DOI:10.13718/j.cnki.xsxb.2018.11.013

基于免疫危险理论的手机恶意软件检测模型①

邹劲松

重庆水利电力职业技术学院 普天大数据产业学院,重庆 永川 402160

摘要:为了提高智能手机恶意软件检测的自适应性和有效性,该文提出了基于免疫危险理论的手机恶意软件检测模型,该模型由 4 个部分组成:数据采集、危险信号生成、共刺激信号生成和预警部分,针对不同的恶意软件,采用微分方法表达危险信号,由自适应抗原提呈细胞产生相应的共刺激信号,最后对恶意软件产生预警. 通过实验验证了该文模型的自适应性和有效性.

关 键 词:智能手机;免疫危险理论;抗原提呈细胞;恶意软件检测

中图分类号:TP391 **文献标志码**:A **文章编号**:1000-5471(2018)11-0078-08

近些年智能手机的数目呈爆发式增长,2016 年我国智能手机网民已达到 6.95 亿,越来越多的人使用智能手机进行上网、聊天和线上支付等操作,这就使得手机用户的个人信息及线上账户财物成为恶意代码软件的目标,导致用户手机崩溃,个人信息丢失,甚至造成严重的经济损失[1-2],因此对智能手机恶意软件的检测研究十分必要[3].

现有的手机恶意代码软件检测技术主要分为静态检测和动态检测. 静态检测技术是对程序代码进行反汇编、系统函数调用等分析,是在恶意代码软件执行前进行的检测[4]. 文献[5]提出了一种权限分析的静态检测方法,该方法通过提取大量正常手机应用程序和恶意程序样本的权限,通过统计样本结果得到两者权限选择的不同点,从而实现对恶意软件的检测. 文献[6]通过挖掘 PE 文件结构信息实现恶意软件静态检测,该方法能够有效实现对加壳和混淆技术恶意软件的检测,避免了现有的基于数据挖掘的检测方法的一些缺陷,如特征选择时存在过度拟合数据的问题. 静态检测方法只能对已经出现的恶意软件实现检测,对于未知恶意软件,则不能进行快速准确地检测.

动态检测技术通过可恢复的环境(如沙箱和手机模拟器等)安装执行软件,实现对恶意代码软件行为特征进行检测与分析[7-8],王盼等[9]提出了一种分布免疫的手机恶意软件检测模型,此算法将改进的反向选择算法和动态克隆选择算法结合,对恶意软件实现检测. Saracino 等[10]提出了一种基于主机的 Android 手机恶意软件检测系统,可以同时分析和关联内核、应用程序、用户及软件包 4 个方面的功能,以检测和发现恶意软件的特征. 动态检测技术在检测的时候特征不稳定,造成检测效率低.

针对以上检测技术缺陷,在研究了已有智能手机恶意软件检测技术的基础上,本文提出了免疫危险理论的智能手机恶意软件检测方法. 该方法根据人工免疫学中的危险理论,采用微分方法表达恶意软件的危险信号,针对不同的危险信号,以抗原提呈细胞自适应产生共刺激信号,最后对恶意软件产生预警.

1 免疫危险理论

生命体的免疫系统拥有一种高度的自适应学习能力,它是一种自适应能力超强的系统,具有十分健全的机制来抵御外来者,保护自身系统内的所有机能. 现代免疫学中将免疫定义为机体对自我或非我的识

① 收稿日期:2017-12-28

作者简介:邹劲松(1975-),男,硕士,副教授,主要从事大数据分析与挖掘及水务管理信息化研究.

别，并排除非我的一种功能.

前人提出的危险理论(Danger Theory，DT)中，免疫系统不是识别 self 和 nonself，而是识别有害的 self 和有害的 nonself. 对免疫识别来说更重要的是入侵的“危险性”，而不是“外来性”. 免疫系统要防范的不是 nonself 而是危险，即危险信号. nonself 不能引发免疫反应，只有危险信号才能诱发效应细胞活化，引起机体发生免疫反应. 危险理论模型如图 1 所示.

