篇一：模糊测试与Spike框架简介

> https://leex0.top/2020/08/02/%E6%A8%A1%E7%B3%8A%E6%B5%8B%E8%AF%95%E4%B8%8ESpike%E6%A1%86%E6%9E%B6%E7%AE%80%E4%BB%8B/

1. 内容介绍

1.1 模糊测试简介

模糊测试（Fuzzing），是一种通过向目标系统提供非预期的输入并监视异常结果来发现软件漏洞的方法。
其核心思想是自动或半自动的生成随机数据输入到一个程序中，并监控目标程序异常，如崩溃，断言(assertion)失败，以发现可能的程序错误，比如内存泄漏等。
即用随机坏数据（也称做 fuzz）攻击一个程序，然后等着观察哪里遭到了破坏。但这一过程却能揭示出程序中的重要 bug。
它是一种介于完全的手工渗透测试与完全的自动化测试之间的安全性测试类型。它充分利用了机器能够随机生成和发送数据的能力。

1.2 Spike框架

Spike是一个模糊器创建工具包，它提供了API，允许用户基于网络的协议来创建自己的fuzzer。其中包含一些通用的模糊测试器。

1.3 网络协议模糊测试

对网络协议进行模糊测试也需要识别出可被攻击的接口，通过变异或生成方式得到能够触发错误的模糊测试值，然后将这些模糊测试值发送给目标应用，监视目标应用的错误。

2. 基本过程

- 确定测试的目标
- 确定输入的向量
- 生成模糊测试数据，可由测试工具通过随机或是半随机的方式生成
- 执行模糊数据测试
- 监视异常
- 判定发现的漏洞是否可能被利用

3. 测试方法分类

- 基于`变异`的模糊测试——简而言之就是正常调用协议，抓包，然后混淆数据包达到生成异常数据包的结果，从而进行测试。

> 这种方法对已有的正常数据集依赖较高。需要有足够丰富的合法输入从而产生足够丰富的测试类型。

例如，png图片除了文件头后面数据内容进行置换混淆得到异常测试数据。

```
00000000:  8950 4e47 0d0a 1a0a 0000 000d 4948 4452 0000 02ca 0000 01da  :.PNG........IHDR........
00000018:  0806 0000 0019 a86f 4600 000c 6569 4343 5049 4343 2050 726f  :.......oF...eiCCPICC Pro
00000030:  6669 6c65 0000 4889 9597 0758 53c9 1680 e796 5412 5a20 0252  :file..H....XS.....T.Z .R
00000048:  426f a248 0d20 2584 1641 40aa 202a 2109 2494 1813 828a 1d59  :Bo.H. %..A@. *!.$......Y
00000060:  56c1 b58b 2856 ac88 8bae ae80 ac05 11d7 ba28 76d7 b258 5059  :V...(V...........(v..XPY
00000078:  5917 57b1 a1f2 2605 74dd 57be 37df 3777 fe9c 3973 e69c 9399  :Y.W...&.t.W.7.7w..9s....
00000090:  7b67 00d0 ebe4 cb64 f9a8 3e00 05d2 4279 4264 286b 425a 3a8b  :{g.....d..>...ByBd(kBZ:.
000000a8:  f408 1001 0a74 4000 d0e3 0b14 324e 7c7c 0c80 65b0 fd7b 797d  :.....t@.....2N||..e..{y}
000000c0:  1d20 aaf6 8a9b cad6 3ffb ff6b 3114 8a14 0200 900c c859 4285  :. ......?..k1........YB.
```

- 基于`生成`的模糊测试——简而言之就是理解协议规约定义，创建文法自动生成动态模糊的测试用例。

> 这种方法对协议的理解掌握程度需求更高。难度更大。

http的post请求如图，其中fuzzable的点可用来生成测试例子。

```
POST /testme.php HTTP/1.1
User-Agent: Mozilla/4.0
Host: testserver.example.com
Content-Length: 256
Connection: close
inputvar=admin
```

```
[fuzzable] [fuzzable] HTTP/1.1
User-Agent: [fuzzable]
Host: [fuzzable]
Content-Length: [fuzzable]
Connection: [fuzzable]
inputvar=[fuzzable]
```

4. 困难点

协议分析，对于特定的程序与配套协议，需要对程序段进行逆向分析才能得到协议内容以创建测试样例，而其中逆向分析工作难度与工作量都较大。

篇二：Spike简单测试使用

> https://leex0.top/2020/08/02/%E6%A8%A1%E7%B3%8A%E6%B5%8B%E8%AF%95%E4%B8%8ESpike%E6%A1%86%E6%9E%B6%E7%AE%80%E4%BB%8B/
>
> 之前了解了一下Spike模糊测试框架，本文记录一下使用过程中遇到的问题与使用测试结果。

