的都是攻击的问题，但是企业方面更多的关注是平台是否安全，结合攻击思路，在防守方面有没有体系化的建议？
猪猪侠: 你这个问题太大了，我一时半会儿回答不了，但是我们可以线下交流。
主持人: 感谢猪猪侠带来的精彩演讲及现场解答。我就说猪猪侠肯定是有着超级棒棒糖的，他今天让大家见识了一下，猪猪侠经常说的一句话，就是安全不在于对方有多强大，而是只要有薄弱地方就会导致功亏一篑。真的是这样的，在座有很多企业朋友，看到这么多案例，你们可以回去思考一下。下面请极路由副总裁康晓宁就只能路由器在家庭网络中的安全考量发言。
康晓宁: 大家可能奇怪我为什么会出现在这儿，其实我也在想这个问题。乌云邀请我讲这个的时候，我也想我去，这不是羊入狼群吗？乌云说你作为厂商的代表，你还是得出现一下，这样的话，让大家知道你们都在想什么，都在做什么，为什么会报出那么多奇怪漏洞，你们也不修什么的这些问题。
康晓宁: 所以我想了一下，我决定还是把自己洗的白白的，在这儿供给大家来那个吧。
康晓宁: 我今天的题目是写了智能路由器在家庭网络中的安全考量。我觉得你们看到这个以后，你们现在不说什么，但是心里肯定是要对我说三个字，说人话。实际上翻成人话，就是家庭网络还是属于人傻钱一般多的状态。现在不是问题，将来总会成为问题的。
康晓宁: 因为我们知道安全一般来说是，主要防护采取的策略，就是说你采用多少的防护取决你这里有多少的价值。在之前，家庭网络它一般来说攻击的价值非常低的，所以一直都没有形成非常普遍的攻击，大家都觉得这片地里没什么价值，但是随着自动化的攻击，以及将来物联网时代一来临以后，家庭的网络价值肯定会大大地上升，所以说有时候这也是我觉得很那个一点。
康晓宁: 就是一个东西在它真正有价值之前，你肯定不会对它投入很多力量，但是等真正有力量再去搞就晚了。现在返回头来看我们这些安全的东西，大部分都是走的先污染后治理的路，所以今天我在这儿简单讲一讲，可能也是说希望大家早一点，先注意到这点，虽然说先污染没什么，但是别污染太过就好。

能也是说希望大家下一点，先注意到这点，虽然说先行来设什么，但是别行来太过就好。
康晓宁: 既然我们是做家庭路由器的，我肯定得先从这方面讲。家庭路由器是家庭网络的唯一入口，如果这个被攻破损失就非常巨大了。未来物联网时代，大量设备将通过Wifi接入，大家知道这个不太安全，比如说我今天刚来这儿，打开手机一看能上的Wifi没几个是真的。
康晓宁: 另外家庭局域网里头现在可能没什么东西，但是将来还是比较风险的。传统路由器我觉得我都不外说了，央视已经报道好几次了，虽然演传统路由器拿我们的产品拍了很多照片，搞的大家好像以为我们有问题。
康晓宁: 传统路由器厂商一般来说它都没有太多安全概念的，我觉得大家应该在这方面稍微有点研究的，可能都看过DV的Blog，它上面好多路由器的构架，发现这里头或多或少的都会有后门，就是为什么这些厂商费劲的在这些设备塞上后门，这个事是有原因的。
康晓宁: 但是传统路由器功能是有限的，所以被攻破的事情你也做不了太多，很多嵌入式设备连Shell都没有，你就算建成了又能怎样呢？虽然是这样，路由器本身的设置还是能在家庭局域网的安全造成一定影响。
康晓宁: 一个就是说DNS设置，刚才猪猪侠已经提过，DNS在大的DNS上有很多安全问题，在局域网上路由器就是更关键的部分。因为我们知道当时央视报的说有很多路由器DNS设置已经改了，一些关键网站会被钓鱼，但是它没说受到多少影响？
康晓宁: 我们在电信那儿来听说，在全国已经有百万的路由器被修改了。另外还有一些小的，结合着将来自动扫描器的话，是有可能做出在路由器上利用端口转发功能将内网暴露在外网的可能性，虽然我现在没见到这方面的成果，但是这是一个可能性。
康晓宁: 什么是智能路由器呢？我们的想法是想把路由器像手机似的，可以自己装APP，把路由器暴露出来，操作路由器，来修改它的行为，就像安卓一样。但是这个说白了，可能也有更多的安全问题。
康晓宁: 我们只是觉得潘多拉的盒子总得有人打开，我们今儿不搞，明儿也会有别人搞。我们毕竟是互联网出身的，我们比传统路由器安全意识好些个，但是真正实施的时候确实出问题，因为我们也是小本生意，我们也雇不起太好的程序员，我们只能雇实习程序员，他们写的代码你也无法控制。
康晓宁: 另外一点，系统功能还是蛮多的，我们做智能路由器都是用的那个板，攻破这个跟攻破一个主机是没有任何区别的。以前可能大家觉得家庭网没有意义，但是现在这个东西攻破了以后可能是非常好的，这点我想大家可能要这方面注意一点。
康晓宁: 我刚才有一个问题丢了，没有给大家讲，就是为什么传统路由器会有一些奇怪的漏洞，大家还都不修。这个原因其实是因为任何做这种设备的，在出厂的时候都最终要进行一次监测的，就是你看到它都装一块了，肯定不是不行的，我们攒电脑还让整个跑一遍。
康晓宁: 这个设备有一定的测试程序，来保证这个系统能正常运行的。但是你说默认都是有密码，这些东西肯定也不太方便，所以一般的厂商都会有一些，不管什么样的，一定都会有后门，而且这个后门功能还都比较强大的，基本都能跑命令的。
康晓宁: 至于这个后门做的好不好，这个完全是一个良心问题。像DVTV报的都是这样的漏洞，这个漏洞怎么说呢？
康晓宁: 虽然大家看起来很严重，但是从厂商的角度我们觉得这是一个小问题，就是这种漏洞可利用价值一般都不是很大，就是如果我要渗透当然是非常好的一个来源，但是因为家庭网络价值不大，渗透的价值也很低，除非它能够被进行大规模的自动化利用，就好像DNS设置那可以，就是通过跨站的方式来搞。
康晓宁: 就是这种产测漏洞利用价值，因为方法不一样，有些通过脚本是无法构造的。其实我们也是有一个后门的，这个我可以直接说，但是我觉得后门做的稍微有一点点良心，命令都是通过ISA签名的，产测软件是由联网那块生成签名。还有一点，产测的功能我们是在头一次完成设置以后，我们会把它关了，但是传统路由器厂商我们没见一个有做这样事的。
康晓宁: 我们在路由安全考虑的，首先是登录安全机制，因为像传统路由器，尤其是某最大的厂子，我也不说厂命了年出货量才达到七千万台，它上面有很明显的跨站，甚至有的URL可以在用户没有密码完成操作，这样简直太容易了，我只要随便在很多网站深挂一些命令，用户只要一访问立即就改掉。
康晓宁: 我们本身有这样的机制，但是密码这个事情说老实话非常难以避免，我们从我们数据来看，我们虽然在安装的时候加了很多的提示，说你们做密码不安全什么的，最后下来大概还是有三分之一用户绝对不会去改默认密码的。我想传统的只可能会更多，不可能更少。
康晓宁: 还有我们在路由器里我们尽量实现最小权限，把这些程序逐渐都限制普通用户，甚至有些，比如爱情还做第三方应用，第三方应用一定要跑到砂箱里。还有一个，我们本身有一些管理功能，这个必须得增强，但是这部分不是传统的互联网的安全范畴，虽然我们做的也并不好，我们被人脱过裤子。
康晓宁: 稍微扩展一点的话，讲到家庭局域网安全的问题了，路由器自身安全只是一部分，但是我们看到Wifi本身安全问题，还有物联设备安全问题比较显著了，还有你在里头用的电脑什么的，这些设备被攻破，你的家庭