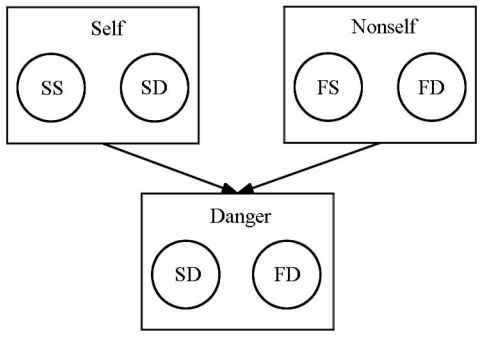

图 1　免疫危险理论模型

由图 1 中免疫危险理论模型可知，真正引起系统异常或者“病变”的是对系统构成威胁的潜在危险. 免疫系统的功能就是自适应地去发现外来物入侵对环境造成的变化，并判断这种变化对环境有害还是有益，并对有害的变化进行抑制(即抑制危险变化)、对有益的变化进行忽略(忽略无害变化)，完成免疫系统的自我学习与自我进化.

一成不变的系统是不会产生任何危险的，只有当系统遭到外来物的攻击时，系统才会打破原有的安全，从而系统各项指标会发生突然的、较大或者微小的变化，这种变化是危险产生的主要诱因. 因此，在智能手机系统中研究危险理论，实质上就是要能及时地发现并定位到引起系统指标变化的异常行为，即我们所说的危险信号，进而实现智能手机自身系统的自我调节与修整，实现自身的稳定.

2　基于免疫危险理论恶意软件检测模型

基于免疫危险理论恶意软件检测模型由 4 部分组成：数据采集部分、危险信号生成部分、共刺激信号生成部分、预警部分.

2.1　危险信号生成

危险被发现得越早，系统被破坏的程度就会越低，如果智能手机被外来物破坏的初始阶段就能被识别出来，智能手机被破坏的程度就会大大降低. 智能手机系统中可能有的危险信号主要包括以下 4 点：

1) 过高的内存占用率、CPU 使用率、存储空间使用率；

2) 蓝牙、WiFi 频繁打开；

3) 过高的短信、彩信发送频率；

4) 手机电池电量迅速下降等.

智能手机系统和生命体一样具有多样性，包括各自不同的系统软件、千变万化的应用软件组合、系统硬件配置不同、以及变化莫测的网络环境等. 个体的不同影响到其危险信号的不同判定，对某个因素上限、下限的界定都有所差异，所以对于不同的个体来说，危险信号也不同. 前面提到过，静止的系统是安全的，只有系统变化了，才有可能产生危险. 因此，对于智能手机危险信号的定义可以从某一时刻系统参数的变化入手，通过微分方法可以获取某一时刻的变化，从而获得系统某一时刻的危险系数. 因此，危险信号的定义可以借鉴微分方法.

系统变量 V 是表征智能手机系统状态的各种系统变量的集合，$V=\{v_i \mid i \in N\}$，其中 v_i 表示某一系统状态变量.

设 R 是观察系统变量的参照系，系统变量集合 $V=G(R)=\{g_1(R), g_2(R), \cdots, g_n(R)\}$，其中 $v_1=g_1(R)$，$v_2=g_2(R)$，$\cdots\cdots v_n=g_n(R)$.

V 相对于参照系 R 的变化可表示为

$$\frac{\mathrm{d}V}{\mathrm{d}R}=\frac{\mathrm{d}\{v_i, v_2, \cdots v_n\}}{\mathrm{d}R}=\left\{\frac{\mathrm{d}v_1}{\mathrm{d}R}, \frac{\mathrm{d}v_2}{\mathrm{d}R}, \cdots, \frac{\mathrm{d}v_n}{\mathrm{d}R}\right\} \tag{1}$$

系统状态是指系统在某一时刻的快照，它是多个系统变量的函数 $SS=f(V)$.

总体的系统状态变化可描述为

$$SC = \mathrm{d}SS = \frac{\mathrm{d}f(V)}{\mathrm{d}V} = \frac{\partial SS}{\partial v_1}\Delta v_1 + \frac{\partial SS}{\partial v_2}\Delta v_2 + \cdots + \frac{\partial SS}{\partial v_n}\Delta v_n \tag{2}$$

危险信号可以理解为恶意的系统变化，这些变化会对系统产生不良的影响. 危险信号就是所有引起系统恶意变化的集合 $DS = \{\mathrm{d}S_i \mid i \in N\}$，它是整个系统所有变化的子集 $DS \subseteq \mathrm{d}V$.