1. 编译问题

macos编译

使用macos编译出现一些问题，未再考虑。

```
spike.c:2551:17: warning: passing 'unsigned char [5000]' to parameter of type
      'const char *' converts between pointers to integer types with different sign
      [-Wpointer-sign]
  return strlen(buf);
                ^~~
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/string.h:82:28: note:
      passing argument to parameter '__s' here
size_t   strlen(const char *__s);
                            ^
115 warnings and 1 error generated.
make: *** [spike.o] Error 1
```

centos编译

centos编译结果正常，但是运行过程中显示一个库文件`libdlrpc.so`引用失败。如图：

```
[buddyholly@localhost src]$ ./generic_send_tcp 10.37.129.3 9999 vultest.spk 0 0
./generic_send_tcp: error while loading shared libraries: libdlrpc.so: cannot op
en shared object file: No such file or directory
[buddyholly@localhost src]$
```

根据查找，运行`src`下`ld.sh`脚本导入环境变量解决。

注：通过`source ld.sh`或`. ./ld.sh`运行，通过`./ld.sh`则无法解决。原因参见此文章。

`ld.sh`如下：

```
#Use this to use any of the generic programs
#try using . ./ld.sh if it's not working
export LD_LIBRARY_PATH=.
```

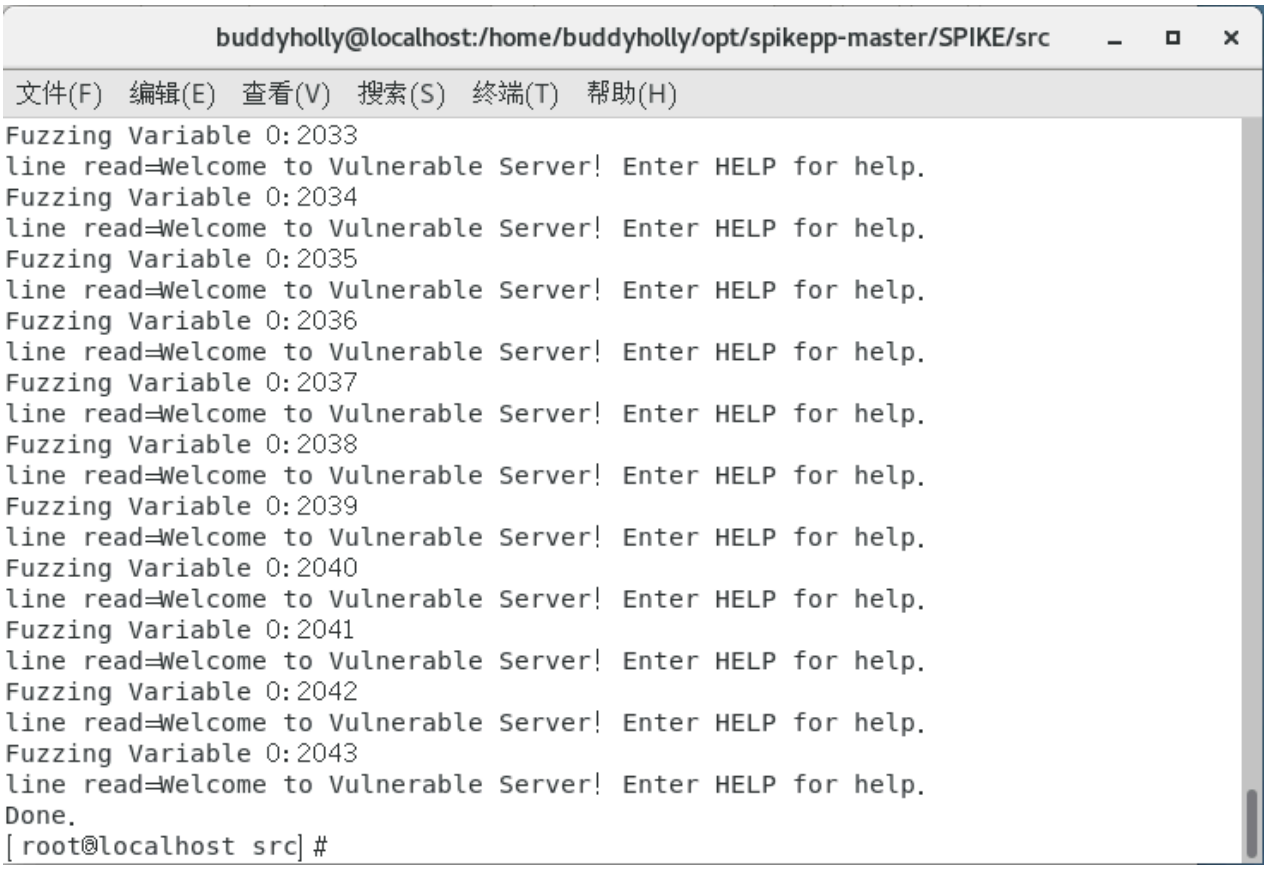
2. vulnserver程序测试

通过使用vulnserver，测试spike能否正确使用。

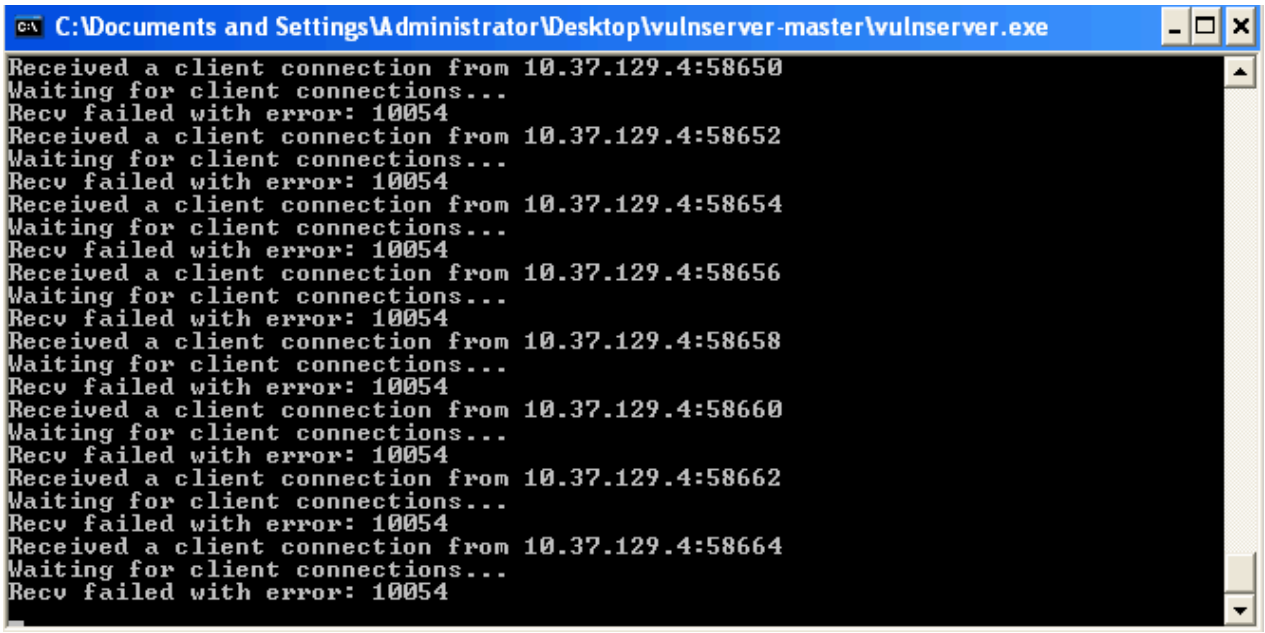
脚本如下：

```
s_readline();
s_string("TRUN");
s_string_variable("COMMAND");
```