局域网本身安全也就完蛋了。
康晓宁: 因为安全和便利总是违背的，很少有人说我能接受说我家里的网络也就是说像在外面一样步步为营，这样很多人都受不了。所以家庭局域网的安全这个问题我觉得应该是非常难解，这不是一个技术的问题。
康晓宁: Wifi协议本身也是有问题的，就是说WEP，WPS这几个东西都是已经很明确，就是可以被搞掉的，还有弱密码，还有像WAP2，即使它的加密还算比较强一些了，但是我还是可以通过，就是说Wifi协议设计不完善，我可以在一开始的时候，就是正在通信的客户端侵下去，然后获取合理的利用包，再设置权限。
康晓宁: Wifi协议设计，就是从密码达到最终密钥过程太短了，如果我们把这个计算量加大的话，其实就没那么容易被穷举了。现在没办法，因为我看这个协议的时候，我心里想，我觉得搞比特币那些人把常用的算法搁到比特币里真的就完蛋了。
康晓宁: 我们知道比特币已经被这帮人推到很可怕的程度，它还有一定的良知，它没有采用现在常用的加密的认证算法。还有家里的Wifi密码，共享软件，这是一个很大的风险，这个事报过几次，但是我不知道厂商利益还是怎么着，大家都不说这个事。
康晓宁: 还有物联安全设备，我觉得这个都没办法了，它们对安全意识非常非常薄弱，我们和物联厂商有接触，当时想要它的控制协议，它们不肯给。我说看一眼，发来一个Word文本，我用一个多小时的电话跟他们讲明白这样做的危害，如果真的这样的话真没办法。
康晓宁: 还有淘宝上的某品牌的智能插座，我看它是22开的，直接就进去了。我在有些群里头看有些人说，国内物联厂商服务器基本都被掏过，还有人说看到某家智能插座的厂商的服务器，看到里头大概有几千人在线，那人说什么时候有一万了我们就停一次电。
康晓宁: 物联安全意识不强，很多设备预算不足，也很难实现比较复杂的认证协议什么的。这个也确实是摆在它们面前的一个问题。像刚才说安全和便利是违背的，尽量和稀泥，能折衷就折衷吧。
康晓宁: 我们在能做到的范围内尽量做一点事，但是对于家庭局域网大多数用户来说，他们对安全完全没有任何概念，所以跟他们讲这个，就是我们这么做，他们甚至觉得不好用，这个我们就真的没有什么太好办法。
康晓宁: 这个我觉得可能是我今天来这儿的最主要的目的，就是我们还是希望作为厂商或者作为有一点点良心的厂商，我们希望和安全社区保持联系，随时进行互动，我尽量地把我知道的安全方面的，比较牛的人，就是我在微博上都发了，就是你不经意的举手投足都对我的价值非常大。
康晓宁: 现在我们也委托乌云进行体检活动，当然每期都非常惨。
主持人: 谢谢。今天来的可能都是白帽子，在物联网方面研究可能欠缺，但是康晓宁讲了很多东西，你们认真听应该能听到不少东西。而且极路由大家很多都在用的，而且我们乌云集每次会上很多。
主持人: 极路由对安全的态度，我非常欣赏它们对安全的态度，他们对安全并不是说有了什么安全问题就隐瞒着，他们会说出来，然后去改正，去修补那些漏洞，我觉得这才是一个正确的态度，再一次把掌声送给康晓宁。
主持人: 接下来出场的是一位重量级人物，他就是百度的首席安全架构师。他带来的议题是站在厂商的角度怎么想一些漏洞问题。
王宇: 大家下午好。首先感谢大家来。我今天跟大家分享的是百度在安全响应方面的经验以及一些事情的处理，这方面之前也没看到对外进行分享过，但是这个对于大多数企业来说，我个人觉得这个还是蛮有意义的。这是我讲的Agende，我的网名叫小宇，目前我是百度云安全部安全架构师。
王宇: 什么是应急响应？这跟大家刚才看的猪猪侠的演讲感同身受，图片上的人是美国国土安全部部长，他叫切尔托福。他讲过，世界上有两种人，一种被黑过，一种被黑了不知道的。
王宇: 我想说的是一个企业安全做的好，安全事件处理的好的话，很多不是事情发生以后才处理的，包括事前的建设，当时事情真正发生的时候我们才能有条不紊地去处理好它。
王宇: 先不说什么是应急响应？我站在这个角度说一下应急响应的范围。第一个是国内叫的SRC，它对外叫应急中心。应急响应要干的事情？
王宇: 第一，它不是比谁找出的后门多，和黑客捉迷藏。因为本身来说，特别是刚开始处理安全事件的同学，很容易陷入这样的矛盾，我要把所有黑客后门找出来。第二，不是把系统黑一遍，然后说黑客就是这样入侵的。
王宇: 我看到网上有很多分享，关于应急响应的案子来说的话，他说我这个网站被黑了，我拿我的扫描器扫了一遍这个网站，我发现漏洞，通过这个漏洞拿到了分享。这是你的一个思路，但是并不代表事实的真相是这样。
王宇: 第三，应急响应和计算机取证时的关系。其实计算机取证和应急响应有不同的目标，虽然有很多技术重叠，我这里强调慢慢做研究。第四，可能、也许、应该，这有好处，我认为攻击者是这样的，但是真的是这样吗？
王宇: 我说的是大胆猜测，小心取证。我列出了我们内部对应急响应的定义范畴，这里列出六条，首先，控制事

情的影响范围，我们保证线上业务正常运转，是我们最终的目的。第二，还原攻击场景，攻击者到底做了什么事情，包括他是怎么样进来的，以及他进来以后都做了哪些事情，通过哪些漏洞进来的。第三，明确攻击意图。
王宇: 就是说我一个人为什么黑你，当然不排除我在互联网大网扫描黑一个机器，这个站在我们角度我们并不太担心，我们担心的是针对企业的APT，或者有针对性的搞你的。第四，找到问题并改进。
王宇: 就是为了避免错误一而再，再而三犯的话，这里我们需要还原他进来的场景，然后找到自己根本问题的。第五，反思自己不足。我们现有的监控体系有什么不足。第六，在中国情况下用的不多，但是也是范围之内，就是司法追究。
王宇: 接下来聊聊应急响应需要做什么方面的技术储备？应急响应是一个比谁更全面系统的过程。还有一些业务专业知识。不管怎么说，小说里传的是黑客无所不能，他甚至可以打败管理员，这个打败实际上是有一定限制的，我们至少有优势，这里我列出两点：
王宇: 第一，我们是坐前面的人，最次的是把业务下线，这是没有办法的办法，为了保住其他业务。
王宇: 第二，我们对业务的了解，我们可能之前没有接触过业务，但是应急响应会有开发运维的同学来问，首先一个很重要的观点，如果你不了解业务和系统，第一时间去找开发和运维的同学了解。
王宇: 第三，我们自己必须做应急响应很广博的知识，我们需要去了解攻击手法的局限性。
王宇: 刚才也说了不要想当然，A五可能被黑了，他告诉运维的说被黑了，运维的说我们这儿出现一个被黑事件，他告诉上级部门，可能最后跨了几个级别到了处理安全的，这对信息的描述可能完全变味，所以我们处理问题一定要找到第一手的人员，比如这个系统的开发人员，或者这个系统运维人员问清楚。
王宇: 大胆猜测，小心求证，我在后面会具体举例子说。
王宇: 说到黑客入侵会带来系统变化，我在这里说一些tips。大家可以看到这里unix的一个输出，可以看到对于一个默认安装的系统，大多数，用发行版的系统，它的Inode分布是集簇化的。
王宇: 还有安装包情况，我们可以看到系统默认安装完成后会有逆序的管理在这儿。这样装的话很明显的可以看到系统被更改了。
王宇: 还有在系统中有一个很重要的事情，就是确定这个事件发生的时间，这里的MAC是指的修改时间，还有访问时间，这不是文件的创建时间。
王宇: 还有关于删除的，比如说我在Unix下做一个事情，我可能删除sislog，真删除了吗？我直接去访问它的kad，我看这里打开的文件句柄的话，这是简单的事情，复杂的也不提了。
王宇: 默认的Tips还有很多，像代码造成的特征，比如以前的驱动的引用，可能会加在一些不常用的驱动，必然会在Log里打出来。刚说到有些事情功夫在事外了。
王宇: 下面我分享一下案例。应急响应对时间的要求非常重要的，要求处理事件的同学，在第一时间对事件进行一些定性和定量的分析，能做出判断，比方说我要下线，还是重装机器，还是进一步观察入侵者。我举的第一个案子，这都是实际发生的。
王宇: 比如我们这里有运维的同学，自己系统的用户目录会周期性的丢文件，但是又没有丢所有文件，怀疑是有人故意破坏或者入侵事件。第一的反映，肯定先给同学安全意识点个赞，就是有什么问题，安全相关的东西会提交到我这儿，这是安全相关教育的，是我们非常需要的。
王宇: 第二个，这个好嚣张啊。但是反过来想会有这么二的吗？然后沟通发生的时间。我在这里进行了简单的操作，我们看了一下用户的Root，大家看一下，看到这个东西的话，第一反映是什么？
王宇: 我花了五分钟把这个问题破了。我第一时间看了一下Crontab App，这个它删不了我们root输入的东西，紧接着我在Crontab跑的东西搜，前面是Rm，然后后面变成删除根目录的行为。这个案子告诉我们，企业日常报过来的案子，根据我的经验是70-80%是带双引号的。这就是第一个议题，先定性再干活。
王宇: 第二个案例，某一日收到Webshell报警，也确认了是一次入侵事件，这个实际上可以记录在我们系统中执行的命令，我们看了一下，非常熟练。可是这是台linux，我们就不便提了，最终我们是盯下了这个人。
王宇: 我们通过分析这个东西能得出一个结论，当然这个结论可能需要你结合其他东西判断，当然猛地一看，正确评估你的对手，实际上是一个后续工作的指导。当然我们这里相对于小白性质的入侵者。
王宇: 当然我们也遇到特别资深的黑客，结果是我们也是和其他的厂商一起合作，因为攻击的也有其他厂商，我们一块儿把这个同学处理了，目前还没有结案，所以具体细节也不便透露。
王宇: 第三个，一个综合性的例子，第二条说的是大家要正确评估你的对手，第三个是结合我们的平台出现的异常，给大家说一下，分析问题发生的思路。也是某日，编译平台接到了报告，说出现了异常。
王宇: 根据前面的准则，先定性。真的像说的那样严重吗？BAE部分网站访问跳转，第一义反应是被撞库了。BAE负责人的网站也受影响，调了密码帐户的近期登陆无记录，无异常，检查了网站的配置文件，也没有被修改，确实是出现问题了。