为了更加有效地检测危险信号，本文将手机系统中所有的变化视为可能的恶意变化，即危险信号，并对良性的变化，即对系统无害的变化进行过滤，自然淘汰掉与危险无关的变化，则有

$$DS = \mathrm{d}V = \{\mathrm{d}v_1, \mathrm{d}v_2, \cdots, \mathrm{d}v_n\} = \{\mathrm{d}g_1(R), \mathrm{d}g_2(R), \cdots, \mathrm{d}g_n(R)\} \tag{3}$$

危险信号就是所有引起系统恶意变化的集合，按照微分方法学的理论，本文将采用危险信号的向后差分近似表达对危险信号进行分析. 危险信号的向后差分近似表达如公式(4)、公式(5)所示.

$$\mathrm{d}s_i \approx g_i(R_i) - g_i(R_{i-1}) \tag{4}$$

$$DS \approx \{(g_1(R_i) - g_1(R_{i-1})), (g_2(R_i) - g_2(R_{i-1})), \cdots (g_n(R_i) - g_n(R_{i-1}))\} \tag{5}$$

危险信号是系统中所有恶意变化的集合，故我们需要区分不同的恶意变化并加以记录，本文中危险信号设计数据结构如图2所示.

图2 危险信号的数据结构

其中，ds_name 表示危险信号的名称，表达的信息是危险信号与哪个系统变量相对应，如 ds_name = {CPU 使用率，内存使用率，SMS 发送频率……}. ds_value 是危险信号的取值，即系统变量的变化量. 本文设计的这种危险信号的数据结构可以很好地记录危险信号并加以区分，是十分有效的设计方式.

2.2 数据采集模块设计

在智能手机的运行过程中，无论是恶意软件的运行还是自身正常程序的运行，都会使智能手机的状态发生变化，这些变化体现在智能手机资源使用情况以及性能变化. 本文借鉴 Schmidt 的研究成果，对智能手机系统的以下指标进行监控(表1)，并对表1中所示数据进行采集与分析.

表1 数据采集内容

指 标	描 述	指 标	描 述
battery	电池电量	mms_sent	MMS 发送频率
cpu_usage	CPU 使用率	bluetooth_open	蓝牙打开频率
ram_usage	内存(RAM)使用率	wifi_open	WiFi 打开频率
in_mem_usage	内部存储空间(ROM)使用率	call	电话频率
ex_mem_usage	外部存储空间(SD卡)使用率	conn	网络连接数量
sms_sent	SMS 发送频率		

从表1中可以看出，需要采集的数据十分丰富，囊括了对智能机十分有意义的所有指标. 所以，对这些数据的采集与分析可以很好地鉴别智能手机中的“危险信号”，从而检测出恶意软件是否存在.

2.3 共刺激信号生成模块设计

抗原提呈细胞(Antigen Presenting Cells，APC)是连接先天免疫和适应性免疫的关键细胞，它能感受危险信号、裁决危险状态并触发免疫保护系统. 机体免疫中的 APC 表面主要由组织相容性复合体(major histocompatibility complex，MHC)分子和 Toll 样受体(Toll Like Receptors，TLRs)构成，本文只关注信号的提呈和融合. 遵循 APC 的基本结构和功能，本文构建的人工 APC 结构如图3所示.

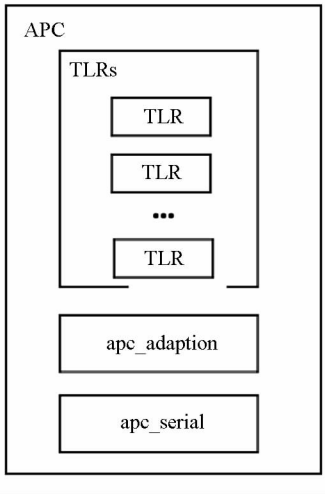

图3 人工 APC 数据结构

其中，TLRs 是 APC 上的 TLR 受体群体，用于识别

不同的危险信号. apc_adaption 被用来判断 APC 是否异常，即是否产生对系统有害的变化. 它随着危险信号的多少而变化，危险信号越多、越频繁，它的值就越大. 当其值超过 TLR 阀值时，APC 产生共刺激信号，否则保持静止. 这种设计与生物体内的 APC 原理一致.