攻击端（CentOS）`./generic_send_tcp 10.xx.xx.xx 9999 vul_test.spk 0 0`：



服务器端（WindowsXP）程序崩溃：



3. flask web测试

搭建一个简单的web服务器(`python flask`)测试spike使用。

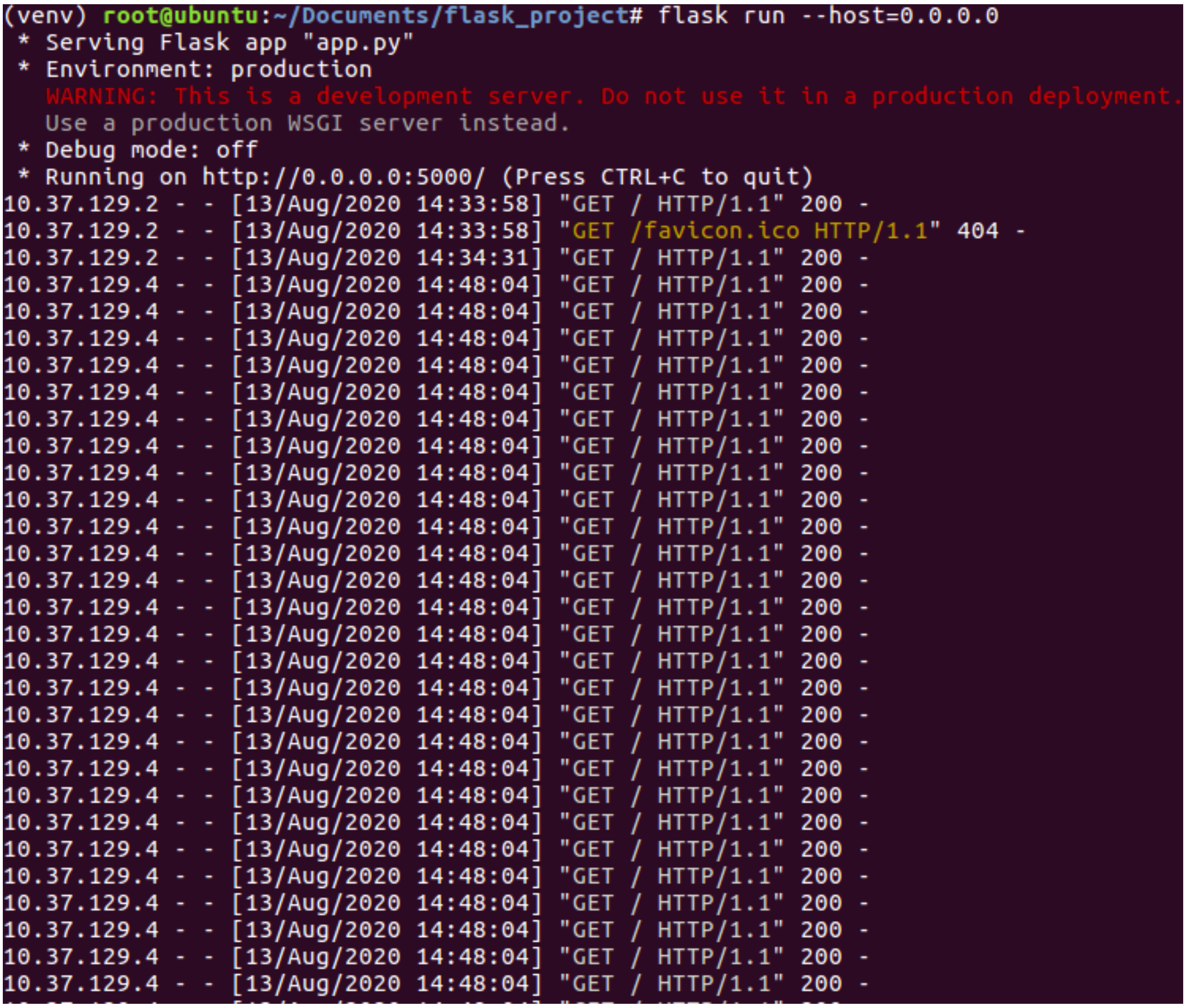
攻击脚本如下：

```
s_string("GET / HTTP/1.1\r\n");
s_string("Host: 10.37.129.5:5000\r\n");
s_string("User-Agent: ");
s_string("Content-Length: ");
s_blocksize_string("block1", 5);
s_string("\r\nConnection: close\r\n\r\n");
s_block_start("block1");
s_string("inputvar=");
s_block_end("block1");
```

攻击端（CentOS）`./generic_send_tcp 10.37.129.5 5000 flask.spk 0 0`：

```
root@localhost src# ./generic_send_tcp 10.37.129.5 5000 ./testscripts/flask.sp
k 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Fuzzing Variable 0:2
Fuzzing Variable 0:3
Fuzzing Variable 0:4
Fuzzing Variable 0:5
Fuzzing Variable 0:6
Fuzzing Variable 0:7
Fuzzing Variable 0:8
Fuzzing Variable 0:9
Fuzzing Variable 0:10
Fuzzing Variable 0:11
Fuzzing Variable 0:12
Fuzzing Variable 0:13
Fuzzing Variable 0:14
```

服务器端（Ubuntu）：



```
(venv) root@ubuntu:~/Documents/flask_project# flask run --host=0.0.0.0
 * Serving Flask app "app.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
10.37.129.2 - - [13/Aug/2020 14:33:58] "GET / HTTP/1.1" 200 -
10.37.129.2 - - [13/Aug/2020 14:33:58] "GET /favicon.ico HTTP/1.1" 404 -
10.37.129.2 - - [13/Aug/2020 14:34:31] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
10.37.129.4 - - [13/Aug/2020 14:48:04] "GET / HTTP/1.1" 200 -
```