段，确实是出现问题了。
王宇: 我去实际的应用机器测了一下反馈，如果知道的就知道了，不知道的话我也不会具体说这个网站是什么。我们要做的就是像攻击者一样思考，攻击者处理我们的网站或者对我们的劫持，他的目的是什么？
王宇: 首先他是跳转到这个网站，比如说Exxxtimes，最主要导致的结果是BAE整体被封禁，这是触底线的事。攻击者是谁呢？宣传XX的，略傻的。是不是炫耀？这也不是炫耀，也没有重定向到某个个人的网站。
王宇: 那么我们确定问题到底发生在哪儿了？我们结合自己了解到的BAE整体框架，这是我们引用了一些内部的，比如BVS大家可以理解为附带均衡，上面跑的CJI，后面挂了一个封闭式系统在这儿。
王宇: 它的报告问题全出现在Python环境，一个机器上有两个lightpd环境，只有一个受到影响。即使受到影响的实例，并不是100%都产生跳转，任意看了几个空间的根目录conf也无异常。
王宇: 我们的目光转向了CGI，BAE多租户特性，同一个cgi会运行多个不同的网站代码，我们做了sandbox的防护，如果出问题，最大可能是Phthon代码对于进程的运行状态进行了改变。
王宇: Python设计之初并不是作为Web的CGI语言开发的，必然导致实现上针对CGI接口暴露过多底层的东西，发现了CGI的CORE文件，简单的Strings了下，发现了大量的exxxtimes，证实了CGI问题的猜想。
王宇: 这样就是定性问题，兵分两路，Phthon环境是经过测试的，能够宽网站影响的接口应该就几个类型，还有动态的执行环境，我会在代码中搜索。另外一个层面上接着去分析，就是说这种攻击实际上是不稳定的，而且它影响其他网站，正常的CGI网站都有运行Max time，在运行完毕后就会被清出内存，攻击者为了规避自身的风险都用国外的，我结合这几个条件对日志进行了分析，发现了几个可疑的运维，当然前面说的我就直接抛掉了。
王宇: 因为那个不影响，先看我当时找到的这样的一个文件，目前看这个地方看不出是什么东西，当然这种正常的写不会这样写，那种写法其实很不单纯的，我和BEA的沟通过，这样的写是某某同学所有的，这个空间多次被封的经历，实际上这与我们最初的猜想蛮像的。我调了一这个同学的测试，名字很明显，Passtest。
王宇: 对可疑网站做过滤的话，因为这个攻击是持续的，很快的我们通过流量系统，把它的一个包被劫出来了，当然有一些解压的过程，我们这里可以看到，实际上是一个攻击代码了。
王宇: 不同的网站用CGI的时候，受到的次数会很多。通过案例三我们得到什么？就是不要盯着技术，特别是对攻击的细节完全没头绪的时候，解决问题但不要局限于问题。当然还有学到一个新的劫持技术。
王宇: 声明一下，最后说一点，目前的BAE架构早就已经更新换代了，我这里只是介绍了当时问题处理的思路。BAE当前的架构不会存在此类的问题。
王宇: 总结一下：未知攻，焉知防；先定性，再干活；正确的评估你对手的水平；大胆猜测，小心求证；跳出纯粹的技术对抗的圈子，以解决问题优先。现在很多事件处理过程中，看似神秘的东西，可能需要一点即破的思路。下面推荐几本书，我是站在共享的角度，不是卖书。
王宇: 《恶意软件分析诀窍与工具箱》，这里奇淫技巧很多。《恶意代码取证》、《应急响应&计算机司法鉴定》、《计算机取证》，写的比较文艺。
主持人: 再次感谢王宇的分享。像百度这样的厂商，它是对安全事件的响应是有一定的机制的，当然在这个机制建立之前他们也踩了很多坑儿，今天来了很多厂商朋友，可能现在的安全制度还在建立中，或者是还在完善中，今天我相信小宇带来的分享可以给你们指很多路的。今天“场里捞”捞到了很多东西，大家可以在屏幕上看一下。
主持人: 接下来给大家带来分享的是一个80后的CEO，虽然是CEO，分享的也是与技术相关的，他就是LBE安全大师的开发者之一，张勇。早晨时茄子给大家分享了一下关于安全攻击的思路，安全就这个样子，有攻有防才有意思，接下来请张总给大家带来一段应用加固。
张勇: 非常感谢利用这样的机会能够在这样的会场跟大家来分享一些在移动应用安全相关的议题。我自我介绍一下，我是张勇，我是LBE安全大师的早期开发者，也是创始人之一，可能我现在挂着CEO名头，但是我还是一个纯粹的工程师，也是一个技术爱好者。
张勇: 关于这个主题其实也是有些小的故事的，在一个月以前，乌云当时找到我们，希望我们能够做一个相关的演讲。我们就很快的想到这样一个话题，是因为我们对这个很感兴趣。半个月前我完成了大纲，大家看后觉得太水了，所以说在一周以前我就推倒重写了，我写的主题是单子上面的主题，前天晚上讨论之后觉得还是太水了，所以觉得干脆再推倒一次，就是再重写一份，熬了两夜，给大家带来这样的主题，就是主流移动应用加固产品攻防分析。
张勇: 这个议题包含两部分：第一，我们跟大家简单介绍一下移动应用加固用户的需求，以及主流移动应用加固的方案和实现的核心细节。第二部分，我会列举市面上五款非常流行的应用加固的产品，来分析它加固的原理和它实现的方法。
张勇: 同时我也会给出这五款产品如何对其进行破解和脱壳的核心的要点。我希望通过这样的PPT一方面让大家

对移动应用加固有些更深的了解，另一方面，我也希望能够帮助一些，或者能够为一些在这个领域内进行研究的安全工作者带来一些资讯和一些信息。
张勇: 为什么需要应用加固？这个问题很简单，在PC上就有，从发展了很多代，在安卓上面从2013年被大家提及，2014年成为非常热的热点。应用加固如此火热的原因有如下几点：首先，安全平台的应用核心代码都是以JAVA书写的，我们知道它很容易被反汇编的。
张勇: 即便大部分开发者都对原码进行混淆，但是核心逻辑是可以通过一些函用关系看出来的。无论是登录也好，验证也好，有经验的工程师还是可以根据调用的关系来找到你的程序里面的核心的点，然后对它进行破解和修改。
张勇: 第二，安卓是一个开放的系统，如果我没记错的话，安卓平台逆向工具应该是1.5时代，也就是三四年前已经发展，到现在非常非常成熟。
张勇: 第三，为什么需要用加固？因为安卓平台淡化了进程的概念，应用是基于消息和事件来运行的，基于这样的情况，恶意代码的植入变得非常容易，想在PC时代植入代码，要对原有的代码进行反汇编，然后插入你的入口点。
张勇: 在安卓根本不用动到原码，只需要在清单文件，增加一些事件的响应，像开机自启动，然后你在你的恶意代码中去接收这些事件响应就可以实现基本的注入。
张勇: 最后，安卓平台，我相信特别是在国内，安卓平台大约有30%的安卓手续已经获得入权限，在这个情况下无论通过APPI或者修改值，都很容易实现注入。实现这样注入话，这样的功能也就在一方面恶化了安卓软件的生态的安全。
张勇: 正是有这样的原因，市场上包括开发者都对于应用加固，都希望能有这样的一些产品来保证自己的软件不会被轻易地重打包，不会被轻易地注入，也不会被轻易地破解，正是有了这样的原因，才有了现在非常火热的应用加固市场。
张勇: 我在国内选了五个比较主流的应用加固供应商。基本上目前国内主流的加固供应上有以下功能：防止恶意篡改，防止内存窃取，防止调试。而且目前的做法通常是用户上传一份已经修改好的APK，然后重打包，反馈给用户的是已经加固过的APK，这套流程背后技术的原理和技术的要点主要有哪些呢？
张勇: 最主要的就是所谓的加壳技术，可PC不太相同的地方是，安卓应用的核心部分代码是JAVA语言书写的，针对很多DX这样的加壳技术应运而生了，在目前三种比较主流的加壳技术：
张勇: 第一，对DX完整的加密。这种加密技术会在加固的时候将DX文件给完整做一次加密，然后保存在APK中，同时用加固方案商动态声明来代替掉原始的。在应用启动的时候，加固的脱壳代码就会自动运行起来，它会对已经加密的进行脱壳并且加载到系统中，同时它还会修改一些东西来运行组件。
张勇: 随着攻防的增加，第二种方案也开始出现，我称为自节码变形方案。它的原理其实也非常显而易见，它在运行时修改文件，使得你从记忆 Dump中盗取的文件是不合法或者不完整，你无法对它进行重新分析和重打包。目前我们看到的像腾讯、360的加固主要采用这样的方案。
张勇: 还有一种就是综合了加密和变形的两种的方案，就是百度应用加固方案，这些加固方案它的具体的实践原理和细节我会在后面的PPT中详细介绍。
张勇: 除了DX加密之外，我们知道APX还有很多东西，像我们写的动态库，还有像资源文件，包括像很多加密方案商提供的防止动态调试，防止侏儒的功能，这些功能把很多加密方案也都有（卡尔）。
张勇: 资源文件的防护通常针对APK中两个比较特殊的目录，一个是RES目录，还有ACS目录，进行保护，像音频、视频，还有图片及其他资源进行加密保存，然后壳代码通常会去确保这些资源在应用读取之前被解密，这个过程是透明的，所以对应用不需要做额外的工作。
张勇: 但是这样的做法可能会有一些问题。首先，跨应用间跨进程可能会失败，影响性能，还有反二次打包，加固时记录APK内容Hash，运行时由壳进行检验。还有反Ptrace，防止注入，多进程相互Ptrace，还有一些厂商采用另外一个方法，就是在运行时不断地轮巡。
张勇: 安卓是多进程系统，你无法保护所有的线程，即便保护了一些线程，其他线程也会被传上去，这样的话都不会很完全。所有安卓的进程都是通过一个叫做Zygote进程报出来的，对于注入者而言或者调试者而言，他根本不需要在你进程运行的时候才Tress，像很多安全软件，像360、腾讯，所谓的超强模式，主动防御其实都是基于这样的技术。
张勇: 换句话说，安全软件可以这么做，恶意软件也好，修改器它们也是可以采用同样的做法。所以说反Tress基本上没有任何功能，从现实中来讲。
张勇: 其实在刚才我列举的这些功能中，最为核心的一点就是Classes Dex加密方案的实现。首先，我想跟大家先介绍一下最标准的Classes Dex整包加密的思路，这个思路也是目前最为成熟和众多产品都应用的方式。