危险信号使得 APC 实现抗体(Ab) 对抗原(Ag) 的提呈，抗体(Ab) 完成对提呈细胞的识别，如果确认抗原危险，则产生免疫应答. 对于抗原提呈，危险信号首先建立危险域，其半径由危险信号的等级确定，等级可表示为 $D=\sum_i w_i\times(DSi-DSi_s)$，$w_i$ 为危险特征的权重，DSi 为不同的危险特征，DSi_s 为危险权重标准值. 对每个危险特征设定一个阈值 T_i，当 $DSi-DSi_s$ 大于阈值 T_i，则认为手机处于危险状态，并发送危险信号. 危险半径为

$$R=\begin{cases}\alpha\times D,\ SMS/MMS\\ \beta\times D,\ 蓝牙\\ \cdots\end{cases}\tag{6}$$

其中 α 和 β 是系数，通过手机安全等级对危险半径调节，之后对手机程序提取检测信息，并编码生成抗原 Ag.

抗体生成过程是对提呈的抗原进行分析，生成相应的抗体，抗原 Ag 和抗体 Ab 之间的距离 $D(Ag, Ab)$ 采用欧氏距离. 当 $D(Ag, Ab)$ 在每个抗体的检测范围内，则能够实现抗原检测.

2.4　预警模块设计

当人工抗原提呈细胞群体产生的共刺激信号达到一定的浓度时，淋巴细胞便被激活，启动适应性免疫. 淋巴细胞的激活预示着系统处于危险状态，表明可能有恶意软件正在运行.

当 apc_adaption 超过阀值时，APC 产生共刺激信号，触发系统产生告警. 危险信号只有达到一定的浓度时才会产生告警，本文将浓度在长度为 T 的窗口 W 内叠加，当浓度达到一定限度时就会触发告警. 浓度的计算公式为

$$CO\_Concentration=\sum CO\_Signal\tag{7}$$

由此可以得出，本文构建的基于免疫危险理论的手机恶意软件检测模型具有自适应性，因为共刺激信号的生成是自适应的. 本文中的危险信号是对系统有害的变化，是系统变化的子集，这些变化的获取是采用微分方法计算所得，几乎不依赖于人工干预. 另外，共刺激信号与告警信号的产生都是由系统自己完成的，并不依赖于人工干预，所以本文提出的模型具有自适应性.

当共刺激信号的浓度大于淋巴细胞激活浓度时，淋巴细胞被激活形成免疫应答，APC 在克隆、杂交以后不断进行变异，不断对 APC 群体进行更新，最终形成自适应模型的 APC 群体，适用于危险信号的检测(图 4).

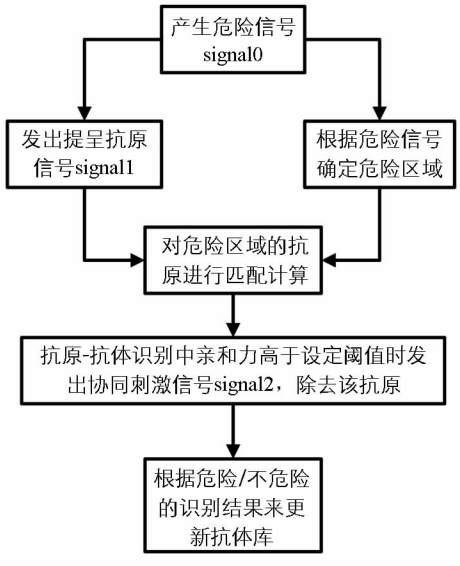

图 4　基于危险理论的免疫算法流程

2.5　模型体系结构与流程

由 2.1—2.4 节对模型各个部分的详细设计描述，可以得到本文模型的总体结构图，如图 5 所示.

