



(12) 发明专利申请

(10) 申请公布号 CN 113031995 A

(43) 申请公布日 2021.06.25

(21) 申请号 202110555122.7

G06F 16/955 (2019.01)

(22) 申请日 2021.05.21

(71) 申请人 北京神州泰岳智能数据技术有限公司

地址 100082 北京市海淀区海淀大街34号8层814室

(72) 发明人 张美慧 李晓宇 周宇 张焱 陈雪

(74) 专利代理机构 北京超凡宏宇专利代理事务所(特殊普通合伙) 11463

代理人 余菲

(51) Int. Cl.

G06F 8/65 (2018.01)

G06F 8/71 (2018.01)

G06F 11/36 (2006.01)

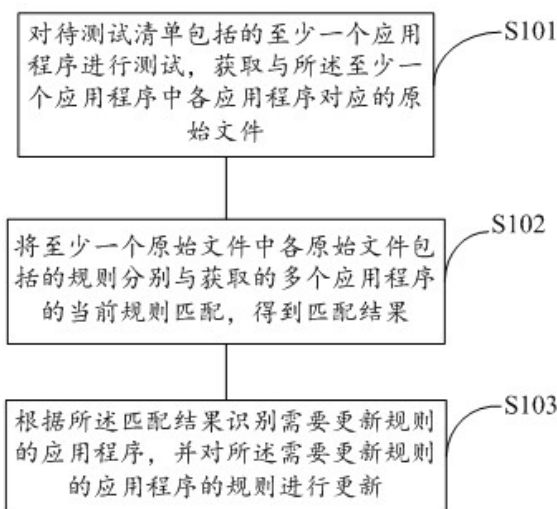
权利要求书2页 说明书9页 附图4页

(54) 发明名称

一种更新规则的方法、装置、存储介质以及电子设备

(57) 摘要

本申请实施例提供一种更新规则的方法、装置、存储介质以及电子设备,所述更新规则的方法包括:对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录测试应用程序获取的规则,所述规则包括上网规则;将至少一个原始文件中各原始文件包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果;根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。本申请的一些实施例通过自动识别需要更新的规则,相对于现有的人工确认规则是否需要更新的技术方案相比,明显提升了数据处理的速度和效果。



1. 一种更新规则的方法,其特征在于,所述方法包括:

对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录测试应用程序获取的规则,所述规则包括上网规则;

将至少一个原始文件中各原始文件包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果;

根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

2. 如权利要求1所述的方法,其特征在于,所述根据所述匹配结果识别需要更新规则的应用程序之后,所述方法还包括:

根据所述需要更新规则的应用程序得到结果清单;

获取所述结果清单和所述待测试清单中包括的相同应用程序的规则,得到更新规则;

将所述更新规则存入数据库。

3. 如权利要求1所述的方法,其特征在于,所述对待测试清单包括的至少一个应用程序进行测试,包括:

采用测试工具模拟真实用户运行所述至少一个应用程序中的各应用程序,获取与所述各应用程序对应的多条上网请求;

根据所述多条上网请求得到多条上网规则,其中,所述原始文件包括所述多条上网规则。

4. 如权利要求1或3所述的方法,其特征在于,所述原始文件的多个字段包括:测试工具生成的编号、源地址、目的地址、协议类型、运行应用程序期间产生上网请求的域名、url地址以及UA特征值,其中,所述的上网规则包括与所述各应用程序对应的运行应用程序期间产生的上网请求的域名、url地址以及UA特征值。

5. 如权利要求1所述的方法,其特征在于,所述当前规则是通过以下过程获取的:

获取当前需要进行版本更新测试的应用程序清单;

从所述应用程序清单中提取有效的应用程序和所述有效的应用程序的编号;

从所述有效的应用程序中提取至少一个有效版本号;

基于所述至少一个有效版本号提取有效规则生成所述当前规则。

6. 如权利要求2所述的方法,其特征在于,所述根据所述匹配结果识别需要更新规则的应用程序,包括:

当确认第一原始文件包括的所有规则均不与所述当前规则中的任意一条规则匹配时,则识别所述第一原始文件为需要更新规则的应用程序;

将第一应用程序作为所述结果清单中的一个程序。

7. 如权利要求1所述的方法,其特征在于,所述将至少一个原始文件中各原始文件包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果,包括:

将所述当前规则以应用程序为单位划分为多组;

将所有原始文件加载至临时数据库表中;

将以所述应用程序为单位分组后的数据,逐条和所述临时数据库表中所有数据进行对比,获取匹配结果;

所述根据所述匹配结果识别需要更新规则的应用程序,包括:

如果一组数据包括的所有规则都匹配不到任何原始文件,判定所述一组数据对应的应用程序需要进行规则更新,则将所述一组数据对应的应用程序作为结果清单中的一项。

8. 一种更新规则的装置,其特征在于,所述装置包括:

测试模块,被配置为对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录对应应用程序的规则,所述规则包括上网规则;

规则匹配模块,被配置为将至少一个原始文件中各原始文件包括的规则分别与预先获取的多个应用程序的原始规则匹配,得到匹配结果;

识别模块,被配置为根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

9. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时可实现权利要求1-7中任意一项权利要求所述的方法。

