



专题:基于 Android 系统的终端安全

# 一种基于 Android 系统漏洞的通用攻击模型

邓习海,冯维淼,马璐萍,李莹

(中国科学院信息工程研究所,北京 100093)

**摘要:**Android 系统中的各类漏洞给 Android 平台的安全性带来了巨大威胁。漏洞利用技术、移动操作系统安全、Android 生态系统安全逐渐成为研究热点。从漏洞利用的角度出发,分析典型系统漏洞的利用过程,提出了一种 Android 系统漏洞利用的通用模型,并构建了一个漏洞利用有效性评估框架。验证结果表明,该模型能够较好地表述黑色产业链中利用漏洞实施攻击的过程。同时,有效性评估框架可以评估特定漏洞对 Android 生态系统安全性的影响。

**关键词:**Android 操作系统;系统漏洞;漏洞利用;有效性评估

**中图分类号:**TP393

**文献标识码:**A

**doi:** 10.11959/j.issn.1000-0801.2016254

## A general attack model based on Android system vulnerability

DENG Xihai, FENG Weimiao, MA Luping, LI Ying

Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

**Abstract:** Various kinds of vulnerabilities in Android system bring great threats to the platform. The vulnerability exploitation technology, the security of mobile operating systems and the security of Android ecosystem have become a research focus in both industry and academia. The exploitation of several typical system vulnerabilities was analyzed, a general model of Android system vulnerability exploitation was proposed, and a novel evaluation framework for the effectiveness of vulnerability exploitation was presented. The experiment result shows that the general model is able to accurately describe the process of hacker attacks using Android system vulnerabilities. Furthermore, the evaluation framework can assess the influence of system vulnerabilities on the security of Android ecosystem.

**Key words:** Android operating system, system vulnerability, vulnerability exploitation, effectiveness evaluation

### 1 引言

随着移动互联网的快速发展,Android 已成为目前全球用户数最多的移动智能手机操作系统<sup>[1]</sup>。Android 智能终端设备作为用户日常社交通信和移动办公的媒介,存储了大量的用户隐私信息和敏感数据,因此成为了恶意

攻击的主要目标。

近年来,Android 系统不断曝光各类漏洞,给用户带来了巨大的安全隐患。根据 CVE 公开披露漏洞库的统计<sup>[2]</sup>,截至 2016 年 7 月,Android 系统公开披露的漏洞共计 431 个,仅 2016 年上半年公开的系统漏洞数量就有 257 个,如图 1 所示,占比约 60%,呈井喷之势。这些漏洞严重影响了

收稿日期:2016-08-23;修回日期:2016-09-28

基金项目:中国科学院先导专项基金资助项目(No.XDA06010703)

Foundation Item: Strategy Pilot Project of Chinese Academy of Sciences(No.XDA06010703)

Android 平台的安全性,威胁到上层应用软件的数据安全、业务安全和代码安全。然而,过长的 Android 生态系统产业链和严重的 Android 版本碎片化问题,使得漏洞补丁需要经历漫长的修补周期才能有效部署到用户的终端设备上。这在无形之中提升了系统漏洞的利用价值,为攻击者提供了可乘之机。

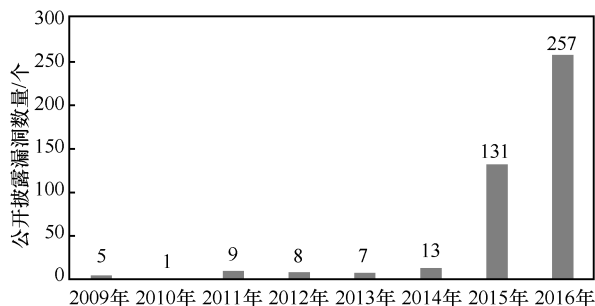


图1 2009–2016 年 Android 系统公开披露漏洞数量统计

持续曝光的公开漏洞成为了黑色产业链的攻击利器。2015 年 7 月,意大利知名监控软件厂商 Hacking Team 的内部信息遭泄露,超过 400 GB 数据被曝光,其中包括 Android、iOS、Windows 等多平台监控系统的源代码,集成了多个 0day 漏洞和公开漏洞的完整的利用代码,以达到实施恶意行为的目的,例如非法提升权限、窃取隐私及敏感信息、监控用户行为、控制音视频设备等,从而实现对全平台的控制和信息收集。Hacking Team 源码泄露事件让原本潜伏于地下的黑客攻击手段公诸于众,其攻击代码的工程化程度高、规范性强、稳定性好、通杀性高,在业界引起极大轰动。从 Hacking Team 事件中不难看出,一旦攻击者利用漏洞攻破系统,所有依赖系统完整性的安全机制都会形同虚设。这一方面揭露了现有安全机制的短板和缺陷,另一方面也促进了业界对各类系统的安全增强研究。