在智能手机上随时从系统中获取各种指标，如电池电量、内存使用情况、CPU 使用情况、内存卡使用情况等等，并且采用数学方法对这些数据进行分析(本文采用的是微分方法)，当某一项指标不符合预期时，就可以认为手机上有不正常现象产生，但是不能根据一项指标的异常就断定恶意软件的存在. 所以，只有当几个关联的指标同时异常时(此时断定手机出现危险，即需要产生危险信号)，才需要手机发出告警. 因此，本文提出的变化感知方法在智能手机上对于恶意软件的检测是适用的.

采用本文免疫危险理论的恶意软件检测模型流程如图 6 所示.

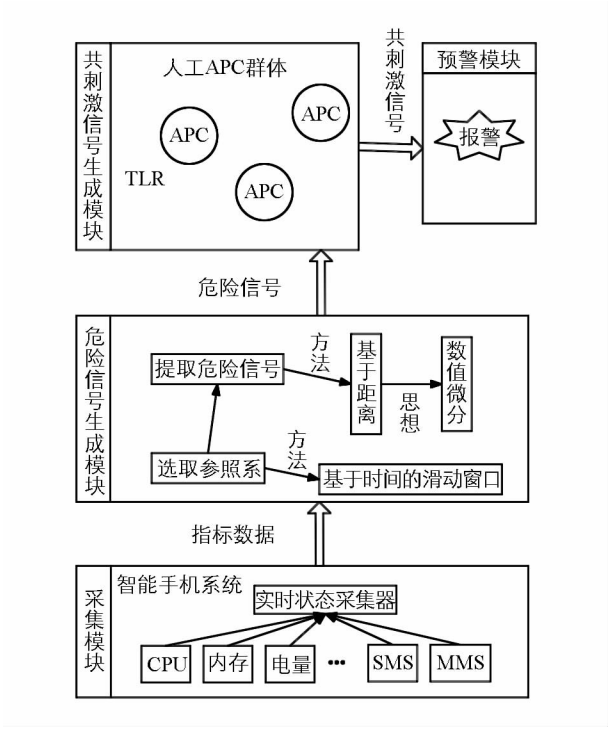

图5 基于变化感知的智能手机恶意软件检测模型框架

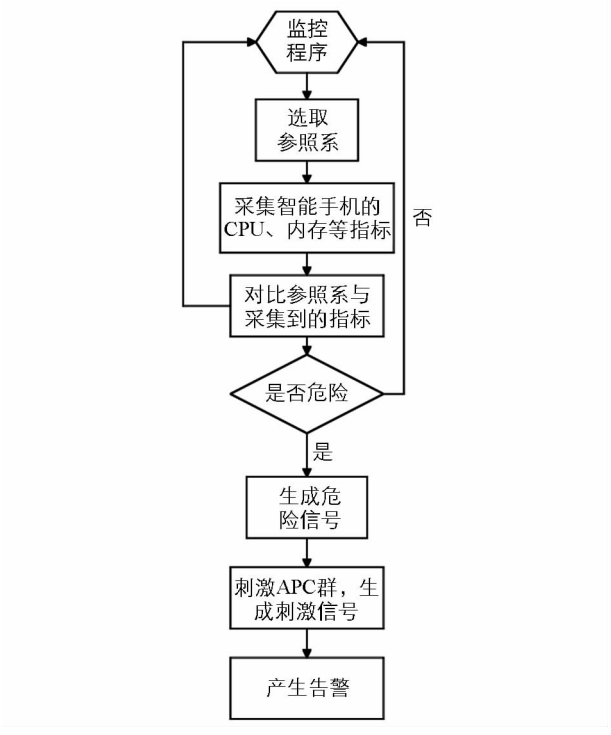

图6 恶意软件检测过程

恶意软件检测模型给我们呈现出了该模型的框架，并且给出了其各个组成部分，那么通过该模型，本文将恶意软件的检测过程设计为如图6所示的既定程序，具体步骤如下：

1) 监控模块从始至终运行在手机上，并不断地检测手机上各种预先设定的指标；

2) 选取参照系；

3) 将监控模块获取的指标数据与参照系作对比，判断是否发生危险；

4) 如果危险则提取危险信号，并发送至APC群，刺激APC细胞群生成刺激信号；

5) 刺激信号发送至告警模块，从而产生告警.