10. 一种电子设备,包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序,其中,所述处理器执行所述程序时可实现权利要求1-7中任意一项权利要求所述的方法。

一种更新规则的方法、装置、存储介质以及电子设备

技术领域

[0001] 本申请涉及大数据地址库校验领域,具体而言,涉及一种更新规则的方法、装置、存储介质以及电子设备。

背景技术

[0002] 近几年,大数据行业急速发展,各种各样的产品和应用层出不穷,数据量和数据累积情况也愈发壮大。随着数据资源的积累,新数据的入库必定需要经过大量的验证和对比后,才能进行入库。为提高资源整合效率和数据质量,为我国大数据产业的健康发展打下基础,为数据安全应用提供保障,同时促进大数据交易等新兴服务模式规范发展。加强大数据地址库规则新增速度和入库质量,对推动我国大数据产业进程,加快技术与标准的相互融合,落实大数据战略具有重要意义。

[0003] 当前获取互联网上网规则,主要利用抓包工具对应用程序(app)及网页(wap端、pc端)进行测试,测试完成后获得完整的上网日志,对获取到的规则进行分析和整理,筛选出有利用价值的信息,对以上信息进行适当的截取和模糊,从而形成需要的规则信息。随着测试覆盖面的扩大和应用程序(app)的版本更新,测试量和人工对比规则的工作量日益提高。测试范围的不断扩大、应用程序的版本迭代等因素都将成为重新测试的因素。当有一款应用程序(app)出现新的版本时,就要对此应用进行测试、筛选、分析比对、入库,随着规则量的增加,分析对比工作日趋繁琐。

[0004] 因此如何提升应用程序的规则对比效率成了亟待解决的技术问题。

发明内容

[0005] 本申请实施例的目的在于提供一种更新规则的方法以、装置、存储介质以及电子设备,通过本申请实施例公开的方法和装置,可以将杂乱的数据变得更加清晰和标准,避免或节省人工费用,减少维护成本,降低运行成本,提高公司业务效率。

[0006] 第一方面,本申请的一些实施例提供一种更新规则的方法,所述方法包括:对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录测试应用程序获取的规则,所述规则包括上网规则;将至少一个原始文件中各原始文件包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果;根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

[0007] 本申请的一些实施例通过自动识别需要更新的规则,相对于现有的人工确认规则是否需要更新的技术方案相比,明显提升了数据处理的速度和效果。

[0008] 在一些实施例中,所述根据所述匹配结果识别需要更新规则的应用程序之后,所述方法还包括:根据所述需要更新规则的应用程序得到结果清单;获取所述结果清单和所述待测试清单中包括的相同应用程序的规则,得到更新规则;将所述更新规则存入数据库。

[0009] 本申请的一些实施例通过对比待测试清单中的应用程序是否与需要更新的应用

程序是否相同来筛选需要规则更新的目标应用程序,减少了数据处理的量并提升了数据处理速度。

[0010] 在一些实施例中,所述对待测试清单包括的至少一个应用程序进行测试,包括:采用测试工具模拟真实用户运行所述至少一个应用程序中的各应用程序,获取与所述各应用程序对应的多条上网请求;根据所述多条上网请求得到多条上网规则,其中,所述原始文件包括所述多条上网规则。

[0011] 本申请的一些实施例通过模拟真实用户获取上网请求来提取测试待测试文件产生的原始文件包括的规则,保证了原始文件的覆盖率。

[0012] 在一些实施例中,所述原始文件的多个字段包括:测试工具生成的编号、源地址、目的地址、协议类型、运行应用程序期间产生上网请求的域名、url地址以及UA特征值,其中,所述的上网规则包括与所述各应用程序对应的运行应用程序期间产生的上网请求的域名、url地址以及UA特征值。

[0013] 本申请的一些实施例通过为每个应用程序分配对应的编号并记录各应用程序产生的上网请求携带的地址类信息获取各应用程序的原始规则,以使得后续地址库中的当前规则可以与这些原始规则比较进而确认是否需要更新地址库中的当前规则。

[0014] 在一些实施例中,所述当前规则是通过以下过程获取的:获取当前需要进行版本更新测试的应用程序清单;从所述应用程序清单中提取有效的应用程序和所述有效的应用程序的编号;从所述有效的应用程序中提取至少一个有效版本号;基于所述至少一个有效版本号提取有效规则生成所述当前规则。

[0015] 本申请的一些实施例中从产生的应用程序app最新版本号下的规则中,滤除已下线规则,保留在用规则生成地址库中的当前规则,这样可以减少和原始文件中包含的原始规则的数据量,进一步达到提高运行效率的目的。

[0016] 在一些实施例中,所述根据所述匹配结果识别需要更新规则的应用程序,包括:当确认第一原始文件包括的所有规则均不与所述当前规则中的任意一条规则匹配时,则识别所述第一原始文件为需要更新规则的应用程序;将所述第一应用程序作为所述结果清单中的一个程序。