张勇: 这样的方式其实会分两步运行，首先在加固的时候，当把APK传到服务器时会解析APK，它会动态地生成一个新的Classes Dex，来代替原先的，原始的会放在其他目录下面。
张勇: 当程序运行的时候，因为在加固过程中壳已经替换掉了程序入口点，当运行的时候，壳代码首先运行起来，在几个函数中它会调起脱壳代码，通常是用C来写的，来进行脱壳操作。
张勇: 在脱壳操作中会做这样的事情，首先它会把DEX文件加入到内存中，通过一些手段在MAX中看不见，这样可以避免一些非常简单的方法来定位到你脱壳后的Classes Dex下的DEX的内存中的地址。
张勇: 第二，我们知道安卓平台有四大组件，这些组件它和PC是不一样的，安卓平台做应用你是无法真正控制你的代码是什么时候运行的，是系统认为你需要你运行的时候，系统会把你启动起来。
张勇: 换句话说，系统负责来启动你的代码，但是在通常情况下，系统会假设所有代码都存在标准的Classes Dex文件中，但是在加固中你的代码已经挪到了另外一个包中，已经不在原始的里，如果试图构建你这些组件的时候不能实现，所以为了避免这个问题，所有的加固方案会去修改Classes Older，来确保系统构造组件的时候成功找到组件。
张勇: 第二，它还会确保标准正常运作，在执行完这些操作的话，最后一步会做签名验证，然后才会真正调起目标进程的对象，并且对它进行初始化，最后完成脱壳操作话就会重新交给操作系统。对这样的方案有没有可壳的可能性呢？当然是肯定的。
张勇: 因为我们知道安卓肯定不会支持任何的加密，或者任何变字节解码的，它只能接受也只能运行标准的解码。换句话说，任何基于Classes Dex整包加密的方案在程序运行之前必然会在内存中对字解码进行解码，然后才输入虚拟机中运行。
张勇: 另外一个事实，因为Classes Dex非常的大，所以虚拟机会通常倾向采用记忆Dump方法来加载速度。并且Classes Dex也类似ERS，也类似PE，所以基于这样的一些方案，就以下的一些脱壳的方案，最基础的方法就是去查看MAX文件，从中找到Classes Dex的地址。
张勇: 去年的年底的时候，当时的加入方案还非常的初级，我记得有好几家，名胜比较响亮的加密应用，可以直接看到解密后的Dex的地址，你可以把它CAP出来，就可以解密了，所以当时等于形同虚设。
张勇: 随着技术的发展，现在的加固方案商都会采取一些手段来保护，比如从内存中直接加内存解码，或者在加载之后通过MAX调用，把文件映射给转化为文件共供血量的过程，在MAX中看不见文件映射的信息，很难找到MAX。
张勇: 即使做到这样还是有机可乘的，第一，通常MAX文件比较大，需要连续的文件，所以在MAX中是能够找到一些蛛丝马迹的。其次，因为在加载DEX文件的时候会采取MAX方法，这决定了它是是和页对齐的，所以我们可以变异MAX的表象，读取它的每一个表象开头的四字节，判断它是否是MAX字位，如果是那它就是MAX，现在在网上也有类似的代码做这个事情。
张勇: 今年五月份的时候，应该是百度的安全实验室的一个工程师，他做了一个项目，他可以实现包括对于不少通用加固方案的脱壳工作，这也是一种方案，以及我今天跟大家介绍的，如何通过反射的方法来获取DEX的方案。
张勇: 下面我介绍一下五款常用加固产品的原理和脱壳的方法。
张勇: 首先是这个行业的创始人在国内，就是梆梆加固，它应该是这个行业最早的倡导者，我相信它也有最多的用户量。针对梆梆的加固我准备了两个案例，第一个，就是梆梆的企业级用户，就是国美在线，第二个，就是我自己的一个上传的产品进行了加密。
张勇: 梆梆实际上对企业用户和免费用户的提供的防护级别不同的，我稍候会向大家解释。目前，梆梆的代码是支持X86平台。对梆梆的分析过程中我发现一些问题，首先就是stub classes并非必须。
张勇: 第二，梆梆为了支持Art，在内部装了一个东西，这个文件是来自于安卓4.4的。对于梆梆这样的产品，对它如何能够实现脱壳包括解密操作呢？我们知道企业版本和免费用户版本是有区别的。
张勇: 对于企业版本梆梆使用了内存加DEX方法。被梆梆加固的应用没有做所谓的DEX OPX操作的，这样的话会导致一些程序加固后的行为有差异，或者性能变差。
张勇: 这时企业版本，做了更多的防护，对于公开版本而言的话，梆梆使用了标准的开放方式，从文件中下载的方式，同时也没有将DEX下载地址往后，就是绕可四字节的方式，使用公开版本加固过后的产品，是可以通过MAX的方式找到加固后的代码。
张勇: 另外，对梆梆分析过程中我们发现对应用的修改比较多，比如它会同时启动三个进程，相互之间tress，其实这是一个伪命题，所以我个人建议这个功能可以砍掉。
张勇: 同时梆梆会在运行过程中会发送一些特殊的kust，这样会降低代码的可靠性。同时脱壳后的话，梆梆在加壳时在原始的文件中也插入了相应的代码，这个在脱壳中都要移除。
张勇: 第二，基于Classes Dex加固方案是来自于爱加密，它的做法和梆梆大同小异，原始的Classes Dex会被加

张勇: 第二，基于Classes Dex加固方案是来自于爱加密。它的做法和梆梆大同小异，原始的Classes Dex会被加密后放在assets下。它的方案相对而言可能更加成熟一些，但是跟梆梆比也有劣势，它是不支持安卓L的，只支持ARM平台。
张勇: 爱加密相对于梆梆在自我调控方面做了比较多的工作，在公开的版本，它在下载Dex时用了四个方法，在MAX中看不到任何痕迹的，一些关键的指针被它替换成错误的指针，这样的话使用标准的脱壳工具是很难对其直接进行脱壳的。如何来解决脱壳问题呢？
张勇: 其实有一个办法，首先目标用户中选择随意的MAX方法，将其传递给GNI，获得内部的一个MAS对象，MAS有一个指针的地址，这个文件就在它所在的区域中，然后在PROMAX中寻找这个地址的页表象，然后真正的DEX就藏在这个地址中，然后用这个地址寻找起始，就能找到DEX，我们测试结果也是跟梆梆一样，使用了将DEX地址从4K对齐位向后偏移8个字节，这样就很好地避开那个。
张勇: 第三个产品就是360。现在360应用中心推出了一款加固宝产品，它也是和梆梆、爱加密一样，是基于Classes Dex加固的这样一款产品，实验原理都是非常接近的。
张勇: 我们在对比的时候，发现我们将360加固解包后和原始的APK有些须的差异，我们认为这是360对APK做了一些所谓的性能优化，包括像构造函数优化等等，这些都是在标准的ODS做的优化。360目前是支持X86，也能在安卓上运行。
张勇: 上面就是最传统的Classes Dex加密方法的案例和分析，下面我想跟大家介绍的是基于字节码变形的实现，会更加有趣一些，这些方案的出现，其实它们为了解决一个问题，就是我刚才跟大家介绍的，对于任何Classes Dex加密的产品，一旦找到了地址之后，所有的加密都无所遁形，你可以轻易地下载出来，对它为所欲为。这就有了字节码变形加密方案，它本身是一个执行机密的脚本。
张勇: 换句话说，虚拟机只要完成了对这个文件的加载，文件中很多部分的格式也好，数据也好，所以它只需要将这些格式破坏掉，记忆中下载不出来的Classes文件是非法文件，失效文件。为了实现这样的目的目前有两种方案：第一，在加固的时候，首先去蕴藏一些DEX中一些关键的代码，然后在加载的时候再由程序修复。
张勇: 因为修复的过程不是发生在文件本身，即便我们轻易找到了DEX的加载地址，下载后还是不可执行的文件。我们有两个方法，一个，是加密的时候把JAVA的方法改成Lunix方法，将字节码进行隐藏，在运行的时候再将字节码恢复，这个是腾讯采用的方法。
张勇: 第二，在加固的时候把DEX的BAD COAD改了，放在其他地方，这个方法是360的方法。第二种方案，在DEX加载后破坏内存中DEX镜像关键结构体，使得你下载出来的结构无法被静态分析工具或者其他工具来分析，包括像DEX，这个方法是目前百度采用的方法。
张勇: 如何对其进行脱壳呢？首先单纯的记忆DUMP是无效的，单纯的对其脱壳的话，是对破坏的一方进行修复，对DEX TEDER在运行的时候将其破坏值在稍候计算出来的。对于像某些支持ARK的方案，还有比较独特的方案，就是劫持系统的DEX的 OAT，这样把原始的偷出来。
张勇: 第二种，像腾讯和360对MAX进行处理，来隐藏其字节码方案，也有对应的方法，就是修复MAX。
张勇: 下面我想跟大家分享一下我们对市面几款使用变形技术的加固产品的技术细节的分析。首先，就是腾讯，腾讯的加固产品应该说是，我个人认为是最别出心裁的，对于其他家的加固产品，通常是对应一个PK上去，加固一个回传过来，但是在腾讯中需要填写需要加固的函数名，为什么会有这样的情况，这跟它的做法是直接相关的。
张勇: 因为腾讯的应用加固方案在加固的时候会将真正的函数的类型从JAVA函数改为linux函数，将关键的结构体给清空，让静态分析工具认为这个函数是没有任何字节码的，就可以跳过，对于未加固的函数没有任何变化。
张勇: 我们注意到腾讯的加固其实并没有真正把字节码藏起来，给放到其他地方，字节码其实还在这个DEX文件中，只不过它修改了函数到字节码的指针，让静态分析工具和虚拟机认为这个函数是不需要字节码的，解密的时候也是同样的做法，找到函数和字节码的对应，恢复就好了。
张勇: 这个是打开了一个腾讯应用加固方案后的，一个被加密的伪函数，这个函数里设置了AK了，它的代码属性被设成NO，就是没有代码。我们打开DEX文件本身，我们发现字节码还保存着，只是你们不知道如何将它和函数利用起来。
张勇: 这个方案其实非常的有开创性，我个人觉得非常赞，但是最大的问题在于，由于ART原理，这个不可能支持ART的，因为在ART运行中，因为ART运行环境首先将DEX文件编译成本地代码，然后在这个过程中它生成的并非是DEX，而是ELF格式的一个本级代码，腾讯这种做法不可能在ART环境下，在运行时重新将这个函数指向另外一个地方，所以目前腾讯的方案仍然是完全不支持ART的，我觉得未来这个方案可能也无法支持ART。
张勇: 解密其实非常简单，使用DVL的函数盖到MASCOAD的地址，计算出这个偏移量，然后就可以修复这个函数的类型。所以我个人认为这个方案是非常巧妙的。当然就像我PPT中写的，这个字节码是无法保存的，安全感会比较差一些。

