ISP-Fuzzer: extendable fuzzing framework

Jivan Hakobyan

jivan@ispras.ru

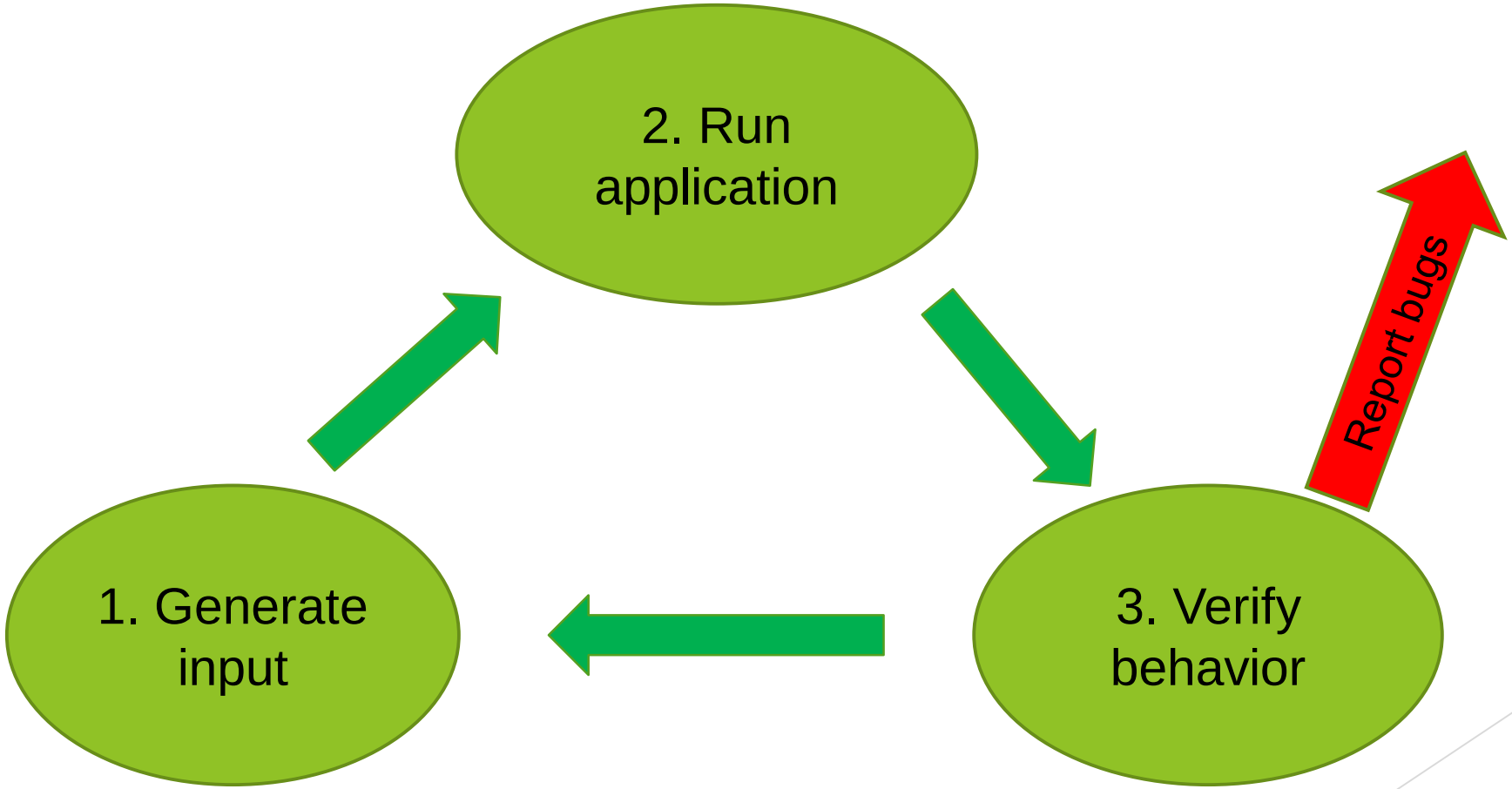
System programming laboratory,

RAU, ISP RAS

Introduction

- In the modern world the development of reliable software is still an essential aspect in the field of information technologies (IT).
- Fuzzing is one of the most popular and efficient methods of dynamic analysis.