[0017] 本申请的一些实施例通过这个标准保证了识别需要更新规则的应用程序的判断标准的一致性。

[0018] 在一些实施例中,所述将至少一个原始文件中各原始文件包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果,包括:将所述当前规则以应用程序为单位划分为多组;将所有原始文件加载至临时数据库表中;将以所述应用程序为单位分组后的数据,逐条和所述临时数据库表中所有数据进行比对,获取匹配结果;所述根据所述匹配结果识别需要更新规则的应用程序,包括:如果一组数据包括的所有规则都匹配不到任何原始文件,判定所述一组数据对应的应用程序需要进行规则更新,则将所述一组数据对应的应用程序作为结果清单中的一项。

[0019] 本申请的一些实施例对原有规则按照应用程序进行分组,然后以各分组为单位分别与测试得到的原始文件中的规则进行匹配,提升匹配速度。

[0020] 第二方面,本申请的一些实施例提供一种更新规则的装置,所述装置包括:测试模块,被配置为对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用

程序中各应用程序对应的原始文件,其中,所述原始文件用于记录对应应用程序的规则,所述规则包括上网规则;规则匹配模块,被配置为将至少一个原始文件中各原始文件包括的规则分别与预先获取的多个应用程序的原始规则匹配,得到匹配结果;识别模块,被配置为根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

[0021] 第三方面,本申请的一些实施例提供一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时可实现上述第一方面所述的方法。

[0022] 第四方面,本申请的一些实施例提供一种电子设备,包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序,其中,所述处理器执行所述程序时可实现上述第一方面所述的方法。

附图说明

[0023] 为了更清楚地说明本申请实施例的技术方案,下面将对本申请实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本申请的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0024] 图1为本申请的一些实施例提供的更新规则的方法流程图之一;
图2为本申请实施例提供的更新规则的方法的流程图之二;
图3为本申请实施例提供的更新规则的方法流程图之三;
图4为本申请实施例提供的更新规则的方法流程图之四;
图5为本申请实施例提供的更新规则的装置的组成框图;
图6是本申请实施例提供的电子设备的组成框图。

具体实施方式

[0025] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行描述。

[0026] 应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。同时,在本申请的描述中,术语“第一”、“第二”等仅用于区分描述,而不能理解为指示或暗示相对重要性。

[0027] 本申请的实施例涉及大数据地址库校验领域,利用本申请实施例公开的方法和装置,可以将杂乱的数据变得更加清晰和标准,避免或节省人工费用,减少维护成本,降低运行成本,提高公司业务效率。本申请的实施例提供了一种自动对比现有规则和新生规则的工具,从而降低人工判断的错误率和提高测试效率。本申请实施例公开的方法和装置可以有效的解决由于人工判断的错误率稿和测试效率低下的问题,本申请一些实施例的规则对比包括历史规则(即当前已有规则)的对比、UA规则的对比等。例如,UA规则就是从user_agent字段中提取出的有效信息。

[0028] 本申请的一些实施例要解决的技术问题包括:对比新规则和已有规则,相同规则无需重复入库,不同的规则新增到数据库中,简化地址库规则入库流程。本申请一些实施例公开的更新规则(即更新应用程序对应的规则)的方法包括:首先生成对比地址库(即由原始规则组成地址库),然后将需要比对的规则对应的原始数据(原始数据为对待测试应用程

序模拟真实用户的操作进行测试的过程中提取的到的规则)上传至指定目录,使用地址库和原始数据进行比对,地址库规则取出一条跟原始数据进行比对,如果地址库内规则能够与原始数据相匹配,则无需更新该应用程序app的规则,如果地址库内规则无法匹配到原始数据,那么该应用程序app需要更新规则。程序执行后生成需要更新规则的应用程序app的结果清单,将待测清单和结果清单进行对比,出现重复的应用程序app,则将重复的应用程序app的规则更新进数据库即可。

[0029] 请参看图1,图1为本申请的一些实施例提供的一种更新规则的方法,该方法包括:S101,对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录测试应用程序获取的规则,所述规则包括上网规则;S102,将至少一个原始文件中各原始数据包括的规则分别与获取的多个应用程序的当前规则匹配,得到匹配结果;S103,根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。相对于现有的人工确认规则是否需要更新的技术方案相比,明显提升了数据处理的速度和效果。

[0030] 需要说明的是,S101的一个原始文件与一个待测试的应用程序对应,且原始文件为模拟真实用户访问应用程序得到的原始数据包括的多条规则。为了减少数据处理量并提升数据处理的速度,在本申请的一些实施例中,在S103之后,更新规则的方法还包括:根据所述需要更新规则的应用程序得到结果清单;获取所述结果清单和所述待测试清单中包括的相同应用程序的规则,得到更新规则;将所述更新规则存入数据库。

[0031] 下面示例性阐述上述各步骤。

[0032] 在本申请的一些实施例中,S101包括:采用测试工具模拟真实用户运行所述至少一个应用程序中的各应用程序,获取与所述各应用程序对应的多条上网请求;根据所述多条上网请求得到多条上网规则,其中,所述原始数据包括所述上网规则。需要说明的是,测试员模拟用户操作,如登录、浏览、下单、咨询等,每一个动作都会产生请求,产生的这些请求就是原始规则。

