* 1. 上传恶意程序
     1. 文件选择选项，可以选择本地文件
     2. 有上传按钮，点击上传按钮之后文件将上传，并准备分析
     3. 分析按钮，可以通过IDA可视化程序中的函数调用关系，用颜色标注程序中的缺陷函数
  2. 恶意程序代码提取
     1. 通过IDA 和 IDApython 将恶意程序以函数为粒度进行分割，将每个函数的基本块以执行顺序建树，并提取每个函数中最长路径。
     2. 将函数的最长路径中的基本块提取出来，基本块中的每个指令序列都记录为 操作指令\_操作数1\_操作数2 的形式
     3. 通过操作数的类型对操作数进行归一化处理，将操作数类型分为8类

|  |  |
| --- | --- |
| Void | 没有操作数 |
| Reg | 寄存器 |
| Mem | 内存 |
| Phrasse | 索引 |
| Displ | 寄存器或者操作数+位移 |
| Imm | 立即数 |
| Far | 立即远地址 |
| near | 立即近地址 |

对于寄存器和寄存器偏移，保留其原本的数据不进行归一化处理

* 1. 将每个函数的指令以空格为分割，记录在csv文件中，每个函数的指令占用一行。
  2. 预测模型
     1. 将数据输入预测模型，能够产生预测结果，0表示有缺陷，1表示没有缺陷
  3. 前端展示
     1. 能够展示预测的函数名以及函数的预测结果，通过不同的颜色进行标注
     2. 产生一个饼图，饼图中分析有缺陷和无缺陷的比例，从而对程序的安全性进行预测
  4. 上传恶意程序
     1. 文件选择选项，可以选择本地文件
     2. 有上传按钮，点击上传按钮之后文件将上传，并准备分析
  5. 恶意程序代码提取
     1. 通过IDA 和 IDApython 将恶意程序以函数为粒度进行分割，将每个函数的基本块以执行顺序建树，并提取每个函数中最长路径。
     2. 将函数的最长路径中的基本块提取出来，基本块中的每个指令序列都记录为 操作指令\_操作数1\_操作数2 的形式
     3. 通过操作数的类型对操作数进行归一化处理，将操作数类型分为8类

|  |  |
| --- | --- |
| Void | 没有操作数 |
| Reg | 寄存器 |
| Mem | 内存 |
| Phrasse | 索引 |
| Displ | 寄存器或者操作数+位移 |
| Imm | 立即数 |
| Far | 立即远地址 |
| near | 立即近地址 |

对于寄存器和寄存器偏移，保留其原本的数据不进行归一化处理

* 1. 将每个函数的指令以空格为分割，记录在csv文件中，每个函数的指令占用一行。
  2. 预测模型
     1. 将数据输入预测模型，能够产生预测结果，0表示有缺陷，1表示没有缺陷
  3. 前端展示
     1. 能够展示预测的函数名以及函数的预测结果，通过不同的颜色进行标注
     2. 产生一个饼图，饼图中分析有缺陷和无缺陷的比例，从而对程序的安全性进行预测