# Exploring Effective Fuzzing Strategies to Analyze Communication Protocols

Yurong Chen, Tian Lan and Guru Venkataramani

ACM FEAST workshop colocated with CCS 2019

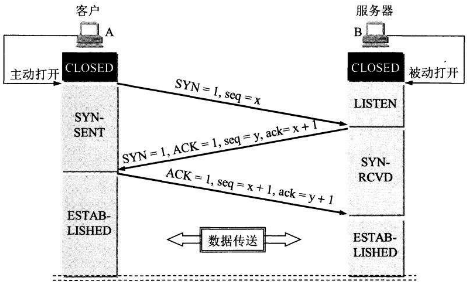
October 16, 2019

# Contents

1. Background and Goal
2. Design and Implementation
3. Evaluation
4. Thought

# What's communication protocol fuzzing?

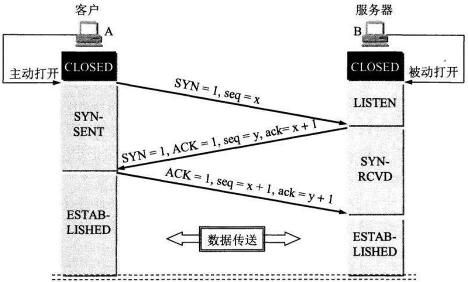
- Stateful
- Dependent packages
- Multiple formats

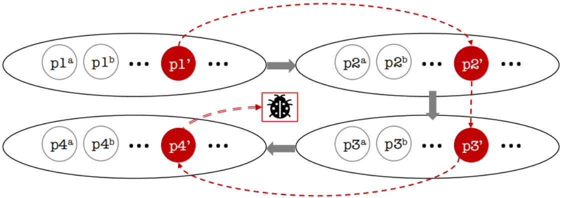


# Limitation

- Blind Fuzzing
- Fuzzing the first packet
- Rely on well-constructed test program

# Limitation


客户
服务器
A
B
CLOSED
主动打开
CLOSED
被动打开
SYN = 1, seq = x
LISTEN
SYN-SENT
SYN = 1, ACK = 1, seq = y, ack= x + 1
SYN-RCVD
ACK = 1, seq = x + 1, ack = y + 1
ESTAB-LISHED
数据传送
ESTAB-LISHED


p1ᵃ
p1ᵇ
• • •
p1'
• • •
p2ᵃ
p2ᵇ
• • •
p2'
• • •
p4ᵃ
p4ᵇ
• • •
p4'
• • •
p3ᵃ
p3ᵇ
• • •
p3'
• • •

# Fork to keep status

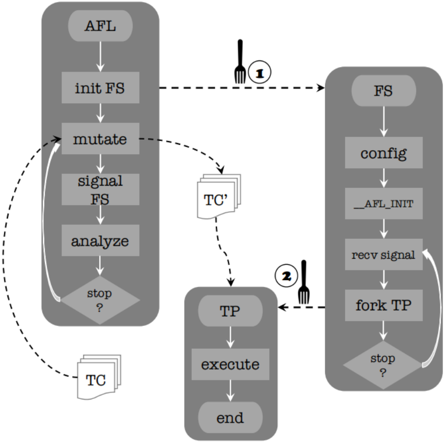
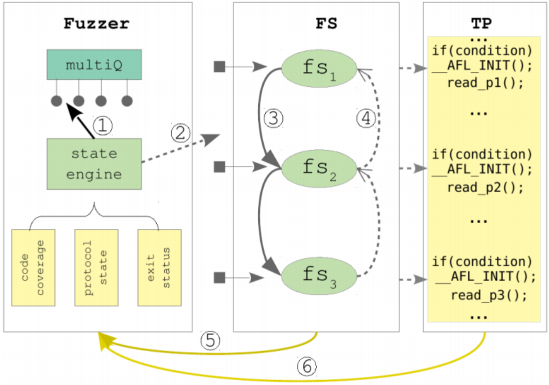


**Figure 2: Simplified AFL forking workflow.** FS: forkserver, TC/TC': testcase, TP: testing program

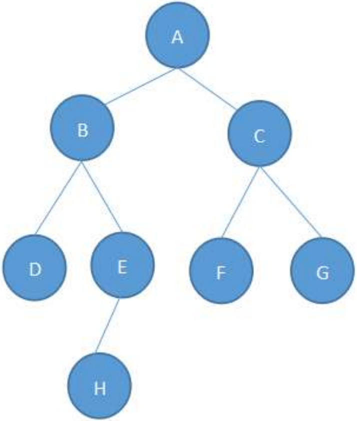
# Fork to keep status



```
...
if(condition)
__AFL_INIT();
 read_p1();

 ...

if(condition)
__AFL_INIT();
 read_p2();

 ...

if(condition)
__AFL_INIT();
 read_p3();
 ...
```

**Figure 3: System Overview of our Stateful Fuzzer Design.** FS: forkserver, multiQ: queues for storing different types of testcases, TP: testing program

# Fork to keep status



# How to choose the status to fuzz?

- During the profiling stage, each packet is fuzzed for a fixed amount of time(one hour)
- Provide an overview of code coverage and fuzzing queue related to each packet
- Higher code coverage and more queue entries will be assign more fuzzing time
- Higher code coverage and more queue entries will have a larger probability to be progressed

# OpenSSL v101

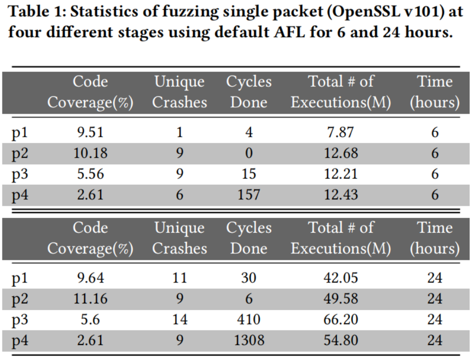


**Table 1: Statistics of fuzzing single packet (OpenSSL v101) at four different stages using default AFL for 6 and 24 hours.**

| | Code Coverage(%) | Unique Crashes | Cycles Done | Total # of Executions(M) | Time (hours) |
|---|---|---|---|---|---|
| p1 | 9.51 | 1 | 4 | 7.87 | 6 |
| p2 | 10.18 | 9 | 0 | 12.68 | 6 |
| p3 | 5.56 | 9 | 15 | 12.21 | 6 |
| p4 | 2.61 | 6 | 157 | 12.43 | 6 |
| | Code Coverage(%) | Unique Crashes | Cycles Done | Total # of Executions(M) | Time (hours) |
| p1 | 9.64 | 11 | 30 | 42.05 | 24 |
| p2 | 11.16 | 9 | 6 | 49.58 | 24 |
| p3 | 5.6 | 14 | 410 | 66.20 | 24 |
| p4 | 2.61 | 9 | 1308 | 54.80 | 24 |

Improved coverage: 19.27%(24hour)

# Thought

Packet type, Packet field value, Packet queue → Crash or not?

Machine learning

Filter test cases

Speed up!