[0033] 在本申请的一些实施例中,S101生成的原始文件的多个字段包括:测试工具生成的编号、源地址、目的地址、协议类型、运行应用程序期间产生上网请求的域名、统一资源定位符url(Uniform Resource Locator)地址以及用户代理UA(User Agent)特征值,其中,所述的上网规则包括与所述各应用程序对应的运行应用程序期间产生的上网请求的域名、url地址以及UA特征值。

[0034] 作为一个具体示例,S101包括:

准备待测清单对待测应用程序app进行测试。准备应用程序app待测清单,在测试手机上下载应用程序app准备测试。例如,待测清单中包括了拼多多、励步英语、导游证考试、包英语等共43款应用程序等待测试,在测试机器(手机)上下载好以上应用程序app,等待测试。

[0035] 使用测试工具(例如,测试工作包括:wireshark或者fiddler等)测试待测清单内的应用程序app,并且保留测试结果对应的原始文件(例如,原始文件格式的格式为.csv,原始文件包括原始数据)。

[0036] 作为一个示例,测试原始文件得到的测试结果的包括的各字段以及各字段的含义如下:

No.:测试工具自动生成的编号。

[0037] Source:源地址。

[0038] Destination:目的地址。

[0039] Protocol:协议类型。

[0040] Host:域名,app使用期间产生的请求。

[0041] url:规则,app使用期间产生的请求。

[0042] info:完整信息,app使用期间产生的请求。

[0043] UA:UA特征值,app使用期间产生的请求。

[0044] Refer:http请求中的referer。

[0045] Cookie:cookie值。

[0046] Domain:域,测试工具自动产生。

[0047] 需要说明的是,在测试待测试清单中的应用程序的过程中注意在测试工具中添加过滤条件,减少噪音数据,提升后期数据处理质量。如:`http.host and ! (http.request.uri contains *) and! (http.request.uri contains .jpg) and! (http.request.uri contains .png)`,以上语句可以过滤掉噪音数据和图片数据。测试规程中注意要模拟真实用户操作app,保证原始数据的覆盖率。测试中要注意测试结果(即原始文件的)保存格式为.csv。

[0048] 在本申请的一些实施例中,S102涉及的当前规则是通过以下过程获取的:获取当前需要进行版本更新测试的应用程序清单;从所述应用程序清单中提取有效的应用程序和所述有效的应用程序的编号;从所述有效的应用程序中提取至少一个有效版本号;基于所述至少一个有效版本号提取有效规则生成所述当前规则。本申请的一些实施例中从产生的应用程序app最新版本号下的规则中,滤除已下线规则,保留在用规则生成地址库中的当前规则,这样可以减少和原始文件中包含的原始规则的数据量,进一步达到提高运行效率的目的。

[0049] 在本申请的一些实施例中S102涉及的当前规则即地址库中的规则,具体地,准备地址库用来对比原始数据。作为一个示例,准备地址库需要手动执行地址库生成脚本,直接进入脚本放置目录,执行生成地址库的脚本即可,脚本执行完毕后,就已经生成当前规则的地址库。

[0050] 作为一个示例,根据当前规则生成地址库的过程如图2所示。图2的根据当前规则生成地址库的方法包括:

S201,获取应用程序app版本更新待测清单。例如,获取当前需要进行版本更新测试的清单,数据库中存储目前所有需要进行测试的app相关信息,每日自动更新。例如,检测得到需要进行版本更新测试的清单中包括:亚马逊、民生、丁香医生等应用程序。

[0051] S202,提取有效应用程序app。

[0052] 提取S201得到的版本更新待测清单列表中有效应用程序app的应用名称和应用id,目的是为了滤除失效应用程序app,只保留当前有效的应用程序app规则,以达到减少数据量,提升数据处理速度的目的。例如:一款应用程序app名称为摩比思维,该应用程序app在数据库中被标记为失效,那么在生成地址库过程中,应当将该app滤除。

[0053] S203,提取有效版本号。

[0054] 提取S202筛选得到的有效应用程序app的有效版本号,即在数据库中提取某一应用程序app最新版本号测试产生规则;因为不同版本下可能产生相同或重复规则,所以取出当前应用程序app的最新的规则,即可减少数据量,从而达到提升运行效率的目的。例如:名称为于斯课堂的应用程序app,4.1.9版本号下没有规则,4.1.8和4.1.6版本下有规则,那么将4.1.8版本下的全部规则取出,作为于斯课堂的当前规则。

[0055] S204,提取有效规则。

[0056] 提取当前最新版本号下的有效规则,即在S203产生的应用程序app最新版本号下的规则中,滤除已下线规则,保留在用规则,进一步达到提高运行效率的目的。

[0057] S205,生成地址库。

[0058] 数据库生成脚本中调用了jar包文件,分别生成UA地址库和基础地址库。UA地址库字段有n_id、ua_feature、bqz_id、bqz_name,字段含义分别为标识id、ua规则、应用id、应用名称。基础地址库字段有n_id、domain_one、brand、domain_comp、service_name、rule、rule_type、rule_type_level、is_app_url、bqz_id、bqz_name,字段含义分别为表示id、一级域名、应用开发商、完整域名、应用名称、规则、规则类型、规则优先级、是否是应用产生的规则、应用id、应用名称。