张勇: 最后一个方案，也就是最近刚刚百度发布的一个混合的方案。百度的方案是混合了Classes Dex加固以及Classes Dex字节码变形的两种方式的一个方案，应该说在目前是最为完善的。
张勇: 百度的方案对于DEX加固这块，它和360，和梆梆和爱加密是完全相同的，也并无太多新意。但是另外一边，它是如何来隐藏DEX的内容呢？
张勇: 当使用百度加固方案后，它会将很重要的DEX的头的重要的位置给清空，包括DEX TIDER，当你下载不出来的时候，这个文件实际上是无法识别，也无法做静态分析的。地可以看到DEX头全是零，包括后面的OPT信息全是零了。怎么样来实现脱壳？
张勇: 脱壳方法相对而言也比较简单，既然百度将信息清空了，我们将其还原就是了。基于DEX的一些特性，它是连续的，并且不允许有空隙的，我们是可以根据Aseent的格式是固定的，而且百度没有将每个字段的长度给清零，所以我们只要算出来第一个Aseent的位置，然后Aseent加，把每个数字填起来就可以了。
张勇: 后面后这个DEX可以成功地被其他的静态工具分析了。相对而言百度的加固我个人认为比较完善的，因为它结合了现在的两种方式，同时也有比较高的Dop难度，还支持X86和安卓平台。
张勇: 但是百度加固会在原始DEX文件中插入很多百度代码，我个人比较不喜欢这样的行为，这是从我们自己工具里摘取出来的代码，它是来重构DEX TIDER。
张勇: 最后我想提一下360早期版本，实际上360最新版本中已经使用了类似梆梆和爱加密的方法，在它早期的方法也是有创意的，这种方法破解难度很大，但是它在目前的版本上为什么没有延续，我觉得还是跟兼容性相关。因为目前来看的话，所有DEX字节码变形的方案和ART都存在着或多或少接入性问题。
张勇: 最后，我最近做的一些研究的总结。首先，我们认为应用加固的产品本身技术也好，包括攻防也好，是高速发展，去年搜集这个关键词应该看不到太多的信息，今年搜加固可以搜到大量的信息，我们也看到随着早期的DEX文件加密的方式的成熟，现在其实也有更多更新的方式在重现，加壳的强度也会越来越高，据传言国内已经有一些厂商实现了类似PC上面的技术，能够进一步提升加固强度，但是目前没有看到在公开市场上有这样的案例。
张勇: 第二，ART是安卓的未来，从安卓L开始。ART本质是在运行前将字节码编译为本地指令，任何试图将这个解码隐藏，都会导致使本地指令无法执行，这也是目前360和腾讯无法解决的。未来可能有其他的方式，像安卓MASS上面，有的人做的方案，能够未来对EOT进行PACH，我们认为ART在加固方面还有很多空间，目前还不是很成熟。
张勇: 最后，应用加固它其实还是有很多兼容性问题的，在实际测试过程中我们也注意到，像不同的硬件平台，像不同版本的系统，甚至像不同厂商的手机，为什么会有这样的问题呢？
张勇: 因为作加固中很多操作是需要通过反射来做的，还有很多操作是许多对结构体进行修改，我们知道安卓是开放系统，任何厂家都可以修改这些结构体，这就导致了在某些手机上面这个结构体的偏移量，或者位置、地址是不同的，这会影响它同其他设备兼容性的问题。
张勇: 在最近一段时间我的很多朋友问我，对坚固问题怎么看？从目前来看，我认为加固并不是万能药，而且有很多脱壳工具。对开发者而言，一个完整的体系才是他所需要的，他需要多管齐下，才能真正解决APK安全问题，单纯加固肯定是不足够的。
主持人: 感谢张总最后还给大家提出一个建议，张总讲的很细致的，做这方面研究的朋友应该可以听得出来。我想问张总一个问题，作为一个CEO，为什么你还坚持这么久做技术呢？
张勇: 我之前已经讲了，其实我自己从本质上讲我还是一个工程师，而且到现在我还坚持写代码，我觉得对创业公司来讲的话，我认为保持一个写代码的习惯，能够让我跟进时代潮流，技术潮流，能更好地把控产品。
主持人: 感谢张总，所以奋斗在一线的工程师们不要灰心。接下来进入下一个议题，下一个议题是关于比特币的，最近它的事故不少，今年乌云知识库上也有一篇关于比特币分析的文章，今天请到了快钱包的创始人，这个内行人带大家了解一下关于比特币的安全。
张健: 大家好。我的标题是比特币网络交易及存储安全。议程上说是金融安全，我没改过，我不知道标题怎么变了，不知道是不是比特币比较敏感。
张健: 谈到比特币，我为什么站在这里？首先大家认为比特币是货币，或者说是互联网金融的一个范畴，为什么和安全会有如此大的关系呢？其实我想说，比特币的诞生可能50%是属于安全的范畴，有50%属于互联网金融的范畴。我第一章从比特币本质讲起。
张健: 我现在想做一个小的调查，就是台下的各位有没有没听说过比特币的人？应该是没有。第二个调查，谈一下有多少人现在不去查，知道现在比特币大概在什么价格？第三个问题，有几个人知道驱快店这个词？
张健: 这样的话我就第一章讲长一点，但是如果大家不知道比特币是什么的话，这个安全就无从谈起。第一点，数字货币化和数字货币。其实数字的货币化不是一个将要发生的事，它是一个已经进展很深入的事。就是我们

现在所用的大部分的交易，特别是大的交易，其实很少用到现金，我们用的都是银行转帐或者信用卡，合适这些根本不涉及到真正的实物的，就是钞票的交流，就是数字。
张健: 比如我转帐给你，只不过是在我的银行户头上减了这个数字，在你的银行户头上加了这个数字而已，这个现在是非常大的趋势。但是为什么还要提数字货币这个概念？就目前的数字货币化其实非常不彻底，它是支付上的，而不是本质上的数字化。其实这个问题在很久以前，在上世纪七十年代、八十年代就有人不断地提出，是否货币可以数字化？为什么呢？因为货币从本质上讲它就是起到了信息的中介，可以这么认为吧，所谓的一般等价物，它本质上其实是中介，它本身不需要有任何价值。
张健: 比如说你现在拿的一百块钱钞票，它本身就是印刷价值，这个成本可能非常少，是谁赋予它的价值，这是很深刻的问题。人类社会经历了太多经济危机，经历了太多波动，但是它的根源跟货币都有绕不开的关系，因为没有人知道现实社会中究竟需要多少钱，我们先不去揣测发钞者他可能会有一些恶意的成分，比如某些国家的政府，它有可能会超发的情况，即使它是想让整个货币体系和金融运行的更加稳定，它也不能够知道究竟需要多少货币。
张健: 这个其实在经济界就有很大的争论。不知道《货币战争》这本书大家看到过没有，里面提到了金本位，就是它要回归到金本位，因为现在的钞票可以随便印的，这个思想其实是完全不可行的，因为黄金有一个最大的问题，就是它根本不能够用于现代的支付。我不可能说我买东西切块黄金给你，这是已经早已被淘汰的东西。
张健: 我顺着刚才的话题讲，其实货币数字化的根基是帐单，我刚才举的例子，我给你转帐一万块钱，其实只是银行系统的数字改变了一下，在我的帐单上减去了这一万块钱，在你的帐单上加了一万块钱，所以说比特币其实本质上就是一个帐单。但是想实现这样的帐单有很多困难，下一章我会讲到，就是如何实现全球唯一无中心节点的数字化帐单。
张健: 当然中心节点这个问题可以稍微提一下，就是如果有任何一个服务器或者组织，或者个人，或者政府，它拥有这样中心化的权利，那它就违背了我们发行的公平，等一会儿我会讲到。
张健: 第二个，全球唯一，这点也非常难。大家可以想像，一个数字或者一段数字，如何能够成为货币？因为数字是可以任意复制的，所以说这个是一个下面要解决的，就是比特币要解决的核心问题，或者说数字货币要成功，要具备的基本条件。第一，发行问题。如何发行和如何保证公平？
张健: 第二，造假问题。如何解决重复花费的问题，和如何保证帐单不可篡改。这个问题比特币是通过驱快店完美地解决了上述两个问题。解决的方法就是驱快店是什么？我花几分钟讲一下，驱快是什么概念？如果把驱快理解为全网的帐单，驱快就是这帐单的一页，这一页不停的往上加，它为什么会成为链条。
张健: 这要谈到密码学，下面我会讲。但是驱快的，谁来增加一页帐单？谁能够为这个帐单记上一笔记录？这个是要通过算例来竞争的，这是比特币最初的设定。也就是说，如果是，比如说你是一个村子，你要发行一个，我们随便举一个例子，一个组织要发行内部的货币系统，如果这个组织的负责人可以记这个帐，他就有动机篡改这个帐单，所以无法保证发行公平的问题。
张健: 所以我说为什么区快店完美解决了两大问题，第一你要取得记帐权就要通过算例来竞争。第二，如果你竞争到这个区快的记帐权，你就可以获得一定比例的奖励，这就是货币的发行。所以说它把记帐和货币的发行完美地通过区快店的方式集成在一起，这是比特币能成功的基本保障。
张健: 谈到密码学，第一就是公钥密码学，前几天的有人黑了宗本聪邮箱，但是现在身份没有暴露，可以肯定他在密码学方面造诣非常深的人，当然不仅仅是密码学，还有金融学，他完美通过数字的方式模拟了黄金的属性，又避免了黄金的最大的缺点，我刚才讲到的就是支付根本不方便。公钥密码学是这样，比特币采用的是公钥密码学中的算法，就是你通过私钥的方法，公钥可以验证，但是公钥不能签这个名。
张健: 这是一个数字签名算法的一个特性，所以它能保证P2B中交易的真实性，就是我能完全确定你这个交易是拥有CELL的人签名发出来的，如果没有签名，马上会被比特币网络拒绝。第二个特性，就是通过公钥推算出私钥，这个在理论上是可行的，但是在计算上是不可行的，比如要花人类一万年甚至十万年时间这个其实可以认为计算上是不可行的，这是公钥密码学，它是比特币的根基之一。
张健: 还有哈希算法，就是这个链条是如何形成的？其实每一个区快它的发行值是有记忆的，它记忆了从比特币区快店诞生，就是第一笔交易到现在所有的信息，都在那一个值里体现出来，这个就是通过不断地哈希，所以它不会漏掉任何一个信息，你试图去更改一个帐单，不管是一个月前，甚至几前年，任何一个字节都会导致整个区快店的失效。
张健: 第二，生成新区块的工作证明，这个值是比特币非常创新的一个概念，刚才我提到的如何能够，就是谁有这个权利记这个帐单，特别最货币来说这是一个至高无上的权利，但是如何不让这个权利被滥用。宗本聪发明了一个方法，就是工作量证明。
张健: 就是这个方法最初源于垃圾邮件的防范。当时垃圾邮件是长期存在的问题，后来有一个人提出了解决方案，让发送垃圾邮件的人要计算一个数字，比如说计算一个哈希值，其实对普通的PC机计算这个值非常快，你