3 实验与分析

本实验主要是为了验证本文所提出的基于变化感知的智能手机恶意软件检测模型的有效性以及自适应性. 为了达到该目标，本文实现了一个基于变化感知的智能手机恶意软件检测原型系统.

3.1 实验设计

为了更好地验证本文设计模型的有效性与自适应性，本实验主要由4个实验组完成，分别为

1) Normal组：采集智能手机处于正常状态，且不运行应用程序情况下的数据；

2) SMS&Call组：采集智能手机运行正常应用程序(发短信、打电话)时的数据；

3) BgServ组：采集智能手机运行植入了BgServ的“经济酒店预订”时的数据(注：BgServ木马具有向控制服务器发送手机隐私信息、发送扣费短信、拦截中国移动和中国联通客服短信等能力)；

4) Replicator组：采集智能手机运行Secret SMS Replicator时的数据(注：Secret SMS Replicator，顾名思义，即“秘密短信复制器”，当该软件被秘密地安装到一部Android手机上时，会在用户不知情的情况下转发用户收到的所有短信). 实验中用到的危险信号的TLR阈值是经验值(表2).

实验采集4组实例数据，分别针对以下4种情况，采集表1中所给出的各项指标变量，根据设置的危险信号阈值，进行恶意软件检测.

第1组(Normal组)：在智能手机系统正常工作状态下，没有其他任意多余的程序运行. 采集从2014年6月1日02：37：20－02：42：19(该时间是Android模拟器中的系统时间)时间段内的数据.

表 2　危险信号对应的 TLR 阈值

指　标	对应阈值	指　标	对应阈值
battery	0.7	sms_sent	0.6
cpu_usage	0.85	mms_sent	0.65
ram_usage	0.7	call	0.8
in_mem_usage	0.9	conn	0.85
ex_mem_usage	0.85		

第 2 组(SMS&Call 组)：在智能手机系统正常工作状态下，没有其他任意多余的程序运行. 系统运行一段时间后，向另外 1 个模拟器发送 1 条短信，并拨打 1 个电话. 数据采集从 2014 年 6 月 1 日 02：50：31—02：55：30.

第 3 组(BgServ 组)：在智能手机系统正常工作状态下，除系统本身随开机启动的程序外，运行植入了 BgServ 的“经济酒店预订”. 数据采集从 2014 年 6 月 2 日 07：20：24—07：25：23.

第 4 组(Replicator 组)：在智能手机系统正常工作状态下，除系统本身随开机启动的程序外，运行 Secret SMS Replicator，并利用另外一个模拟器，向运行该恶意软件的模拟器发送短信. 数据采集从 2014 年 6 月 2 日 11：15：52—11：20：51.

Normal 组和 SMS&Call 组均为系统正常情况下的实验，作为对照组. 完全理想的情况下，这 2 组实验不会产生任何报警. BgServ 组和 Replicator 组 2 组分别运行 BgServ 和 Secret SMS Replicator，应该有报警产生.

3.2　实验结果分析

通过执行实验，采集到的数据结果如表 3 所示. 从表 3 可以看出，恶意软件组的报警次数明显比对照组的报警次数多. 表 4 是 4 组实验的 TLR 数量统计情况，显示了在 4 组实验中用人工 APC 群体计算共刺激信号浓度时，刚开始计算时 TLR 的分布情况，以及计算完毕后 TLR 的分布情况.

表 3　实验结果

组别	参照组		恶意软件组	
	Normal	SMS&Call	BgServ	Replicator
APC 代数	59 代	59 代	59 代	59 代
报警次数	1	13	29	42
报警率	0.02	0.22	0.49	0.71
数量最多的前 3 个 TLR	cpu	cpu	cpu	mem
	mem	call	mem	cpu
	conn	sms	in_mem	sms

由表 3 可知，有恶意软件存在的组，报警率高于参照组，并且通过对报警的观察，可以发现参照组的报警是零星出现的，并无聚集性；且报警时 TLR 的分布呈现出随机的特征，没有明显的规律可循，有很大的可能性是由于噪声导致的. 而在有潜伏软件存在的组，报警相对集中，且有规律可循. 由此可见，本文设计的模型对于检测智能手机中的恶意软件是可行的.