近年来,Android 系统不断引入新的保护机制来提升操作系统的安全性,增强其对抗攻击的能力,缓解系统漏洞带来的危害。自 Android 4.1 版本开始,Android 系统逐步引入了地址空间布局随机化 (ASLR)、数据执行保护 (DEP)、代码段只读、禁止特权执行(PXN)、栈缓冲区保护等漏洞利用缓解技术,在一定程度上提高了攻击难度,减小了漏洞攻击面。Android 4.4 版本起引入 SELinux 内核,使用强制访问控制机制(MAC)替代自主访问控制机制(DAC),通过严格执行安全策略来增强内核安全。Android 5.0 之后进一步严格化 SELinux 安全策略,限制进程的行为能力。这些安全增强机制和缓解技术都为 Android 系统提供

了一定程度的保护。

然而,对应的漏洞利用技术和攻击手段也在不断演化。堆喷射(heap spray)技术、返回导向式编程(ROP)等漏洞利用技术可以在一定程度上绕过安全机制,增加了攻击利用的成功率。现有安全机制需要进一步增强和演化,以弥补缺陷。Android 系统的攻与防是一场持续且激烈的对抗。

从整个 Android 生态系统来看,其安全威胁来自物理攻击、恶意应用、网络威胁、云端数据泄露等多个维度,其中恶意应用造成的危害最为严重。Android 操作系统采用权限机制、沙箱机制、应用签名等方式保护平台的安全性,但是系统漏洞造成的危害往往能够绕过这些安全机制,对系统状态造成严重威胁。

本文选取典型的 Android 系统漏洞进行分析,总结利用漏洞实施攻击的过程,并提出一个 Android 系统漏洞利用的通用模型;然后提出一个漏洞利用有效性评估框架,对这个通用模型进行有效性验证;最后对整个 Android 生态系统的安全防护提出有效建议。

## 2 漏洞利用通用模型

Android 生态系统的安全威胁来自各个方面<sup>[3]</sup>,这些安全威胁可以被攻击者所利用,借助各种技术和工具来实施恶意行为,例如窃取隐私信息、监视用户行为、实行远程控制、恶意破坏系统、恶意扣费等,给 Android 用户的隐私和财产安全带来巨大威胁。传统的安全研究方法按照设备终端、操作系统、应用软件、网络安全进行划分,现有的安全研究中已有一些研究提出了 Android 安全评估模型,但是没有从整个生态系统角度进行评估。

利用 Android 系统中的漏洞进行攻击以谋取经济利益已经形成了成熟的黑色产业链。然而,黑色产业链如何利用系统漏洞实现具体的攻击、这些攻击给 Android 系统造成了何种危害,这些问题并没有得到足够的研究。本文从漏洞利用和攻击角度出发,提出了一种 Android 系统漏洞利用的通用模型,并构建了一个漏洞利用有效性评估框架,对该通用模型进行评估和验证。

### 2.1 Android 系统漏洞利用通用模型

Android 系统漏洞利用的通用模型有助于结构化地描述漏洞利用过程,利用已有攻击行为和研究结果,指导分析、识别漏洞利用过程,进一步对漏洞进行检测和安全预警。本文提出的 Android 系统漏洞利用的通用模型主要包



括三大部分:漏洞资源库、攻击业务资源和攻击实例,如图 2 所示。此外,Android 漏洞利用涉及两个主体:远程控制台和 Android 终端设备。主体和通用模型之间通过互联网进行通信。

## 2.2 模型组成介绍

### (1)漏洞资源库

漏洞资源库的维护是通过漏洞收集和挖掘实现的,不同组织之间可能存在漏洞交易(漏洞信息交换或者售卖购买过程)。漏洞的具体覆盖、映射和利用情况,需要通过漏洞检测或者系统分析获取。漏洞资源库包括系统漏洞信息库、漏洞信息映射库和漏洞利用脚本库。