案，让发送垃圾邮件的人要计算一个数字，比如说计算一个哈希值，其实对普通的PC机计算这个值非常快，你感受不到，但是对滥发邮件的人这个运算量是很大的。
张健: 所以这个想法是源于垃圾防范，它非常完整的保证了比特币的公平和安全。公平可以很好理解，因为谁提供的算例最多，谁就能优先算出这个值。这个值哈希函数一个特点，就是正着算很简单，倒着算不可能。所以说打包区块，或者说竞争新区块的算法是什么呢？
张健: 就是暴力破解一个值，这个值只是我规定的一个特定的值，比如小于某个值，你只要试出来了，你就拥有这次记帐权，你就可以赢得一个区块，打包一个比特币。现在比特币算例非常非常强大，特别是经历了去年价格暴涨之后，现在它的算例应该在200的P，P的概念我不知道大家熟不熟悉？
张健: 就是每秒钟大概要运算200的P的SH256哈希运算。这个算例其实是一个天文数字，大家有兴趣的话可以算一下这个数字有多大。
张健: 下面回顾一下比特币发生的安全事件。其实比特币虽然诞生于安全，诞生于密码学，但是同样在那方面非常非常脆弱。这里面丢失的比特币数量应该都是上万的，还有上十万的。在我展示的七八笔事件中，丢失的比特币数量就高达一百多万，我没有仔细算，大概有120万左右，现在比特币发行的数量是1300万，就是我记录的这几个事件占了全网比特币事件的10%。
张健: 但是真实的丢失，虽然这上面有一些不一定真实，但是真实的丢失应该远远不止如此，我估计在20%以上，包括丢掉，包括被盗，所以这反映了一个非常严重的问题，就是比特币很难伪造，却不难窃取，可能这个话题大家比较感兴趣。
张健: 就是比特币非常难于伪造，不仅仅是我刚才讲到的两个，一个是公钥的数字签名算法，还有哈希值在保证从公钥推算私钥不可行，但是一旦泄露，你的比特币就会被一扫而光。也就是说你存在银行的钱可能没这么容易丢失，但是你放在机器上的比特币，有可能在你不经意间就丢掉了。
张健: 不知道大家看到这个图有没有什么特别的想法？虽然这是一个窃取的图，但是我想说的是这是一个新闻，这个新闻说的是，美国的一个大学安全方面的教师发现有一些黑客黑掉了很多类似于这样的摄象头，用于比特币的挖矿，这个事挺多逗的，摄象头能挖多少矿？
张健: 现在进入案例的分享。第一，自己弄丢，这是一个最简单的丢的方式，但是这种方式占的比例相当大，我没有一个具体的统计，比如说这个交易所在2011年的时候，它由于升级自己的，就是它把服务器放在亚马逊，由于提示资源不够，它升级自己的系统，结果不知道，因为什么方式，设置被改了，造成文件全部丢失，而且它还没有备份，结果损失了17000比特币，导致这个交易所直接关门，造成这些比特币直接出售。
张健: 可能有一些人已经知道这个事情，就是最后这一条，一个公司破产的时候，它宣称丢掉了75万个用户的比特币和10万个自己的比特币，加起来超过4亿美金，这是一个非常惊人的数字。
张健: 第二，黑客攻击。这个案例挺有意思，这个交易所在一年四个月的时间内被人黑了三次，每一次黑的数量都挺大，第一次是4.3万个，第二次是1万8千个，第三次是4万个，加起来差不多有10万个，而且每次黑的方式不同，第一次黑的托管商，第二次闯入帐户通过电子邮件，第三次稍候我会讲一下。
张健: 刚才我问大家比特币价格的时候，不知道有几个人查了，它价格现在人民币不到三千元。这个交易所很有意思，它第三次被盗的时候，在它被盗的前一天，这个交易所创始人在QQ群里大量出售他的LR和B CORN，而且说要多少有多少。
张健: 所以说很多人怀疑这是监守自盗，因为被黑了很多次，不在乎自黑一次，然后彻底关门。因为我一个朋友在这个事件中丢失了500个比特币，所以他要求我一定提下这个人，这个人叫周彤。
张健: 案例三，攻击服务托管商。刚才那个交易所第一次被黑是黑了它的托管商，这个托管商出的事太多，其中一个跟比特币有关，就是超级管理员密码泄露，因为不费工夫就可以更改用户的密码，因为他是一个相当于后台客户服务的帐号被泄露，这个黑客拿到了超级管理员权限之后就扫描所有程序，最后只选择了攻陷八台
张健: 就是因为这八台上运行了比特币程序，也就是说这个黑客就是冲着比特币来的，这八台中出现了这三个人，就是刚才的那家交易所，还有一个矿石，它应该是排第三、第四的，还有一个比特币当时的首席科学家，这个人居然也丢失了50个比特币。所以这个事件给我们的启示，千万不要把私钥放在云平台，即使你没有问题，他可能黑掉你的上级。
张健: 域名劫持，这也是一个挺典型的案件，黑客利用公开的信息，获得了这个拥有者的出生地点和母亲婚前姓名，说服注册商交出了域名和控制权，进而又控制了他的电子邮件，最后进入他在一个交易平台的帐户，然后把这个交易平台全部侵了。
张健: 虽然得逞的原因因为公开的这些情况，可能跟他并没有直接关系，但是他没有开启双边验证，这是一个安全上的漏洞。但是这个公司也挺有故事的，其实这里面每个公司都挺有故事的，这些公司的创始人也是在今年初，因为洗钱被FBI逮捕了，现在应该还在FBI那儿。
张健: 这个事件是今年六月份一个挺有意思的事情，因为今年六月份美国发行局组织了对丝绸之路的比特币拍

卖，丝绸之路是当时一个非常大的比特币，卖东西的，应该是电商网站，但是它卖的东西都是毒、品之类的，所以最后通过FBI的钓鱼把这个人抓了，扣押了它所有的比特币有一个人有14万个，当时是余额最大的比特币地址，就是FBI扣的丝绸之路地址。

张健: 美国发行局不慎泄露了拍卖的名单，这个攻击者就伪造的，也不是伪造，就是发一个采访的文件，只是带了一个木马而已，发给了一个竞拍的公司，这家公司是澳大利亚的一家比特币的公司，这个剩下的事情就比较简单了。

张健: 这个案例体现在安全上的问题，就是即使他侵入了你的系统并不一定拿走你的比特币，所以刚才猪猪侠提到的一个事情，就是很多安全方面的大漏洞不在于你在后的方法做了多少，而在于里最薄的方面是不是犯了非常愚蠢的错误，所以说其实你只要不犯错误，可能就不会出现这样的问题。

张健: 下个案例是最新，刚刚发生的一个案例，就在上个月，国内的一家平台，这个不是比特币，但是也是比特币的衍生货币，我们统称为山寨币，这个币也值点钱，被盗了5100万个。

张健: 被盗的过程几乎是全公开的，可能黑客不愿意暴露自己的地址，直接在他们的地址上谈判，这是我贴出来的黑客在论坛里说的话，他说，CEO90%的服务用了同一个密码，所以这是一个挺有意思的事，如果大家没有了解的话可以去看一看，因为这个事件完整地曝光了。