表 4　TLR 分布

	Normal		SMS&Call		BgServ		Secret SMS	
	前	后	前	后	前	后	前	后
battery	55	46	54	38	66	28	56	22
cpu_usage	62	79	63	73	58	158	58	123
ram_usage	61	78	58	58	48	96	74	182
in_mem_usage	58	57	59	51	59	84	58	31
ex_mem_usage	52	57	49	60	49	31	56	29
sms_sent	43	43	52	64	54	37	46	54
mms_sent	57	50	52	36	56	14	59	20
call	52	52	55	64	61	28	42	23
conn	60	58	58	56	49	24	51	16

自适应性分析：由表4可知，在Normal组中刚开始计算时，每个APC上的TLR都是随机装配的，各个TLR的分布很平均，计算完毕后TLR的分布略有变化；总体来看在计算完毕时，TLR的分布还是较为平均的. cpu和mem的TLR数量增多的原因是智能手机在不运行任何额外程序的情况下，CPU使用率和内存使用率仍会偶有变动，而其他指标的状态基本不会有大的变化.

在SMS&Call组中，刚开始计算时TLR均匀分布，计算完毕后TLR的分布发生了变化，总体来看TLR的分布差别不是很大，仍然较为平均，这是因为在SMS&Call组的实验过程中，除了随开机一起启动的程序外，还运行了1次打电话程序及1次短信发送程序，导致电话指标和短信指标略有变动，其它指标的状态变化不大.

在BgServ组中，刚开始计算时TLR均匀分布，计算完毕后TLR的分布发生了较大变化，由于BgServ在运行过程中会占用大量的CPU以及内存资源，并且由于BgServ会将收集到的手机信息存储在手机本地文件中，因此也会占用一定的内部存储空间. 在检测过程中能够检测到CPU、RAM内存、ROM内存变化(即识别到危险信号)的TLR受体被保留，数量逐渐增多，而其他TLR受体被淘汰，数量逐渐减少，导致TLR分布出现较大变化.

在Secret SMS Replicator组中，刚开始计算时TLR均匀分布，计算完毕后TLR的分布发生了较大变化，由于Secret SMS Replicator每发现手机接收到1条短信，都会调用短信发送API，将该短信秘密转发给监控手机，在这个过程中对内存以及CPU的消耗较大. 因此，对于安装了Secret SMS Replicator的手机，在接收到短信的瞬间内存占用率以及CPU占用率的变化非常明显. 同时，SMS的发送频率也有小幅变动. 因此，在计算过程中识别mem,cpu,sms的TLR受体被保留，数量增多，其他受体被淘汰，数量减少.

APC群体会随着当前系统的不同状态进行自适应调整，能识别当前系统状态中危险信号的TLR受体的比例增多，识别不到危险信号的TLR受体比例降低. 因此，对于不同的恶意软件，以及同一软件运行的不同阶段，TLR受体的分布也不相同. 由此可以证明，本文设计的模型在检测恶意软件方面具有自适应性.

误差分析：理论上Normal组是不会产生报警的，但实际上却产生了报警. Normal组产生报警的原因各式各样，并没有任何规律，这也说明Normal组的告警是由不同的事件造成的. SMS&Call组在产生报警时，报警数量最多的3个TLR分别是call(电话拨出频率)、cpu(CPU使用率)、sms(短信发送频率)，这是由于在该组实验中运行了短信发送程序以及电话拨打程序，导致指标 *sms* 的TLR以及指标 *call* 的TLR有少量报警. 另外，由于运行程序时CPU使用率增加，因此指标 *cpu* 的TLR也有少量报警.

初步分析，造成误报警的主要原因有以下几个：

1) 本文选用了9个指标，由于Android模拟器功能的限制，导致某些指标的值是恒定的，不会发生变化(如电池电量)，指标量偏少应该是造成误差的主要原因.

2)启动采集器的瞬间对CPU的占用较为明显，采集器运行时也会占用一定的CPU资源；采集器一边运行，一边将采集到的数据保存到txt文件中，因此对手机的内部存储空间使用率也有一定影响.