Process of Fuzzing



Related Work

AFL

Related Work

AFL

kAFL

WinAFL

Related Work

AFL

Syzkaller

kAFL

WinAFL

Related Work

AFL

LibFuzzer

Syzkaller

kAFL

WinAFL

Related Work

AFL

LibFuzzer

Syzkaller

VUzzer

kAFL

WinAFL

Related Work

AFL

LibFuzzer

Syzkaller

VUzzer

kAFL

WinAFL

Related Work

AFL

LibFuzzer

Syzkaller

VUzzer

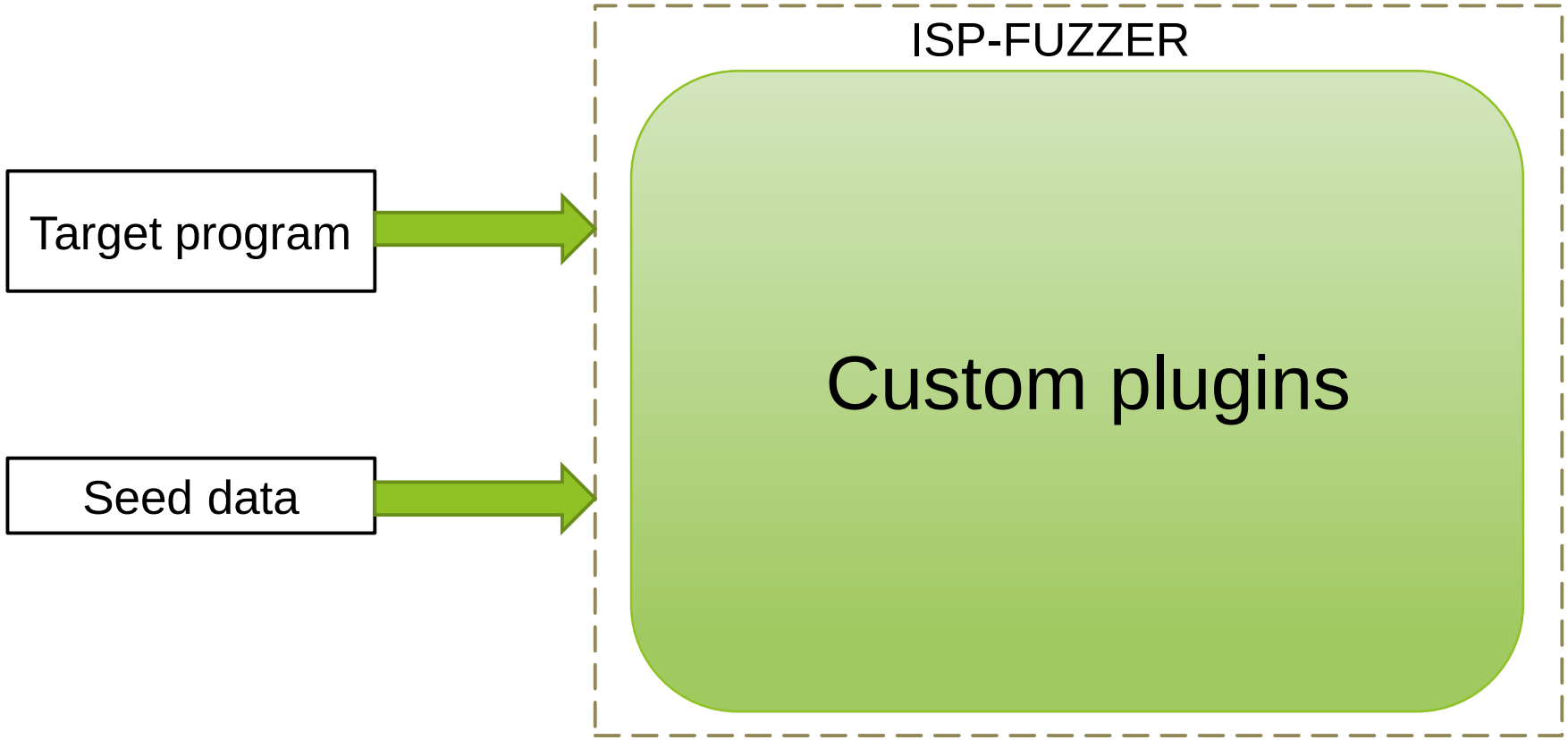
kAFL