[0059] 在一些实施例中,S102包括:将所述当前规则以应用程序为单位划分为多组;将所有原始文件加载至临时数据库表中;将以所述应用程序为单位分组后的数据,逐条和所述临时数据库表中所有数据进行比对,获取匹配结果;所述根据所述匹配结果识别需要更新规则的应用程序,包括:如果一组数据包括的所有规则都匹配不到任何原始数据,判定所述一组数据对应的应用程序需要进行规则更新,则将所述一组数据对应的应用程序作为结果清单中的一项。

[0060] 作为一个示例,S102包括:

将S101测试应用程序app时产生的结果文件上传至指定的ftp目录。

[0061] 执行规则对比程序。进入脚本存放目录,手动执行脚本,等待对比结果生成。对比过程的原理:上述方法生成的地址库内包含多条规则,原始数据内包含本次测试的应用程序产生的所有请求信息,使用地址库内的规则与所有原始数据进行对比,地址库内的规则能够匹配到原始数据中的一条或多条规则时,说明地址库内规则有效,能够用此规则识别到指定应用程序app应用;当地址库内规则无法匹配到原始数据时,说明此规则无法用来识别应用程序app,原因有二,一是app应用本身无法产生有效规则即无法识别,二是规则失效需要向地址库补充新的规则来识别app应用。

[0062] 需要说明的是,对比过程的执行逻辑的实现过程包括:将S101的原始数据对应的文件加载到临时数据库表中,此步的目的是将待测试的43款app所产生的全部原始数据生成为一个数据库表,用来进行下面的数据比对;数据分组,将根据当前规则生成的地址库表中的规则,以应用程序app为单位分组;数据比对,将以应用程序app为单位分组后的数据,逐条和地址库表中所有数据进行比对,能够匹配成功全部或部分,即可判定为规则有效,规则匹配不到任何数据,即可判定为规则失效;如果一个应用程序app的所有规则都匹配不到任何原始数据,判定app需要进行规则更新。例如:华数tv这款app,在地址库中有8条有效规则,取出第一条规则,用第一条规则和原始数据进行匹配,匹配成功,说明该规则有效,即用这条规则,可以识别到这个app;如果8条数据中有6条数据有效,说明现有规则能够达到识

别该app的目的,则该app无需更新规则;如果8条数据中没有任何一条有效,说明现有规则无法达到识别该app的目的,则该app需要重新入库有效规则。循环比对,直至所有规则比对完成。S102的匹配结果即未匹配成功的应用程序清单。

[0063] 例如,执行S102后,匹配结果包括575条数据,说明本次程序运行,地址库内有575个应用程序app无法匹配到原始文件,这些app应用的规则需要进行下一步排查,以最终确定是否需要更新对应的规则,已完成对应用程序的识别。

[0064] 在本申请的一些实施例中,S103包括:当确认第一原始文件包括的所有规则均不与所述当前规则中的任意一条规则匹配时,则识别所述第一原始文件为需要更新规则的应用程序;将所述第一应用程序作为所述结果清单中的一个程序。本申请的一些实施例通过这个标准保证了识别需要更新规则的应用程序的判断标准的一致性。

[0065] 例如,将匹配结果对应的结果清单(例如,上述包含575条数据)与待测清单(例如,上述43款应用程序)进行对比,二者重叠部分,就是进行排查的部分。排查的目的在于查找需要更新规则的app的新规则,因为重叠的这些app是需要进行规则更新的app,说明原有的规则已经失效,需要去原始文件中寻找新的规则。将待测应用程序清单和程序执行结果清单进行比对(例如,可以采用人工比对),二者重复的数据,就是需要进行规则更新的应用(app)。实现原理解释:程序执行结果是地址库规则和原始数据对应的文件进行匹配的过程,当规则匹配不到原始数据时,说明规则失效,该app需要更新规则,那么该app进入结果清单,所以结果清单内数据都是需要进行规则更新的app,那么待测清单和结果清单相重叠的部分就是需要测试员进行规则更新的部分。整理规则进行入库。将需要规则更新的应用程序app进行规则排查,重新寻找可用规则,将规则填写在excel表格内将数据整理成设定的格式,即可进行入库。

[0066] S103涉及需要更新的规则包括:url规则和ua规则,其中,url规则是从原始文件中的url字段中提取,ua规则是从原始文件的ua字段中提取的,URL规则是指用户操作应用程序后,服务器和应用程序端(即app端)的交互信息,交互信息中的url字段中包含有效信息,将url字段内信息简化后制成的规则。UA规则是指用户操作应用程序后,交互信息中的user_agent字段中包含有效信息,将user_agent字段内信息简化后制成的规则。也就是说,更新的规则包括两种类型,分别是URL规则和UA规则,以上两种规则在原始文件中可以找到相应的字段。URL规则指用户上网日志中的url字段中提取出来的网页地址整理成的规则,UA规则指将包含在User_agent字段中的信息整理成的规则。

[0067] 本申请的一些实施例提供一种更新规则的装置,所述装置包括:测试模块,被配置为对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始文件,其中,所述原始文件用于记录对应应用程序的规则,所述规则包括上网规则;规则匹配模块,被配置为将至少一个原始文件中各原始文件包括的规则分别与预先获取的多个应用程序的原始规则匹配,得到匹配结果;识别模块,被配置为根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

