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mutaion、将mutation之后生成的新的测试用例写入文件。

2、测试用例的Mutaion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：mutaion\_menthod、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，mutaion的方法、mutaion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mutaion。

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mutaion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--

1. 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"

1. 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