

测试用例的 Mutation

一、前提说明

1、测试用例的 Mutation 是指将基础的测试用例,通过 mutation 的方法,生成新的测试用例,这一过程主要分成以下几个部分:读取基础测试用例文件、对每条基础测试用例进行 mutation、将 mutation 之后生成的新的测试用例写入文件。

2、测试用例的 Mutation 相关资料都在目录 “injection.backup” 文件夹中。里面包括七个主要目录: extra、lib、mutation、plugins、sql-injection、tamper、thirdparty, 和两个文件: mutation_method、tamper.py。其中, extra、lib、plugins、tamper、thirdparty 文件夹是程序运行的必要目录,包含基础的程序运行资源文件,tamper 文件夹中包含 41 种 mutation 方法,每种方法单独以一个文件来实现, mutation 的方法、mutation 实例以及对应的文件在 mutation_method 文件中进行描述; sql-injection 文件夹是基础测试用例存放的文件夹,目录可以在程序中自行设定; mutation 文件夹是经过 mutation 之后生成的新的测试用例的文件夹,每个文件对应基础测试用例相应的文件名; tamper.py 脚本是主程序,运行该脚本程序实现基础测试用例的 mutation。

3、程序的运行环境 python2.7 环境,安装之后运行 tamper.py 脚本即可。

二、测试用例 mutation 的具体步骤

下面介绍使用基础的测试用例进行 mutation 生成新的测试用例的具体步骤,以下所有操作都是基于 tamper.py 脚本进行操作:

1、设置基础测试用例路径及文件名

这个步骤,是将需要进行 mutation 的基础测试用例放入 sql-injectin 文件夹,将 tamper.py 脚本中 INPUT_FILE_NAME 设置成需要 mutation 的基础测试用例名,如对 sql-injection 文件夹下的“fuzzdb”文件进行 mutation,那么需要将“INPUT_FILE_NAME = “fuzzdb””。

如果需要更改基础测试用例的路径,需要修改 tamper.py 文件中的 INPUT_PATH 变量,将其设置成相对应文件夹相对于 tamper.py 文件的相对路径,如需要将“sql-injectin”文件夹设置为“testcases”文件夹,只需要修设置“INPUT_PATH = “./testcases/””。

该部分要求基础测试用例每行一条测试用例,文件格式如下:

```
'  
'--  
' or 1=1--  
1 or 1=1--
```

2、设置 mutation 生成文件名后缀

这个步骤是设置生成测试用例的后缀名，程序默认输出文件名称格式为：“XXXX_mutation”，其中“xxxx”表示基础测试用例文件名称。

如果需要更改生成测试用例文件的后缀，可以通过修改 `tamper.py` 程序中 `OUTPUT_FILE_NAME` 变量，将下面红色标记部分改成相对应的名字：

```
OUTPUT_FILE_NAME = ".join([OUTPUT_PATH,"/",INPUT_FILE_NAME,"mutation"])
```

如果需要更改生成测试用例文件的输出路径，可以通过修改 `tamper.py` 程序中 `OUTPUT_PATH` 变量，将下面红色标记部分改成相对应的名字或者相对于 `tamper.py` 文件的相对路径：

```
OUTPUT_PATH = "mutation"
```

3、设置去重之后的输出文件名后缀

因为第二步 `mutation` 之后，在新的测试用例集中会存在重复的测试用例，这一过程是对输出文件去重，并重新写入文件，文件输出路径为“`mutation`”，输出文件名默认格式为：“xxxx_purge”，其中“xxxx”表示基础测试用例文件名称。

如果需要更改生成测试用例文件的后缀，可以通过修改 `tamper.py` 程序中 `PURGEOUTPUT_FILE` 变量，将下面红色标记部分改成相对应的名字：

```
PURGEOUTPUT_FILE = ".join([OUTPUT_FILE_NAME,"_purge",".txt"])
```

4、运行 `tamper.py` 脚本，输出 `mutation` 之后的文件