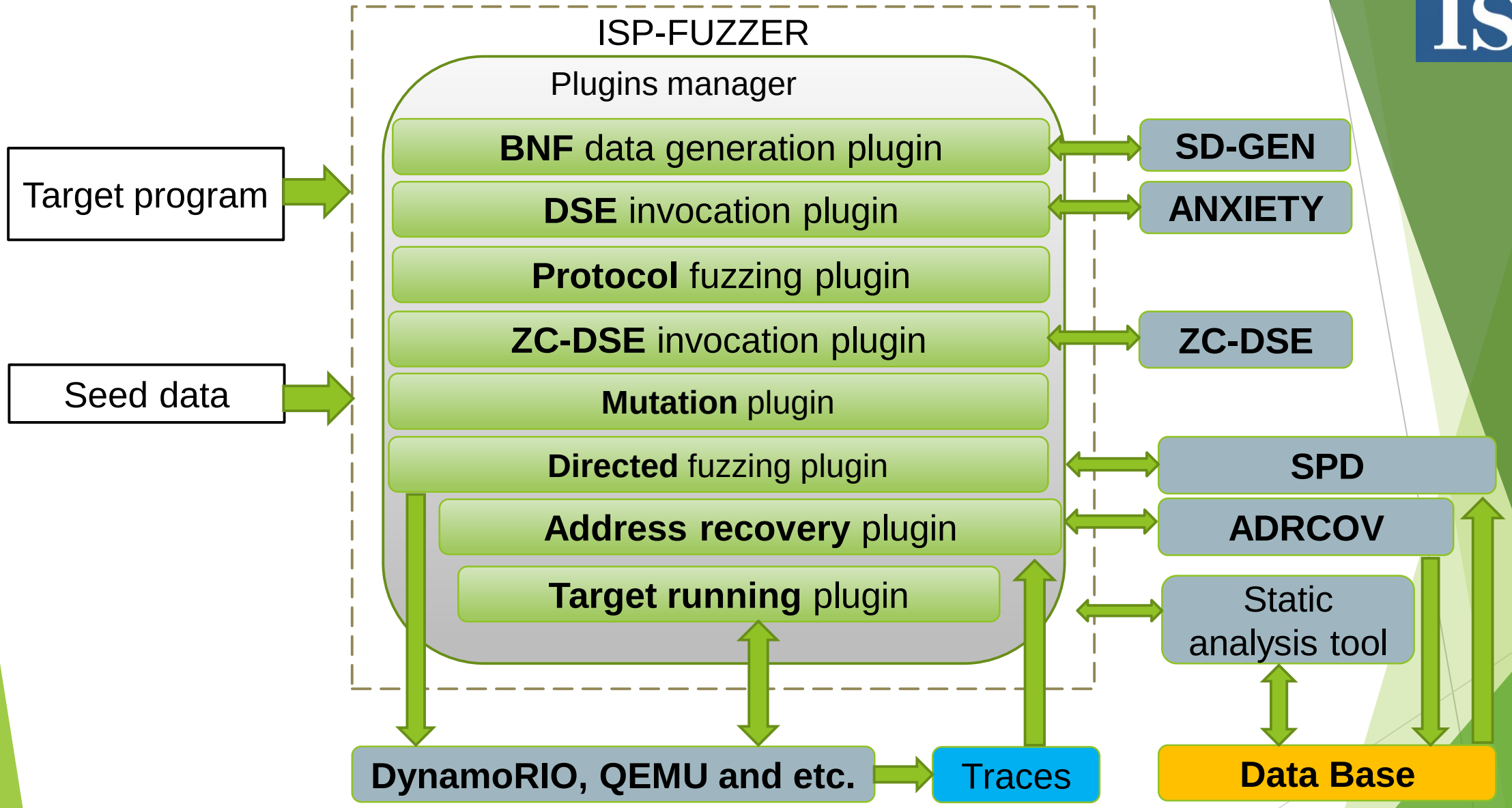
WinAFL

Goal of the Work

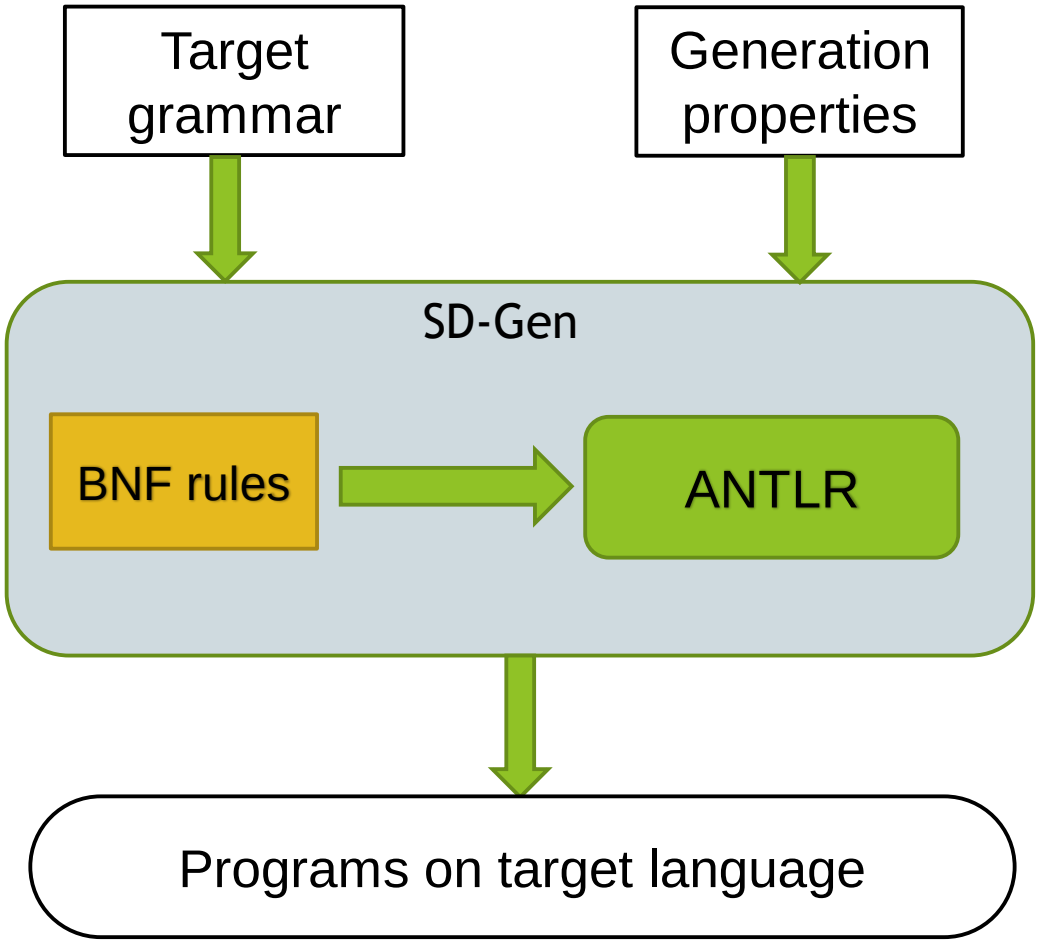
1. To create flexible and easily extendable fuzzing framework that can be applied for variety of tasks. It must be efficient and easily adoptable for new tasks (combine with static analysis, support BNF grammars fuzz, support network protocols fuzz and etc.).
2. Provide opportunity to run on distributed systems.

Infrastructure of the ISP-Fuzzer




ISP-FUZZER
Plugins manager
BNF data generation plugin
DSE invocation plugin
Protocol fuzzing plugin
ZC-DSE invocation plugin
Mutation plugin
Directed fuzzing plugin
Address recovery plugin
Target running plugin
Target program
Seed data
SD-GEN
ANXIETY
ZC-DSE
SPD
ADRCOV
Static analysis tool
DynamoRIO, QEMU and etc.
Traces
Data Base