- 系统漏洞信息库:系统漏洞库是 Android 系统漏洞信息库,是 Android 系统中存在并已经被暴露出来的漏洞信息,系统漏洞库包括公开漏洞集合和未公开漏洞集合。
- 漏洞信息映射库:动态映射库主要映射漏洞、终端可利用脚本的相关性。包括系统漏洞标识,终端标识和脚本利用标识,通过该库,能够查询到某类终端存在的漏洞类型以及可以利用的脚本索引。Android 系统升级、厂商补丁都会对该映射库造成影响。
- 漏洞利用脚本库:漏洞利用库主要是针对某种机型

和某个漏洞编写的利用脚本,运行该脚本,能够实施攻击并获取攻击效果,比如运行提权漏洞的利用脚本能够得到权限提升<sup>[4]</sup>,运行代码执行漏洞的利用脚本能够执行指定的攻击代码。

### (2)攻击业务资源

攻击都涉及对原系统状态的改变。一次完整的攻击过程,可以简单地描述为攻击前提、动作和后果。攻击过程可能涉及一系列单独的攻击行为,每个攻击行为可能是单独对某个系统漏洞的利用。根据攻击的动作和后果,将攻击行为抽象为以下 4 类。

- 扩散:传播,复制。不涉及被攻击系统的状态,但是与攻击主体有关。
- 获取:获取特定信息。
- 破坏:对原系统状态进行改变,并使其状态无法恢复正常状态。
- 控制:在不改变破坏系统状态的情况下,能够影响系统的状态切换。

不同的攻击可能涉及其中一种或者多种行为组合,根据不同行为组合的危害程度,可以对攻击行为进行危害比较。攻击业务资源是工程化实现漏洞利用的抽象形式,本模型包括漏洞利用引擎、攻击业务组件、中间件、攻击策略构造器、分发和植入模块等部分。