[0068] 如图3所示,本申请一些实施例的规则更新方法包括:S301,生成地址库(可以采用手动方式生成);S202,上传原始数据对应的规则文件;S303,规则对比,即对比地址库中应用程序的当前规则和原始数据对应的原始规则;S304,判断新旧规则对比相似度是否满足

条件,如果确定为新旧规则相似则执行S205即不必更新规则,S206如果确定新旧规则不相似则执行S207的如何新规则;S308,程序执行结束。

[0069] 如图4所示,匹配过程包括:S401,生成原始数据地址库(例如,所述地址库是根据有效应用程序有效版本获取的数据库中的有效规则,即地址库用于存储当前规则);S402,对地址库中的数据进行分组;S403,数据比对(即将当前规则与原始数据进行比对,就是用生成的地址库与测试出来的那些原始文件内的数据进行比对),当规则相似时,则执行S404将应用程序添加至匹配成功的应用程序清单,当规则不相似时,则执行S405将应用程序添加至匹配失败的应用程序清单及S406生成unmatch.txt文件。

[0070] 请参考图5,图5示出了本申请实施例提供的更新规则的装置,应理解,该装置与上述图1-图4方法实施例对应,能够执行上述方法实施例涉及的所有步骤,该装置的具体功能可以参见上文中的描述,为避免重复,此处适当省略详细描述。装置包括至少一个能以软件或固件的形式存储于存储器中或固化在装置的操作系统中的软件功能模块,该更新规则的装置400,包括:测试模块501,被配置为对待测试清单包括的至少一个应用程序进行测试,获取与所述至少一个应用程序中各应用程序对应的原始数据,其中,所述原始数据用于记录对应应用程序的规则,所述规则包括上网规则;规则匹配模块502,被配置为将至少一个原始文件中各原始文件包括的规则分别与预先获取的多个应用程序的原始规则匹配,得到匹配结果;识别模块503,被配置为根据所述匹配结果识别需要更新规则的应用程序,并对所述需要更新规则的应用程序的规则进行更新。

[0071] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的装置的具体工作过程,可以参考前述方法中的对应过程,在此不再过多赘述。

[0072] 本申请的一些实施例提供一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时可实现上述第一方面所述的方法。

[0073] 如图6所示,本申请的一些实施例提供一种电子设备600,包括存储器610、处理器620以及存储在存储器610上并可在处理器620上运行的计算机程序,其中,处理器620通过总线630从存储器610读取程序并执行所述程序时可实现上述图1-图4中所述的方法。

[0074] 处理器620可以处理数字信号,可以包括各种计算结构。例如复杂指令集计算机结构、结构精简指令集计算机结构或者一种实行多种指令集组合的结构。在一些示例中,处理器620可以是微处理器。

[0075] 存储器610可以用于存储由处理器620执行的指令或指令执行过程中相关的数据。这些指令和/或数据可以包括代码,用于实现本申请实施例描述的一个或多个模块的一些功能或者全部功能。本公开实施例的处理器620可以用于执行存储器610中的指令以实现图1中所示的方法。存储器610包括动态随机存取存储器、静态随机存取存储器、闪存、光存储器或其它本领域技术人员所熟知的存储器。

[0076] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,附图中的流程图和框图显示了根据本申请的多个实施例的装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现方式中,方框中所标注的功能也可以以不同于

附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0077] 另外,在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0078] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0079] 以上所述仅为本申请的实施例而已,并不用于限制本申请的保护范围,对于本领域的技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。应注意到:相似的标号和字母在下面的附图中表示类似项,因此,一旦某一项在一个附图中被定义,则在随后的附图中不需要对其进行进一步定义和解释。

