# VRLT说明文档

# 命名规范

* 文件夹：英文全拼小写，多个英文单词用“-”隔开
* 文件：英文全拼小写，多个英文单词用“\_”隔开
* 函数名：英文全拼小写，多个英文单词用“\_”隔开，首个英文单词尽量使用动词(eg. open\_shell)
* 变量名：英文全拼小写，多个英文单词用“\_”隔开，对于大家熟知的单词可以使用单词缩写(eg. buf, tmp , fp)，对于没有实际意义的变量可以使用单个字符(eg. i, j, k)，全局变量以g\_开头

# 框架简介

* VRLT.py主调程序，负责UI
* tools完成一定功能的Python程序，供VRLT.py调用
* Src
  + report 程序运行产生的报告信息
  + 主要的漏洞和攻击程序
    - documents存放说明文档
    - vulnerabilities存放漏洞程序
    - check 存放检测程序，检查攻击是否成功，将结果返回给主调程序
    - input存放漏洞程序所需的输入文件（shellcode）
    - output存放待检查文件，供check中的检测程序使用
    - Config 配置信息

# 要求

* 漏洞程序和攻击一一对应
* 漏洞程序必须从input中读取相应的shellcode
* 对于每一个漏洞程序都要在documents中写相应的文档说明，尽量详细
* 用C语言写漏洞程序，并编译成可执行文件，vulnerabilities只存放.c和可执行文件
* Input中存放两个文件，一个存放shellcode，另一个是正常文件，供程序正常运行使用，正常文件的命名是在shellcode文件命名的基础上加上‘\_normal’
* 用C语言写检测程序，并编译成动态链接库，在程序的开头注释编译选项，程序中的检测函数定义为char\* check(){}，返回值类型为char\*，函数名为check，不需要参数，check文件夹中只存放.c和.so文件
* 针对vulnerabilities中的每一个漏洞程序的攻击，在check中必须存在相应的检测程序，检测攻击是否成功，并将结果反馈给主调程序，如果检测程序需要攻击程序的输出作为判断，则将输出信息存放到output中，供check使用
* 对于检测程序check的反馈信息有如下规定：
  + 检测程序的反馈信息将作为产生报告的重要依据，所以反馈信息要尽量详细
  + 如果检测到攻击失败，则返回值的起始部分必须为“attack fail,”,再在它的后面加上攻击失败的原因
* 程序的配置信息写在Config文件中
  + 格式为：属性名=属性值
  + 现有的属性：
    - type 定义程序的类型
    - compile 编译程序的命令
    - aslr 定义aslr的状态, 其值为on或off
* 定义了一系列的环境变量
  + BASE\_PATH =src文件的绝对路径
  + 以下NAME代表具体的文件名
  + VUL\_NAME\_PATH=vulnerabilities文件夹下NAME文件的路径
  + CHECK\_NAME\_PATH=check文件夹下NAME文件的路径
  + OUTPUT\_NAME\_PATH=output文件夹下NAME文件的路径
  + INPUT\_NAME\_PATH=input文件夹下NAME文件的路径
  + DOCUMENT\_NAME\_PATH=document文件夹下NAME文件的路径
  + REPORT\_PATH=report文件的路径
  + CONFIG\_NAME\_PATH=Config文件的路径
  + NORMAL=TRUE/FALSE为TRUE正常运行，为FALSE运行攻击程序
  + ATTACH=TRUE/FALSE为TRUE调试程序，为FALSE运行程序

# 使用

* VRLT.py为主调程序，提供了如下命令：

1. load 加载程序
2. show p 显示可以利用的所有程序
3. show type 按类型进行划分显示
4. cat report name 查看产生的报告，默认查看report.txt，

可指定报告的名字name

1. save report name 保存产生的报告，默认保存为report.txt，

可指定报告的名字

1. del report name 删除产生的报告，默认删除report.txt，

可指定报告的名字

1. select i 选择第i个要执行的程序
2. select i1 i2 ... 选择第i1,i2,i3...个要执行的程序
3. select all 选择所有的程序执行
4. select type name 选择name类型的程序
5. run 运行选择的漏洞程序,并进行攻击
6. run normal 运行选择的漏洞程序,正常运行
7. attach i 附加调试第i个程序
8. aslr status 获取ASLR的状态
9. aslr on/off/conservative 修改ASLR状态
10. help run 查看相关命令信息
11. q 退出

# 扩展

* 首先在src中建立一个文件夹，命名为漏洞名，攻击名，shellcode名的组合，例如buffer\_overflow\_code\_injection\_write\_file其含义为漏洞是缓冲区溢出，所采取的攻击为代码注入，攻击的结果为对一个文件进行写操作
* 然后在文件夹中建立vulnerabilities,input, output, check, documents文件夹和Config文件，
* 按要求在vulnerabilities,input, output, check, documents文件中建立相对应的文件，注意命名要与一开始建立的文件夹一致
* input中存放两个文件，一个文件内容为攻击所需的shellcode，供漏洞程序读取以及一个正常的输入文件，注意命名
* 如果攻击程序有输出，均保存到output中
* 按照规范在check中自定义检测程序
* 按要求编写Config文件
* 编写详细的文档说明
* VRLT.py主调程序一般情况下不需要修改，按照使用说明使用即可，使用过程中如果遇到问题可协商进行修改