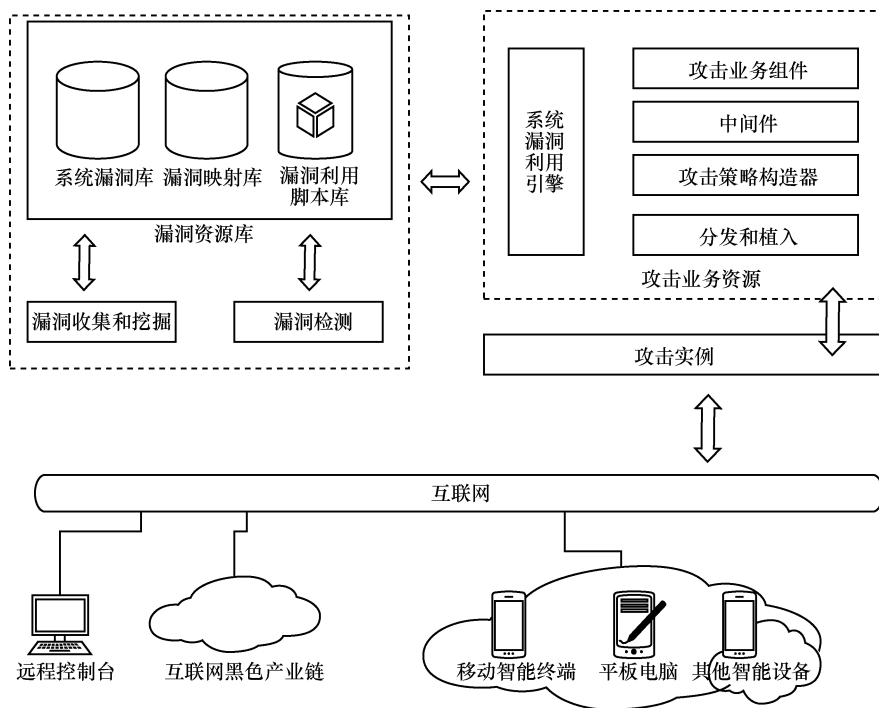


图 2 一种漏洞利用模型

- 系统漏洞利用引擎:攻击的实际发生是根据实际攻击场景定制的,而非直接运行漏洞利用脚本。针对特定的攻击场景,攻击者从漏洞资源库中选取可用漏洞,进行结构化封装和实现,提供给攻击业务使用。
- 攻击业务组件:攻击业务组件是实际攻击的逻辑和业务实现,比如窃听功能、窃取通讯录、关键系统进程注入、远程截屏、破坏系统文件等。
- 中间件:中间件是从漏洞利用到成功调用攻击业务组件时需要用到的通用组件,包括例如权限持久化工具、进程注入组件、进程隐藏、终端信息检测组件,远程控制指令解析器、信息回传、代码加固等技术。
- 攻击策略构造器:攻击过程需要对漏洞利用、中间件和攻击业务组件进行合理的调用,这部分内容抽象为攻击策略构造器。
- 分发和植入:漏洞被恶意利用需要进行漏洞利用脚本的分发、植入或者触发。不同系统漏洞利用过程中,由于系统利用方式不同,采用的触发方式也可能不同。有些攻击的实现,需要在终端上运行相关软件,而有的只需要某种通用文件进行触发。通常分发和植入方式包括,捆绑植入、App 市场分发、诱骗下载、钓鱼 Wi-Fi、BAD-USB、文件触发等。

### (3) 攻击实例

攻击实例包括恶意软件、木马、攻击接口、恶意短信。典型的盗号木马、扣费软件、僵尸网络木马、系统窃听软件等都属于攻击实例集合元素。

## 3 有效性评估

有研究者使用 FUM 评分<sup>[5,6]</sup>计算漏洞对系统安全性的影响,评分包括 3 个度量维度( $f$ 、 $u$  和  $m$ )的加权, $f$  表示漏洞声明周期内从不受漏洞影响的设备比例, $u$  表示能及时进行补丁升级从而不受影响的设备比例, $m$  表示漏洞生命周期中某种设备平均漏洞数量。该评分能够反映某类终端安全性指标,评分越高表示该类设备安全性越高。但是该评分属于对某一类设备安全性的整体评估,并不能表示当前系统安全状态,也没有对于漏洞实际影响的有效性进行评估。针对 FUM 评分存在的问题,提出用于评估漏洞利用模型有效性的 V-A 评估框架,如图 3 所示。

### 3.1 V-A 有效性评估框架

V-A 有效性评估框架采取自上而下、由部分到整体的方法。漏洞利用有效性指标分为两类:漏洞信息和攻击信

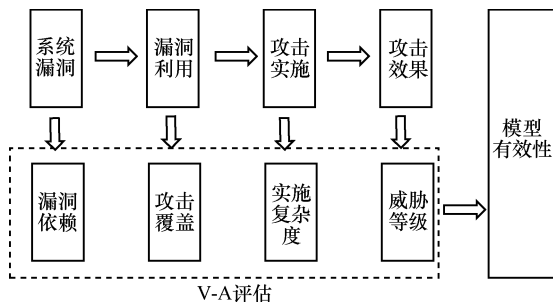


图3 漏洞利用有效性评估框架(V-A 有效性评估框架)

息。漏洞信息主要包括漏洞的详细信息、依赖关系等,攻击信息包括利用该漏洞进行攻击实施的复杂度、覆盖范围和威胁等级。在实际的 Android 生态系统中,漏洞利用具有一定的覆盖范围,即只对特定的某些终端有效。由于 Android 本身版本更新较快,各种定制化系统也层出不穷,各个厂商可利用的漏洞分布也存在差异化,从漏洞覆盖角度进行评估操作性很差。另一种可行的方式是对目标样本进行攻击测试,获取攻击成功的终端比例,得到攻击覆盖范围,从而替代直接的漏洞覆盖情况。本文对漏洞利用有效性评估进行如下建模。

#### (1) 漏洞信息 $V=(D,I)$

$D$  是对其他漏洞的依赖系数,用于评估对其他漏洞的依赖情况。对其他漏洞依赖程度越高,单个漏洞有效性就越低。为了便于计算,这里直接平分漏洞比例,比如单个漏洞实现攻击,则  $D$  为 1,需要  $n$  个漏洞同时发起攻击,则单个漏洞依赖系数为  $1/n$ 。 $I$  是漏洞描述信息,包括漏洞唯一编号、受影响的系统版本和终端类型。