[0080] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应所述以权利要求的保护范围为准。

[0081] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

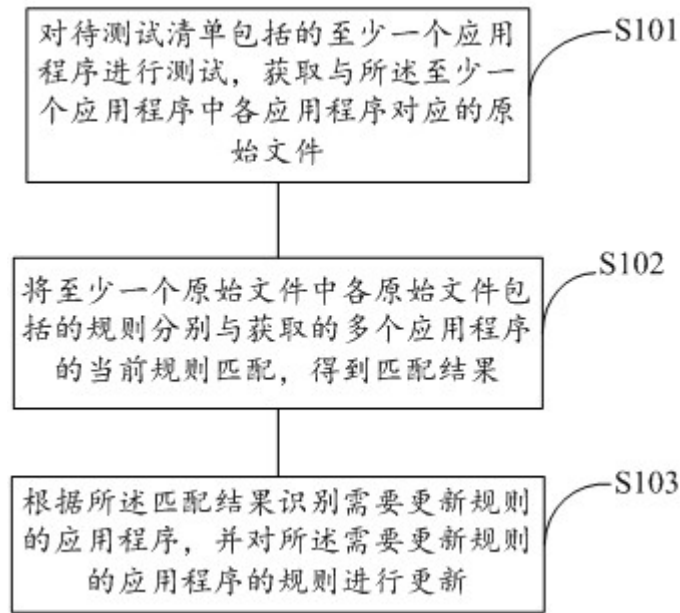


图1

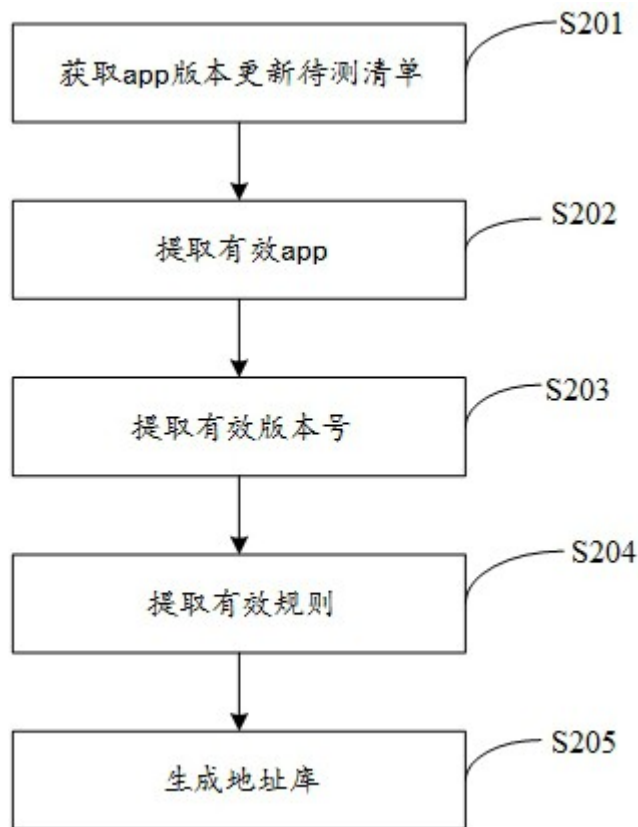