张健: 也就是说开始谈判过程中，比特儿可以强硬的取款，但是后来因为种种的担心，其实他内心比较心虚的，后来逐渐被黑客掌握了谈判的主动权。所以这个NXT事件还有另外一个原因，就是这个山寨币不能做冷钱包，冷钱包是什么意思？就是把私钥可以放在自己家里。这样的话黑客就无法侵入。

张健: 下面我总结一下比特币在交易和存储方面面临的安全风险。第一，存储的风险。这分两大块：第一，私钥文件损坏，或者比特币丢失，就是私钥直接丢失了。

张健: 合适这个刚才我也提到了，这个造成的丢失不在少数，包括最大的一个丢失事件，门头沟那个事件，85万个比特币，有部分人怀疑私钥是管理不善，就是没有备份的后果是灾难性的。应该有三四个人找过我，其中一个人相当典型，他属于本地的私钥损坏了，把钱包发给我，问能不能恢复，说恢复的钱分我一半。

张健: 我经过很多次尝试后真的不能恢复。所以这样的情况下，不管你有多少资产，很快就丢掉了，就是说没人能花这笔钱，就是它永远的丢掉了。其实丢掉这些钱让比特币整体的数量稍微减少了一些。

张健: 第二种情况，就是私钥被盗，这个也是非常普遍的。可能每个月，甚至每周都在发生，这个对安全性要求非常高，包括企业，我们公司也会采用冷钱包的存储架构去做这样的安全防范。

张健: 也就是说，因为你要用比特币进行支付，所以你必须在业务的层面保证有一定的私钥或者有一定的逼债，所以你不能百分之百冷存储，所以现在很多交易所将大部分币冷存储，一旦黑客侵入，而且侵入的级别比较深，拿到这些私钥也不会损失在线下冷钱包的钱，这是一个现在普遍的一个架构。

张健: 其实交易安全风险，比特币还是做的非常好，这也是比特币安全性一方面，叫双发攻击，因为比特币是P2P的架构，什么意思呢？就是每花一笔钱，这笔交易是通过这个结点性质广播给全网，就是说全网不同节点接受这个交易的节点时间是不同的，所以有可能你可能同时广播两个交易，一个交易是发给交易的商家，另一个交易是发给自己，通过不同的节点广播，一旦出现了，假如说你又同时掌握了比较大的算例

张健: 你就可以通过自己的算例优先确认你广播发给自己的这笔交易，而推翻发给商家的那笔交易，所以商家会误认为收到这笔钱，但是后来事实上这笔钱消失了，这就是双发攻击。避免这个攻击的方法非常非常简单

张健: 因为这个攻击的成本非常之高，所以说只要在发送大额交易的时候，比如说几千个或者上万个，等待足够多的确认数量就可以避免这个攻击，因为攻击程度概率随确认数量的增加呈指数级下降的。

张健: 谈到比特币私钥容易被盗，它的脆弱性问题，其实所有人都认识到这个问题，包括比特币核心开发团队，所以在2012年采访中，比特币首席科学家说正在添加一个新的特性，这个特性可以降低比特币钱包受到攻击后的脆弱性，这个特性叫多重签名交易。

张健: 什么意思呢？简单讲一下，就是传统的私钥被盗，相当于这个地址商的钱全部被人拿走，但是通过多重签名交易生成，就是多个私钥共同生成这个地址，然后有M个私钥签名这个地址上的交易，然后M和N都可以自定义，只要M小于N就可以，这就造成了安全方面的特性

张健: 不仅仅对存储，还有交易，就是对不同的角色赋予不同的私钥，保证比特币本身的安全性。比特币第一次从技术层面上实现了真正的可执行的担保交易，比如说支付宝的担保交易，它是用支付宝信誉担保的，买卖双方都相信支付宝平台，但是通过多重签名技术，步入说三把私钥生成一个地址，只要其中两把私钥同意才能动这个私钥的钱

张健: 我们有一个买方、卖方和担保方，平常正常的交易，就是我们可以把币打到这样多重签名的地址，如果买方、卖方都同时同意，这笔交易完成，这笔交易就不需要中间方或者仲裁方介入，一旦出现纠纷，仲裁方可以决定这笔钱归给谁，但是仲裁方没有可能能把这笔钱拿给自己。

张健: 所以说买方和卖方也不能单方向把钱拿给自己，这就造成了一个从技术层面上杜绝了担保交易互相不可信的问题。
张健: 最后做一下广告，快钱包是国内第一家采用多重签名的比特币钱包，我们采用了两把私钥，一把私钥是保存在我们的服务器上，另一把保存在用户，本地，从本地生成，我们也不存。这样任何一笔交易都要有两私钥签名才能发出去，这种情况下是彻底避免了单点的风险。
张健: 也就是说，其实你很难保证一个系统是没有任何漏洞的，所以说我们希望做到在严重的情况发生的前提下，我们仍然能保证资产安全。当然不代表我们做就是降低我们服务器的安全。只是说我们在根本上杜绝了单点的风险。
张健: 所以说我们的安全方案是远远优于本地的钱包和普通的在线钱包的。企业级就是我刚才讲的，我们可以为不同的角色，因为企业级我们可以分配更多的角色，包括审核员、管理员、发起者，所以可以定制更加安全的钱包解决方案。
主持人: 下面请梆梆安全CTO陈彪谈一下移动金融客户端安全风险分析。
陈彪: 今天时间比较紧，我重点介绍一下我们提供的解决方案，主要就是梆梆加固的技术，梆梆有两代技术，1.0和2.0。安全评估大家都知道了，就是安全分析人员模拟黑客行为对你的系统进行模拟入侵。
陈彪: 带手机端的话，基本有一个规范，就是中国人民银行的客户端软件的支付规范，这个里面就规定了客户软件应该怎么做？然后我们现在基本按照这个软件来做相应的安全评估。这里的话主要是用到的一些工具，如果做安卓开发的就非常清楚了。
陈彪: 可能提一下XPOS，这个是当我们能够，就是装载这个框架之后我们只要写一个插件，这个方法能注入到任何WEB的程序里。比如一个银行给你一个LOGIN，我就能够用这个方法写个插件，直接截取到用户名和密码。
陈彪: 根据这个我们做了安全评估，基本从这几个方面做的，第一方面，程序安全这个方面来做的，包括它一些安装卸载这部分，另外一个就是从代码安全，就是它是不是具备一定的防侏儒这些能力。另外我们检查所有数据的输入到数据的存储以及传输过程中是否存在安全风险？实际这块银行比较关注的。
陈彪: 另外，我们在安卓上会暴露很多的组件，这些都会形成一些风险。当你弹出一个界面的时候，后台木马会弹出一个假的，把当前的覆盖掉，你在这个界面上输入的就会被后台的木马拿走了。
陈彪: 我们可以检查通信安全和业务安全，我们现在做的POC基本能实现转帐，大部分银行客户端转帐过程都能篡改掉，通过侏儒代码方式，直接把客户端转帐的过程的交易修改掉。我们可以检查系统安全这块，就是说如果后台有一个木马，你的程序输入用户密码的时候，会被它直接拿到。
陈彪: 下面我们做一些基本的测试，第一个就是反编译，基本来说这块就是我们的强项了，就是我们后面解决方案，重点是针对安卓反编译来做的。后面没有加固的话，通过反编译这个能拿到前部的原码，最后拿到服务器等算法都能拿到的。
陈彪: 比如对某个金融客户端重打包，基本这种情况加上任意COAD都可以加入的，我们这个图里只是添加了一些广告的代码。还有一种，我们通过代码注入方式，注入新的COAD，注入进去之后能把这个客户端里所有敏感数据，我们想要拦截哪些敏感数据全部都能拿到。
陈彪: 基本上来说，如果说一般金融应用开发客户端来说，它们没有做一些保护的话，在这几项上基本可以做到很容易攻击的。
陈彪: 下面看一下它可能面临的风险，我归纳了几类：用户信息安全，交易的安全，二次打包的风险，有可能通过逆向发现移动端和服务器端的漏洞。像我们通常用手机银行的话会用软键盘输入密码的时候，很多时候软键盘在开发的时候，做的事情会发现，软键盘上只不过把界面掩盖了，但是在内存里你输入的用户密码一直存在的。
陈彪: 简单的HOOK掉一些之后就能拿到用户名和密码。金融的客户端因为很多情况下是外包的，就是程序员开发的水平是有一些弱的，就是很多本地存储上根本就是会有很多问题，我们发现很多客户端在下面一些数据都是没有经过加密存储的。
陈彪: 这个地方会有非常多的方式了，除了刚才那两种还会有像界面劫持，二次打包，这个就是重新打一个APK，比如说在你点确定的时候我把你的密码拿到。另外通过键盘劫持。这个比较有趣的，就是说我们发现某一个密码直接做输入法，这样的话你输入任何东西它都可以拿到的。
陈彪: 另外，当用户做交易的时候实际上也可以被攻击道，这个例子就是我们做的简单的POC，左边这边是试图转给张三这个人，但是我们通过注入之后，直接修改网络函数，这样的话他在服务器中，就是把他的包改掉，在服务器端收到的就是要转给另外一个人，在这种情况下能够直接实现，把他的真个业务交易，转帐的过程修改掉。当然这些东西还是有很多，其他方面都可能存在攻击的风险。