#### (2) 攻击信息 $A=(C,F,S)$

$C$  为实施复杂度,利用漏洞实现攻击的复杂度标识,包括漏洞利用难度、攻击植入或者触发难度的标识。 $C$  最大值为 1,表示无法对目标集合实施攻击;最小为 0,表示对目标集合很容易实施攻击。 $C$  值的计算可以通过对样本集合进行分析获取,即无法成功实施攻击的终端比例。

$F$  为攻击覆盖范围。在不考虑木马分发、漏洞触发条件下,对目标集合进行攻击,能够获取攻击效果的终端比例,攻击覆盖范围与漏洞的覆盖范围有关。

$S$  为攻击威胁等级,是攻击效果实现对系统造成的威胁等级。对攻击效果进行等级划分,定义为 1~10 等级,最高等级记为 10,最低等级记为 1。能够提权至 root 权限的威胁攻击,设置为 10,远程执行代码为 8。



### (3) 漏洞利用有效性评估指数

有效性评估指数根据漏洞信息和攻击信息计算获得。对于单个漏洞有效性计算采用参数相乘得到,即对漏洞依赖系数、漏洞复杂度、攻击覆盖情况以及攻击威胁等级进行评估, $E=D \times (1-C) \times F \times S$ 。

## 4 模型验证

针对第2节提出的攻击模型进行有效性验证。首先通过漏洞检测获取目标样本机型中漏洞分布情况,并选取典型机型进行攻击测试,获取攻击成功的比例,计算攻击覆盖范围。在验证过程中,由于实施复杂度涉及漏洞利用技术、系统版本、具体攻击行为等多方面因素,不易计算,本次验证暂时将其设为0.5。

### 4.1 前期准备

漏洞资源库的建设主要包括系统漏洞库的收集和挖掘、漏洞映射库的维护、漏洞利用脚本库的构建和收集。上述模型验证中,系统漏洞库主要通过查询公开的漏洞进行获取,工作重点在于漏洞映射库的维护和漏洞利用脚本库的构建。

机型—漏洞映射信息可通过针对特定机型进行漏洞检测来收集。关于机型的选择,目前用户手中的 Android 手机品牌、机型及相应使用的 Android 系统版本繁多,如果仅依靠已知的主流品牌(如三星、华为)等的主流机型进行漏洞检测来收集机型—漏洞(特定机型中存在的漏洞信息)映射信息,会遗漏大量市场中被使用的机型,造成机型—漏洞映射信息片面、机型覆盖率不符合实际情况的严重问题。

为了使得收集的机型—漏洞映射信息更加贴近市场中真实的漏洞存在情况,需要对市场上使用率高的机型进行适配,构造专用漏洞利用脚本。在开源项目 AndroidVTS<sup>[7]</sup>的基础上,定制了一个漏洞检测 App:IIE-VTS。该应用可以通过应用分发或联机检测的方式使用。通过将该应用投放市场给实际的用户使用,收集市场中真实存在的机型—漏洞映射信息。

IIE-VTS 目前包含以下 23 个漏洞: CVE-2014-7911、CVE-2014-4322、CVE-2015-6602、CVE-2015-1474、CVE-2015-3829、CVE-2015-1528、CVE-2015-3636、CVE-2014-3847、CVE-2015-7888、CVE-2015-1538-1、CVE-2015-1538-2、CVE-2015-1538-3、CVE-2015-1538-4、CVE-2015-1539-3824、CVE-2015-3828、CVE-2015-3864、CVE-2015-6608、CVE-2015-3825、CVE-

2014-4943、CVE-2014-3153、CVE-2013-6282、CVE-2013-4787、CVE-2011-1149。

IIE-VTS 中增加了漏洞利用验证模块。漏洞检测用于检测 Android 终端设备是否存在漏洞,而漏洞利用验证模块则是通过实施具体攻击,验证存在的漏洞能否被利用。漏洞利用验证模块根据漏洞检测结果,更新远程漏洞映射表并获取漏洞利用脚本(shellcode)进行验证,并比对验证前后系统状态,向服务器返回验证信息。

