# 项目开发文档

## 1 项目位置

项目位置位于github网站私有库pentest下sandbox目录中, 其目录结构如下:

sandbox

---- PHP\_Sandbox 验证沙箱可行性的简单项目

-------- Dockerfile 用以构建执行风险php文件所需要的沙箱文件

-------- libwebshell.so 将危险函数重写后, 重新编译后的php动态链接库

-------- php.ini 采用重新编译后php动态链接库的php配置文件

-------- sandbox.py 使用flask-restful编写的restful接口文件, 用来提供http请求功能

-------- test.php 一个简单的webshell, 用以测试沙箱运行结果

---- Sandbox\_Service PHP沙箱正式项目目录

-------- DB\_Init.py 数据库初始化脚本

-------- webshell.db 用来记录PHP文件执行信息的数据库文件

-------- Dockerfile 创建执行风险php文件所需要的沙箱文件

-------- sandbox.py 使用flask-restful编写的restful接口文件, 用来提供http请求功能

-------- libwebshell.so 与PHP\_Sandbox目录下同名文件相同

-------- php.ini 与PHP\_Sandbox目录下同名文件相同

-------- outerDB\_Init.py  宿主机调度程序使用的数据库初始化脚本

-------- outerSandbox.py 宿主机调度程序

-------- venv python virutalenv目录, 用来创建项目开发环境

---- dev 搭建开发环境使用的文档

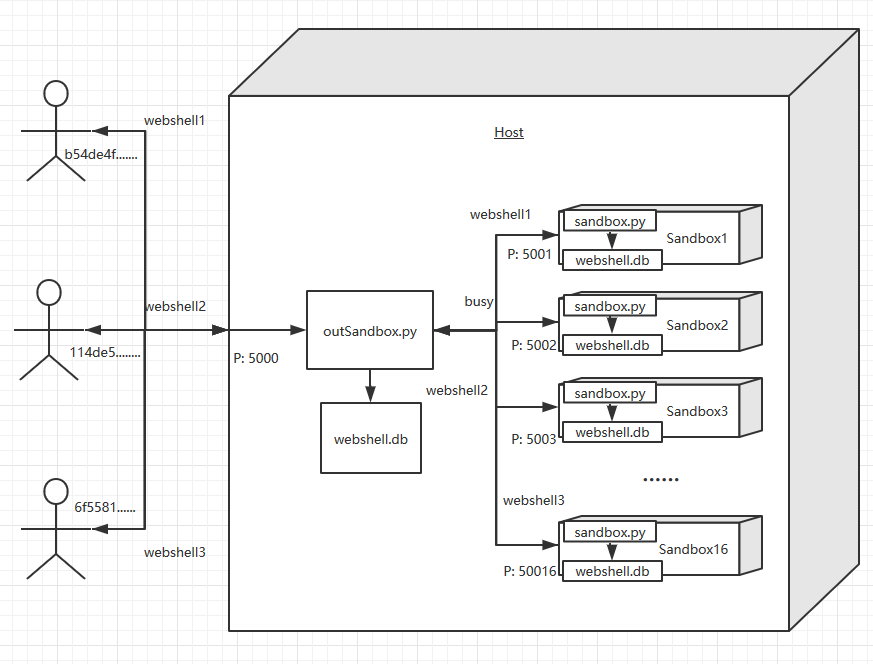
-------- dev.sh 用来记录启动沙箱时需要的shell命令

-------- Dockerfile 一个简单的能够运行titanagent的沙箱环境

## 2 项目说明

本次项目主要使用python语言进行开发, 使用了flask-restful框架. 同时也采用docker技术作为沙箱容器.

### 2.2 项目结构



### 2.1 项目执行流程如下

1. 使用流程.
   1. 通过`curl 127.0.0.1:5000/webshell -F [file=@test.php](mailto:file=@test.php) action=upload`形式进行webshell文件上传. 会得到文件hash值作为响应.
   2. 上传的文件会在沙箱中进行执行, 并将其执行信息记录在数据库中, 其中包括调用的危险函数名, 传入的参数, 运行所需的时间等.
   3. 可通过`curl 127.0.0.1:5000/webshell/[hash]`形式取得该文件的执行情况, 从而进一步帮助判断此文件是否为webshell.
2. 程序执行流程
   1. 使用Sandbox\_Server下Dockerfile文件, 通过执行命令`docker build [sandbox] .`来创建PHP沙箱镜像. 之后可通过多次执行`docker run -p 5000:5001 -it -d [sandbox] bash`来创建多个PHP沙箱容器.
   2. 在PHP沙箱容器运行成功后, 通过执行命令`python sandbox.py`启动PHP执行接口脚本, 以开启容器内沙箱接受http请求功能. 通过执行命令`python DB\_Init.py`初始化数据库文件webshell.db.
   3. 通过在宿主机上执行命令`python outerSandbox.py`来启动宿主机调度脚本, 以启动宿主机任务调度功能. 通过执行命令`python outerDB\_Init.py`初始化调度数据库文件webshell.db ( 开发未完成 ) .
   4. 当宿主机接收到外部执行命令`curl 127.0.0.1:5000/webshell -F [file=@test.php](mailto:file=@test.php) action=upload`上传的文件test.php时, 调度脚本查询宿主机内开启的沙箱容器的运行状态并找到一个空闲状态的容器, 将文件内容传入该容器, 并将调度信息记录入webshell.db数据库文件中. 否则加入队列, 等待空闲容器.
   5. 空闲容器接受到php文件后, sandbox.py脚本会计算该文件hash值, 返回hash值并开启一个线程执行该php文件. 执行过程中发现的执行信息与hash值会记录到webshell.db数据库文件中.
   6. 当宿主机接收到外部执行命令`curl 127.0.0.1:5000/webshell/[hash]`请求的hash值后, 调度脚本查询调度信息数据库文件webshell.db, 并找到该文件传入的沙箱容器, 将该hash值请求转发至该容器.
   7. 容器接受到hash值请求后, 查询执行信息数据库webshell.db, 并将该执行信息返回.