**iOS静态检测API详细设计**

* com.bupt.ios.commonData

包功能简介：数据结构包，存放全局的变量

1. ProjectParameters

|  |  |
| --- | --- |
| 类名 | ProjectParameters |
| 功能 | 存放每次检测功能的参数 |
| ipaPath | 待检测ipa文件路径 |
| idaPath | IDA Pro路径 |
| detempPath | 反编译等逆向工作的根目录 |
| exeFilePath | ipa文件中可执行文件的路径 |
| idbPath | ida解析可执行文件生成的idb文件的路径 |
| idaPythonPath | 解析idb的python文件夹目录 |
| appName | ipa文件的名字 |
| resultPath | python解析idb生成的结果的路径 |

1. CommonData

|  |  |  |
| --- | --- | --- |
| 类名 | CommonData | |
| 功能 | 存放全局变量 | |
| rulePool | 简介 | 规则库，存放解析后的规则 |
| 类型 | List<SecRule> |
| addSecRule | 功能 | 添加一条规则 |
| 参数 | SecRule |
| getRulePool | 功能 | 获取规则库 |
| 返回 | List<SecRule> |

* com.bupt.decompile

包功能简介：负责对ipa进行逆向反编译

1. IdaAnalyse

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | IdaAnalyse | | |
| 功能 | 调用IDA Pro分析ipa生成idb文件并分析 | | |
| genIDB() | 权限 | public：对外接口 | |
| 功能 | 调用IDA分析IPA生成IDB，并把文件转移到detemp/日期/ida下带解析 | |
| analyseIDB() | 权限 | private：内部方法 | |
| 功能 | 在genIDB中被调用，用于解析IDB文件 | |
| copy(Stirng soure, String des) | 权限 | private：内部方法 | |
| 参数 | String source | 源文件路径 |
| String des | 目的地址目录 |
| 功能 | 复制文件到指定目录 | |

1. UnzipIpa

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | UnzipIpa | | |
| 功能 | 解压IPA文件 | | |
| unzip(String filePath,String desDir) | 权限 | static public | |
| 参数 | String filePath | 待解压文件路径 |
| String desDir | 解压目录 |
| 功能 | 解压IPA文件 | |

* com.bupt.ios.scheduler

包功能简介：调度程序，包括整个程序的入口，负责工程多个功能模块的调度功能。

1. MainSheduler

|  |  |  |  |
| --- | --- | --- | --- |
| 类名 | MainScheduler | | |
| 功能 | 程序的入口，负责调度其他Scheduler | | |
| main(String [] arg) | 权限 | public static | |
| 参数 | 参数1 | IPA路径 |
| 功能 | 入口函数 | |

1. PreScheduler

|  |  |  |
| --- | --- | --- |
| 类名 | PreScheduler | |
| 功能 | 预处理 | |
| scheduler() | 权限 | public：对外接口 |
| 返回值 | boolean：预处理成功返回true |
| 功能 | 1. 判断ipa文件是否合法 2. 读取配置文件中的ida路径，并在ProjectParameter中set 3. 清空detemp文件，并生成当前分析缓存路径 |
| clearDetemp()  clearDir() | 权限 | private：内部方法 |
| 功能 | 清空detemp文件夹 |

1. AnalyseSheduler

|  |  |  |
| --- | --- | --- |
| 类名 | AnalyseScheduler | |
| 功能 | 解析调度器 | |
| scheduler() | 权限 | public：对外接口 |
| 功能 | 1. 解压ipa 2. 定位可执行文件 3. 逆向ida生成idb并解析 |

* com.bupt.ios.rule

1. SecRule

|  |  |
| --- | --- |
| 类名 | SecRule |
| 功能 | 存放规则的数据结构 |
| ruleID(String) | 规则ID |
| ruleName(String) | 规则名称 |
| description(String) | 规则简介 |
| ruleType(int) | 规则类型：分为1和2。  1表示恶意API，匹配上规则表示存在漏洞  2表示保护API，没匹配上表示存在漏洞 |
| content(List<Function>) | 规则中需要匹配的API的列表 |
| riskLeve(int) | 危害等级 |
| solution(String) | 推荐的解决方案 |
| isEnable(boolean) | 是否启用规则。true启用 false禁用 |

1. Function

|  |  |
| --- | --- |
| 类名 | Function |
| 功能 | SecRule里面用到的API的数据结构 |
| FuncType (boolean) | API类型：分为true和false  最终SecRule匹配结果是SecRule里面每个API匹配结果相与得到，如果API类型为false，则API匹配结果取反，再同其他API匹配结果相与 |
| ClassName (String) | API需要匹配的类名，可以为“null” |
| FunctionName(String) | API名字，可以为“null” |
| Parameters (Map<String,String) | API的参数列表，类型是个Map，key是寄存器名称，value是参数的内容 |

* IDAPython

1. IDAPython.py

三个全局变量：

|  |  |
| --- | --- |
| 变量名 | regs |
| 类型 | 字典 Dictionary |
| 功能 | 针对每个方法，记录方法内部寄存器的状态，在解析每个方法前，会清空 |
| key | 寄存器名称，如X0，X1 |
| value | 寄存器内容 |

|  |  |
| --- | --- |
| 变量名 | msg\_send\_dict(弃用) |
| 类型 | 字典 Dictionary |
| 功能 | 每次调用msg\_send时，记录当前寄存器状态 |
| key | 当前方法名 |
| value | 类型是个List，List中的元素是regs |

|  |  |
| --- | --- |
| 变量名 | BL\_dic（弃用） |
| 类型 | 字典 Dictionary |
| 功能 | 针对除了msg\_send之前的API调用 |
| key | 被调用的API |
| value | 类型是个List，元素是API的调用者 |

|  |  |
| --- | --- |
| 变量名 | BL\_dic（改版） |
| 类型 | 字典 Dictionary |
| 功能 | 针对每个B或者BL调用时，寄存器信息 |
| key | Function Name （每个类中每个方法） |
| value | 类型是个List：  第一个元素是BL的方法名，如msg\_send；  第二个元素是BL指令的内存地址；  第三个元素是 一个dict：存入寄存器信息 |

关键方法

|  |  |
| --- | --- |
| 方法 | parseMOV |
| 传人参数 | addr：指令的地址 |
| 功能 | 解析当前的MOV指令 |

|  |  |
| --- | --- |
| 方法 | parseLDR |
| 传人参数 | addr：指令的地址 |
| 功能 | 解析当前的LDR指令 |

|  |  |
| --- | --- |
| 方法 | parseB |
| 传人参数 | addr：指令的地址 |
| 功能 | 解析当前的B系列的指令 |

|  |  |
| --- | --- |
| 方法 | parseSTR |
| 传人参数 | addr：指令的地址 |
| 功能 | 解析当前的STR系列的指令 |

|  |  |
| --- | --- |
| 方法 | parseOther |
| 传人参数 | addr：指令的地址 |
| 功能 | 解析当前其他指令 |