漏洞利用的方式不尽相同,已有研究<sup>[8]</sup>总结出常用的利用步骤主要如下。

#### (1) 搜集特定机型的信息,创建有利的攻击环境

例如,对漏洞触发可能造成的系统崩溃等情况进行预判,防止在利用漏洞的过程中造成系统崩溃重启或不可预测的情况。设计通过合适的方式将系统引入一个可控的状态下。

#### (2) 编写 shellcode 指令

shellcode 是一段包含攻击代码的机器指令。利用 Android 系统漏洞可以将 shellcode 注入目标设备的内存中。

#### (3) 触发

利用漏洞将 shellcode 注入目标设备与目标进程所在的内存当中,劫持目标进程的执行流程,从而控制目标进程去执行 shellcode。这个过程可能受到漏洞缓解技术的阻碍,例如地址空间布局随机化(ASLR)、数据执行保护(DEP)等,需要借助相应的漏洞利用技术。例如通过堆喷射技术在预测地址上分配可控内存,以对抗地址空间布局随机化(ASLR);借助返回导向式编程(ROP)技术能够对抗数据执行保护(DEP)。

### 4.2 漏洞检测和验证

本次验证过程中,建立的设备目标样本集包含机型数量 28 个,覆盖三星、华为、小米、魅族、酷派、联想等 12 个品牌。被检测的 Android 手机系统版本分布如图 4 所示。

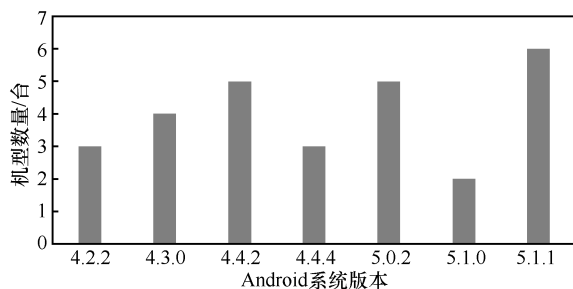


图 4 Android 系统版本分布

利用 IIE-VTS 对上述终端进行检测, 获取漏洞信息。经过统计, 检测漏洞总计 117 次, 版本 4.2.2~4.4.4 漏洞数量最多, 平均终端漏洞数在 6 个以上, 如图 5 所示。

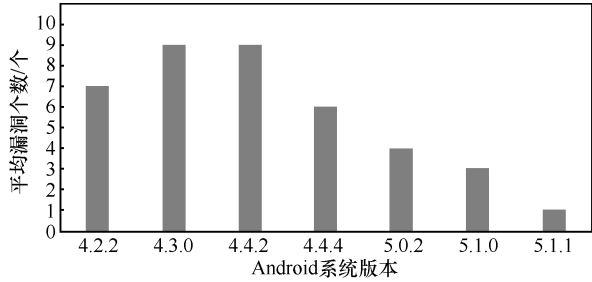


图 5 不同版本平均检测漏洞数量

利用 IIE-VTS 工具, 在检测漏洞信息基础上, 选取具有该漏洞的设备终端进行漏洞利用验证。由于漏洞数量较多, 根据不同终端设备进行漏洞利用脚本编写的工作量较大, 特别是涉及多个漏洞的依赖攻击实现。在本验证过程中, 选取如下 3 个漏洞进行漏洞利用脚本的适配。

表 1 为 3 个漏洞信息, 包括漏洞 CVE 编号、漏洞在 Android 系统中的覆盖情况、实际攻击验证过程中成功验证的终端数目以及攻击效果(威胁等级)。

表 1 漏洞验证情况

| 漏洞编号                    | 受影响 Android 版本 | 实际验证终端数量/台 | 威胁等级     |
|-------------------------|----------------|------------|----------|
| CVE-2015-3864           | 5.0~5.1        | 3          | 远程执行代码   |
| CVE-2014-7911/2014-4322 | 4.4.4 以下版本     | 15         | 提权至 root |
| CVE-2013-6282           | 2.x~4.3        | 11         | 提权至 root |