陈彪: 第三个风险就是二次打包。就是说二次打包把原来的程序重新分装一下发布到市场。目前来看，游戏这个领域当然是受二次打包比较严重的。像银行的话，我这里列出两个银行，有些银行基本7%的盗版率。
陈彪: 二次打包以后通常会加一些恶意代码，比如加一些用户名窃取掉的代码，或者加一些恶意的病毒，我们知道在安卓上插一个包，这些病毒的可以通过其他的机子，你不需要点击你的APP，有可能这个病毒也可以工作了。
陈彪: 另外，我们跟一些金融客户，评估它们客户端，发现他们很多的漏洞，这是乌云上报的漏洞，后面是我们发现的客户端代码的一些逻辑上的漏洞。就是说这个漏洞我们分析它的逆向代码以后，发现它的代码URL带一个参数，这个参数明显是一个数值，直接把这个数值往上加，就可以提到其他用户的信息。
陈彪: 还有一些漏洞，这可能有点小，右边的这个漏洞就是说，这个客户端做的比较奇怪，转站的时候一般用双通道来通线，通过短信发一个平台密码给你，但是这个客户端还把这个密码发给了自己的客户端程序，就意味着只要截获网络的流量我就知道你的动态密码是什么了。
陈彪: 上面主要讲的几个金融的风险，下面其实我重点想讲一下这方面的解决方案。我们认为如果你要开发一个真正相对来说安全的客户端，可能要做好几件事情，第一，你的开发人员应该遵循针对移动应用的规范，比如我们刚才讲的，只要我HOOK进去搜查内存就可以搜到所有用户密码，就是因为他存了用户密码。
陈彪: 这个可以采用一些安全的组件，比如我们接触的一些客户，他们里面会采购一些软件ASK，会清场，通过做关键交易之前进行扫描，看机器上有没有安装相应的木马，可能还需要定期找一些专业公司跟客户端做一些安全评估。
陈彪: 第二，我今天重点讲的应用加固。程序经过加固以后，要上线到渠道，但是这个时候其实你也需要监测，看市面上有没有出现对你这个包破解、钓鱼或者山寨的应用。安全加固应用刚才LBE张勇提到挺多，刚才跟他在私下里也交流了一下，他没有拿到我们第二代样本，第一代讲的跟张勇很想象的，就是我们拿到APK加固的时候，把它加密，然后放到APK的资源中去，通过运行的时候修改这个程序的入口，让程序的入口先执行脱壳代码，脱壳代码执行完以后交给真正解完密的代码。
陈彪: 我们在里面会增加一个文件。有一个人对我们进行了分析，他也指出了一个弱点，我们在内存里面是有一个明文的区域的。刚才张勇帮我介绍很多，第一代的问题，不论你怎么做你很难组成大团。我知道爱加密之前会抹掉ADESK，然后针对360最近的，我们最近拿到的一些样本，我们发现360加固的时候，因为有百度脱壳工具，360检查了SPOSE存不存在，如果存在这个程序就不运行。
陈彪: 这个方法有很多了，从我们角度来看，这些方法都是治标不治本的措施，就是你永远无法从本质上解决内存的下载。我们现在给金融客户，包括一些银行的他们在采用，还有游戏客户在采用，真正使用的是第二代的加固技术，这块技术利用了JAVA里有一个特性，JAVA语言在执行的时候并不像C，我们可以对比C1和JAVA加固的应用，像苹果的也做了加密，但是C程序直接跳转到那个地址去执行代码，所以很多的像C的加壳，或者针对非编语言加壳的话要把所有语言解开，然后跳过去执行。
陈彪: 我们知道JAVA有一个特点，它执行某个方法的时候，你这个方法可以现在暂时不存在，它可以通过CLOAD去找这个方法，找到以后做去执行。我们第二代是从整个DESK级别降到方法级别，也就是每个方法我们是单独加密的。当这个方法被虚拟机调用到的时候我们才会按需地去解密，同时我们会保证这个虚拟机在内存，这个解密后的COAD在内存里是不连续的。
陈彪: 另外，我们目前正在加的一些相关的安全措施，比如说会加一些，因为你可以说是我试图重新再内存再构造这个东西，但是这个方法是没有解密的，你只有强制的办法去让我把你所有的方法都解出来，或者修改虚拟机，无非就是加一些方法去处理这些东西。这种我称为方法替换方式。
陈彪: 当虚拟机执行某个方法的时候，这个引擎，我们的加固引擎只解这个方法，只有的话，比如说在我们这个基础上，加固起来以后，在内存里面有可能只有不到四分之一的代码被解开了，还有四分之三的代码实际上是没有不解开的，你只有点，点到了全部路径后才可能被解开，我想这也比较困难。
陈彪: 这是我们现在推出的，就是给客户用的，我们现在还在做第三代的技术，这是基于代码，指令的替换，就是带内存跑的方法的代码，实际上是被我替换过的，而不是真正的原始的代码，你拿到这个代码以后，你还得需要知道我怎么映射这些OPCOAD。
陈彪: 你们发现某个APK里面，看到里面全是空窗版的，基本上是经过我们处理过的架构。这个方法也不需要有什么DEX，就是所有的启动到正常运行的过程中，你通过DEX下载，永远跟你在ADK拿到的Classes Dex是一样的。因为我们做的比较早，所以有些客户并不是对JAVA保护很K，因为我们知道很多的编程框架，像游戏厂商，代码都放在一个里面，我们对这些代码都可以做相关的保护。
陈彪: 这个用法是一般的厂商告诉我们，你要加密哪些文件，APK有哪些文件我们可以给他做。APK文件所有密钥的，就是布局文件全部都加密了。也就是说，这个的防御，比如这种包你单纯拿到了Classes Dex你是拼不回去的，因为那些资源全部经过加密，所以你拿到JAVA层面是没有太大用处的。

陈彪: 这块是我们现在给企业基本上都是采用这套方案的。我们也做企业的SO库，我们也有比较成熟的方案。这块如果说你想做APK特别强劲的话，我们基本上可以把这个APK所有文件做加密，除了系统原来的图标。现在我们给一些银行，如果它采用了ATM的框架我们都会做成ATM5加密。
陈彪: 提一下渠道监控，对银行我们也可以提供这样的服务，这个服务我们会提供一个爬虫，这是到市场上抓第三方的应用程序。通过加速度的分析引擎，这个能够分析出两个APK之间是否相似，我们现在有几个大的引擎，就是说基本信息，像图标或者你的名字，因为我们知道有很多的应用，它就纯粹钓鱼的，我们甚至有一个银行的客户出现过这种情况，它的APP还没开发完也没有上线，但是市面上已经有叫这个名字的APP存在了。
陈彪: 所以基本信息比较我们就能发现出相似的山寨后的，直接是钓鱼的APP。另外它在你包的基础上修改的话，它大部分的资源跟你是基本接近的。也就是说我们通过资源也能够比较出两个APK是否相似的。
陈彪: 最后就是代码纯粹的比较，因为相信代码破解的话，也就是往里面加一个包，真正修改或者在入口的地方修改一下，然后通过代码相似度也能够比较出来，通过这个引擎我们也可以及时关注银行或者金融的程序是否有山寨或者盗版。
陈彪: 总结一下，目前对像手机银行可能面临的这些安全风险，所以我们一般跟客户谈的时候，首先加强客户端开发人员的培训，就是开发的时候考虑到很多安全的问题，然后我们可能会跟一些合作厂商定期的对这些移动的客户端进行评估，评估完了以后我们希望他们使用一些加固，这样的话阻止别人山寨，上线之后我们还要实现监控，看是否出现山寨、钓鱼或者直接盗版的情况。
主持人: 谢谢陈总。接下来请剑心对今天的峰会做一个简单的总结。
剑心: 总结起来，我们知道国内安全最大的问题，不是说我们技术有多差或者什么，是环境问题。环境问题总结起来就是封闭的问题。所以在座的了解到，说这个技术我不能说，那个企业说这个事不能对外说，说了怎么样。这才是真正阻止我们的安全环境的一个根本问题。乌云一直以来，大概四年以来一直以开放的思路做安全，核心的法就是说让社区能够通过开放的方式成长，也很高兴今天看到了13岁的白帽子，得有一个好的社区让他学习、引导他。
剑心: 我觉得是用开放的思路做社区。还有我觉得很多企业，企业是最让人头疼，以前大家认为第一时间不会说，为什么不会说？因为一个企业如果出了安全问题，一定是安全工程师最先知道，他不会反馈给领导，他的领导不会反馈给公司决策层，所以整个企业看起来一片太平，但是我们知道安全，包括今天这么多议题，我们可以看到我们的安全滥到什么程度。
剑心: 对用户也一样，我们希望以开放的思路在安全上好好做一做。今天峰会也是这个思路，峰会上午是白帽子的专场，下午是企业。我们也看出这很符合实际，上午大家很自由，下午慢慢有点拘泥了，就是总是不止痒的感觉。如果乌云通过这个峰会有一个改变。看到下午有很多人在这儿挖人，我挺反感这个事情的，我看到很多好的人，去了阿里巴巴，在网上不说话了，这个ID不活跃了，一到企业里去可能被非技术人员就给限制了，我觉得这是很不好的，而且企业分享的时候会有一些，当然这不能怪企业本身，它的立场是对的，但是我们不希望这样的方式出现，即使出现大家也是买过票的。
剑心: 核心的意思，希望大家一起，包括社区，白帽子，乌云企业，一起把这个事情做的更开放一些，长久一些，而不是把两边人挖走。同时我们想法很活跃，我们做了一个乌云的俱乐部，之前运营了一段时间，现在在798那儿，晚上的话有酒会，大家可以过去聊一聊。将来的话我们还是会以开放的思路做这个俱乐部，每个月我们乌云可能会有一个定期的安全方面的开放的东西，就是除了挖人的都可以过去多聊聊，多学学。HR挖人的就不要过去了。跟极客相关的，跟俱乐部相关的都可以去我们那儿分享。

转载于:https://www.cnblogs.com/miyeah/p/3991035.html