图2

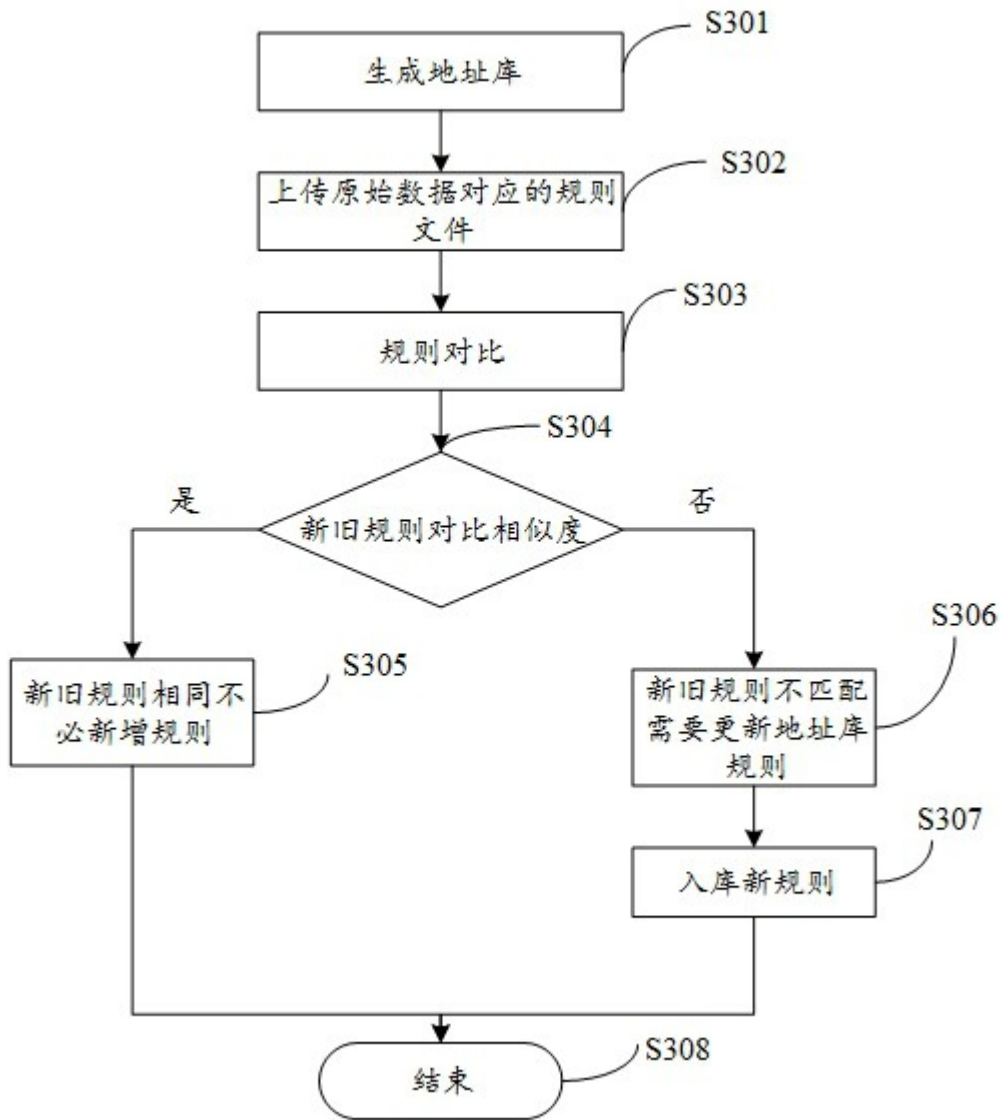


图3

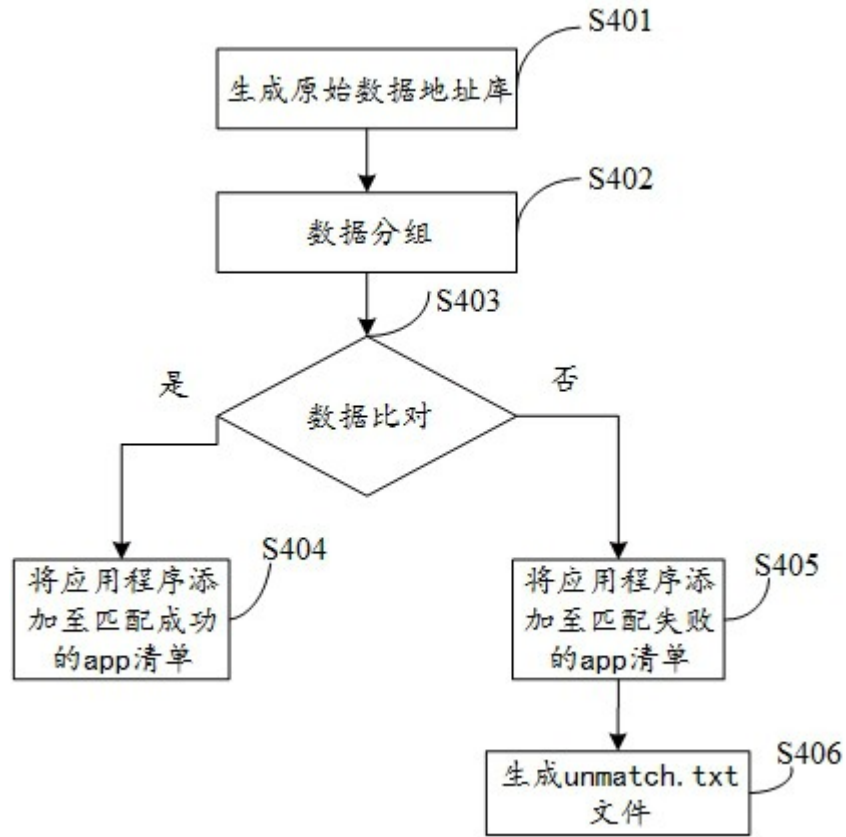


图4

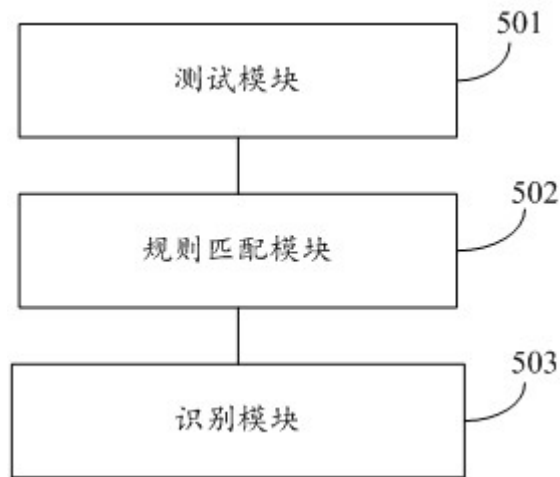


图5

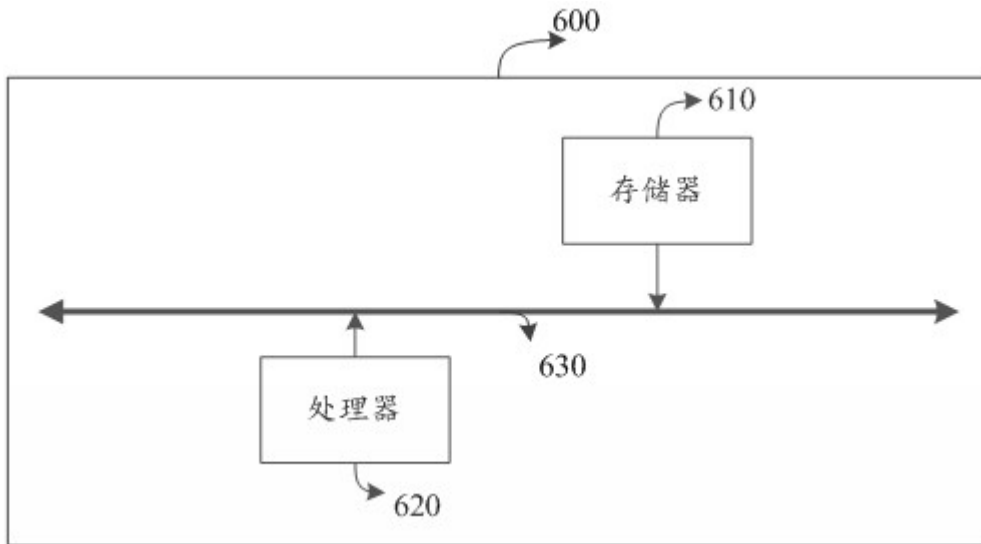


图6