表 1 中漏洞利用方式及效果如下所述。

(1)Stagefright CVE-2015-3864

- 攻击途径: 用户通过手机 Web 浏览器, 访问嵌入了恶意 MP4 文件的 Web 页面。Android 系统在解析该 MP4 文件的过程中, 触发攻击代码, 反弹 shell 给指定的服务器。服务器获得目标设备的反向 shell 后, 可以执行任意 shell 指令。
- 漏洞利用基本原理: 漏洞代码在解析 MPEG4 文件的“tx3g”标签块时, 处理不当导致整数上溢。通过溢出, 让攻击者获得代码执行的能力, 从而引导程序执行 shellcode。
- 攻击效果: 远程服务器打开一个 nc 服务器, 监听指定端口号的连接。服务器所在的 PC 和手机接入同一个局域网(在实验中为保证正常通信)联网后 PC 的 IP 地址为 192.168.137.188, 机型 2 IP 地址为 192.168.137.15, 二者可以相互 ping 通。手机上通过 Chrome 浏览器访问网址 192.168.137.188/metaphor/web/rce.php(该页面用于渲染 MP4 文件), 结果为 90% 的概率可成功获得反向 shell, 进行远程代码执行。图 6 为服务器获得反向 shell。

```
andy@andy-Lenovo-K4450:~$ nc -p 12345 -vv -l -t
[listening on [any] 12345 ...]
connect to [192.168.137.188] from android-e51f148740a4270d.mshome.net [192.168.137.15] 45019
id
uid=1013(media) gid=1005(audio) groups=1006(camera),1026(drmrnc),1031(media_rn),3001(net_bt_admin),3002(net_bt),3003(linet),3007(net_bw_acct) con
text=ur:mediaserver:so
ls
acct
cache
charger
config
d
data
default.prop
dev
etc
file_contexts
firmware
fstab.hammerhead
init
init.environ.rc
init.hammerhead.diag.rc
init.hammerhead.rc
init.hammerhead.usb.rc
init.rc
init.trace.rc
init.usb.rc
init.zygote32.rc
mnt
persist
proc
property_contexts
res
root
sbin
sdcard
seapp_contexts
selinux_version
sepolicy
service_contexts
```

图 6 服务器获得反向 shell



## (2)CVE-2014-7911 和 CVE-2014-4322 组合攻击

- 漏洞利用基本原理:通过 CVE-2014-7911,将应用权限提升至 system, 有权限访问/dev/qseecom 驱动设备;再利用该驱动特有的漏洞 CVE-2014-4322,将应用权限从 system 提升至 root。
- 利用此漏洞攻击的难点在于:需要机器硬编码信息,需要针对每种机型进行硬编码匹配编写攻击脚本。

## (3)CVE-2013-6282

- 基本原理:Android Linux 内核函数 get\_user()/put\_user()缺少必要的参数检查,没有对输入的目标地址进行验证。get\_user()/put\_user() 是系统调用函数,用于在用户空间和内核空间传递数据。攻击者可利用该漏洞任意读写内核内存,从而提升权限。
- 利用此漏洞攻击的难点在于:在漏洞利用过程中修改内核符号表/proc/kallsyms 需要机型硬编码信息,针对每种机型进行硬编码匹配编写攻击脚本。

## 4.3 有效性分析

针对目标集合的模型有效性计算,计算出 3 个漏洞的有效性评估值  $E$ ,见表 2。从表 2 中可以看出,3 个漏洞有效性评估指数各不相同。有效性评估指数与威胁等级、漏洞覆盖范围关系密切。CVE-2013-6282 的有效性评估指数  $E$  值最高,虽然其暴露时间较长,但是在测试目标样本中漏洞覆盖率高,该漏洞能够直接提权至 root 用户,能完全获取终端系统控制权,漏洞危害极大。CVE-2014-7911、CVE-2014-4322 漏洞有效性评估指标计算过程中引入依赖系数,由于这两个漏洞需要一同配合实现提权攻击,所以设置其依赖系数为 0.5。在有效性评估指数计算过程中,实施复杂度  $C$  主要用来描述攻击者实现攻击的难易程度,由于攻击实施的难易程度与系统漏洞利用特点及攻击者技术储备有关,无法知晓实际攻击者的恶意软件分发、漏洞触发、木马植入的难易程度,所以在计算过程中,将复杂度  $C$  默认设置为 0.5。

## 5 结束语

Android 生态系统安全依赖于及时的系统更新和漏洞补丁,然而目前 Android 终端设备更新并不及时、更新周期过长以及非热门设备被厂商忽略更新。严重的碎片化也给

设备安全性造成更严峻的后果,手机厂商和终端用户之间的信息形成严重不对等——对系统漏洞信息和终端安全状况十分了解的手机厂商和对终端安全状况基本不了解的终端用户。这些客观条件为黑客利用系统漏洞进行终端攻击、用户隐私窃取、木马扩散、远程控制提供了便捷途经。