实验中虽然存在误报情况，但是对照组的报警次数明显较少，与运行恶意软件的2组相比，还是有很大差距. 对照组报警率都在25%以内，而恶意软件组报警率较高，都在45%以上.

对于误报率降低的改进方法是采用阳性、阴性机制，即避免对自身抗原发生免疫应答，以此来降低误报率，这是未来工作需要研究的内容.

将本文检测模型与瑞星手机杀毒软件进行实验对比，可以得到本文方法检测时间短于瑞星，具体实验结果见表5.

表5 实验对比结果

检测方法	检测时间/s
本文方法	3.407
瑞星杀毒软件	13.843

实验结果表明，本文设计的基于免疫危险理论的手机恶意软件检测模型切实可行，并且具有良好的自适应性.

4 结　论

针对现有智能手机恶意软件检测方法缺乏自适应性的问题，本文提出基于免疫危险理论的手机恶意软件检测模型. 该模型由数据采集、危险信号生成、共刺激信号生成以及预警等 4 个部分构成，该模型根据人工免疫学中危险理论，采用微分方法表达恶意软件的危险信号，针对不同的危险信号和阈值，以抗原提呈细胞自适应产生共刺激信号，最后对恶意软件产生预警. 实验结果表明，本文模型对于智能手机恶意软件的检测可行有效.

本文模型基于 Android 手机系统，下一步将实现具有不同安全机制的 IOS 和 Windows 系统的恶意软件模型.

参考文献：

[1] 马晋杨，徐　蕾. 基于 Android 系统的手机恶意软件检测模型 [J]. 计算机测量与控制，2016，24(1)：156－158.

[2] 胡迎春，熊　江. 基于 SMS/MMS 和 Bluetooth 的智能手机恶意软件传播模型研究 [J]. 西南师范大学学报(自然科学版)，2016，41(9)：107－112.

[3] 李宏鸾. Android 平台的恶意代码检测技术的研究 [D]. 成都：电子科技大学，2013.

[4] 蔡志标，彭新光. 基于系统调用的 Android 恶意软件检测 [J]. 计算机工程与设计，2013，34(11)：3757－3761.

[5] 周裕娟，张红梅，张向利，等. 基于 Android 权限信息的恶意软件检测 [J]. 计算机应用研究，2015，32(10)：3036－3040.

[6] 白金荣，王俊峰，赵宗渠. 基于 PE 静态结构特征的恶意软件检测方法 [J]. 计算机科学，2013，40(1)：122－126.

[7] NARUDIN F A，FEIZOLLAH A，ANUAR N B，et al. Evaluation of Machine Learning Classifiers for Mobile Malware Detection [J]. Soft Computing，2016，20(1)：343－357.

[8] CANFORA G，MEDVET E，MERCALDO F，et al. Acquiring and Analyzing App Metrics for Effective Mobile Malware Detection [C]//Proceedings of the 2016 ACM on International Workshop on Security and Privacy Analytics. New Orleans：ACM，2016.

[9] 王　盼，梁意文. 手机恶意软件检测的分布式免疫模型 [J]. 计算机工程与应用，2016，52(16)：30－35.

[10] SARACINO A，SGANDURRA D，DINI G，et al. Madam：Effective and Efficient Behavior-Based Android Malware Detection and Prevention [J]. IEEE Transactions on Dependable and Secure Computing，2016，15(1)：83－97.

Mobile Malware Detection Model Based on Immune Danger Theory

ZOU Jin-song

Putian Big Data Industrial College，Chongqing College of Water Resources and Electric Engineering，Yongchuan Chongqing 402160，China

Abstract：In order to improve the adaptability and effectiveness of malware detection in mobile phones，a mobile malware detection model based on immune danger theory has been proposed in this paper. The model consists of four parts：data acquisition part，hazard signal generation part，co-stimulation signal generation part and warning part. Using differential method to express different dangerous signals，then the model produce corresponding co-stimulatory signals according to adaptive antigen presenting cells，and finally give early warning to malware. The experiment verifies the adaptability and effectiveness of this model.

Key words：mobile phone；immune danger theory；antigen presenting cells；malware detection

责任编辑　夏　娟