黑客到底如何利用系统漏洞实现终端攻击以及到底如何评估 Android 生态系统的安全性成为人们关心的问题。从 Android 系统漏洞利用角度出发,提出了一种漏洞利用的通用模型,并对漏洞有效性提出 V-A 评估框架。最后利用开发的 IIE-VTS 工具,对目标样本集合进行模型有效性验证,并计算漏洞利用有效性评估值。通过漏洞有效性 V-A 评估框架,也为从漏洞利用角度来衡量 Android 生态安全状态提供了参考。

针对 Android 生态系统的安全问题,需要从多个维度进行综合解决<sup>[9]</sup>。首先,要加强安全操作系统的研究,从安全模型、安全体系结构着手,对移动终端操作系统安全性设计和实现进行增强;同时完善漏洞挖掘、信息分享和漏洞修补分发的运作机制,让厂商能够在漏洞暴露后及时推送安全补丁和系统升级;最后需要加快相关法律法规的建设,规范终端生产厂商的行为,加强 Android 生态系统的安全建设。

## 参考文献:

- [1] GARTNER. Gartner says worldwide smart phone sales grew 3.9 percent in first quarter of 2016 [EB/OL]. (2016-05-19)[2016-07-01]. <http://www.gartner.com/newsroom/id/3323017>.
- [2] CVE. Google android vulnerability statistics [EB/OL]. (2016-07-29)[2016-07-29]. [http://www.cvedetails.com/product/19997/Google-Android.html?vendor\\_id=1224](http://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224).
- [3] 张嘉宾. Android 应用的安全性研究 [D]. 北京: 北京邮电大学, 2013.  
ZHANG J B. Security research of the Android Apps[D]. Beijing: Beijing University of Posts and Telecommunications, 2013.
- [4] WEI X, GOMEZ L, NEAMTIU I, et al. Permission evolution in the android ecosystem [C]//28th Annual Computer Security Applications Conference, Dec 3 -7, 2012, Orlando, Florida, USA. New York: ACM Press, 2012: 31-40.
- [5] THOMAS D R, BERESFORD A R, RICE A. Security metrics for the android ecosystem[C]//5th Annual ACM CCS Workshop

表 2 漏洞利用有效性计算

| 漏洞编号                    | 依赖系数 $D$ | 实施复杂度 $C$ | 漏洞利用覆盖范围 $F$ | 威胁等级 $S$ | V-A 有效性评估 $E=D \times (1-C) \times F \times S$ |
|-------------------------|----------|-----------|--------------|----------|------------------------------------------------|
| CVE-2013-6282           | 1        | 0.5       | 11/28        | 10       | 1.96                                           |
| CVE-2015-3864           | 1        | 0.5       | 3/28         | 8        | 0.42                                           |
| CVE-2014-7911/2014-4322 | 0.5      | 0.5       | 15/28        | 10       | 1.33                                           |

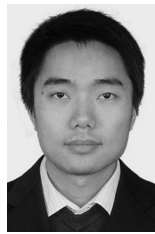
on Security and Privacy in Smartphones and Mobile Devices, October 12–15, 2015, Denver, Colorado, USA. New York: ACM Press, 2015: 87-98.

- [6] AVO. Calculating the score [EB/OL]. [2016-07-29]. <http://androidvulnerabilities.org/>.
- [7] Android-VTS. Android vulnerability test suite[EB/OL]. [2016-07-29]. <https://github.com/AndroidVTS/android-vts>.
- [8] PERLA E, OLDANI M. A guide to Kernel exploitation: attacking the core[M]. New York: Syngress Publishing, 2010.
- [9] 蒋绍林, 王金双, 张涛, 等. Android 安全研究综述[J]. 计算机应用与软件, 2012, 29(10): 205-210.  
JIANG S L, WANG J S, ZHANG T, et al. Overview of the Android safety research [J]. Computer Application and Software, 2012, 29(10): 205-210.

#### [作者简介]



邓习海(1989–),男,中国科学院信息工程研究所助理工程师,主要从事移动安全攻防方面研究工作。



冯维森(1982–),男,中国科学院信息工程研究所工程师,主要从事移动安全攻防方面研究工作。



马璐萍(1986–),女,中国科学院信息工程研究所工程师,主要从事移动安全攻防方面研究工作。



李莹(1992–),女,中国科学院信息工程研究所硕士生,主要从事 Android 系统安全方面研究工作。