BNF Data Generation Plugin

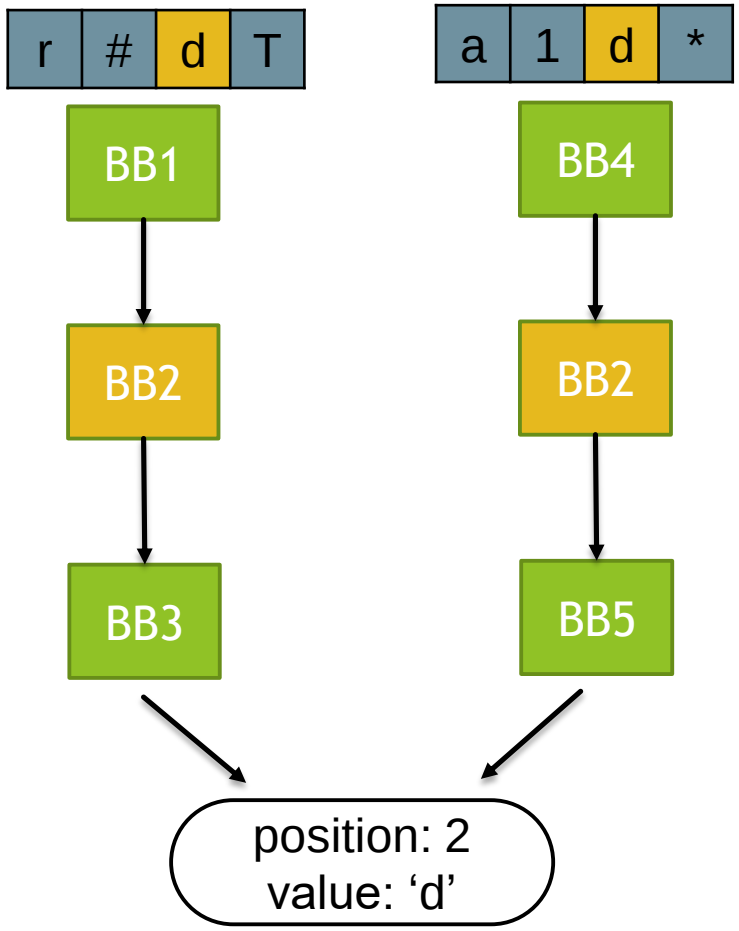


Result of BNF Data Generation Plugin

Program Name	BB coverage AFL	BB coverage ISP Fuzzer + SD-Gen	BB coverage growth (%)
gcc – 7.1	24325	26107	+4.6
g++ - 7.1	25985	30103	+15.1
python – 2.7	7521	7962	+5.3
php – 7.1.7	1997	2017	+3.5
luac – 5.3.5	12751	16274	+21.5
gfortran – 7.1	23726	24950	+3.7

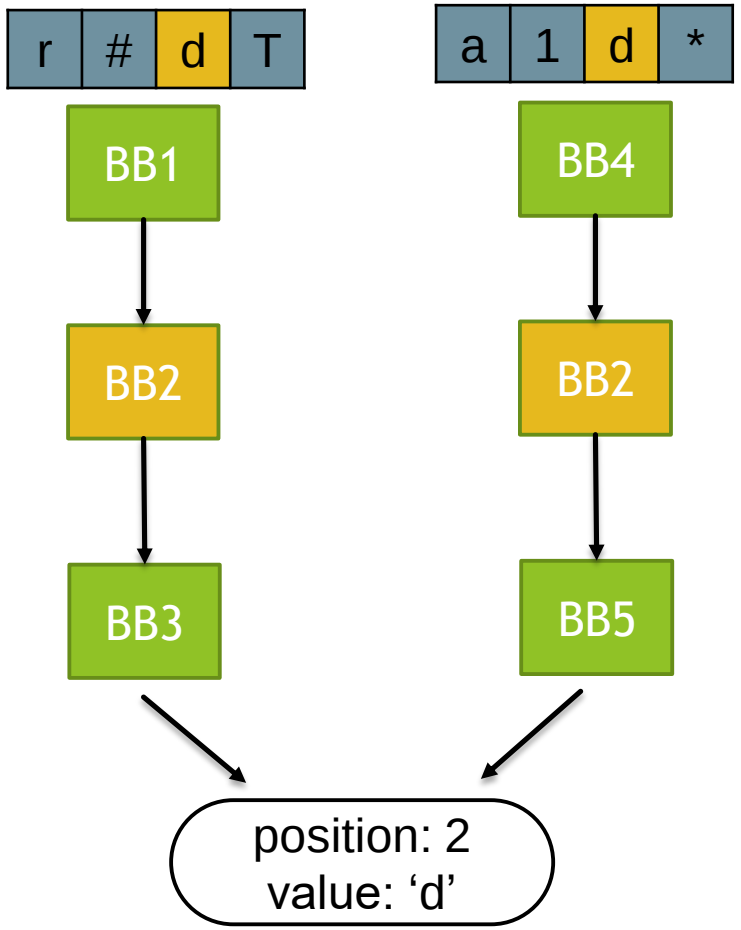
ZC-DSE Invocation Plugin

```
void foo(char* buffer) {
  int j = 0;
  if (buffer[0] == 'b') j++;
  if (buffer[1] == 'a') j++;
  if (buffer[2] == 'd') j++;
  if (buffer[3] == '!') j++;

   if (j == 4) {
     abort();
   }
}
```



ZC-DSE invocation plugin

```
void foo(char* buffer) {
  int j = 0;
  if (buffer[0] == 'b') j++;
  if (buffer[1] == 'a') j++;
  if (buffer[2] == 'd') j++;
  if (buffer[3] == '!') j++;

   if (j == 4) {
     abort();
   }
}
```

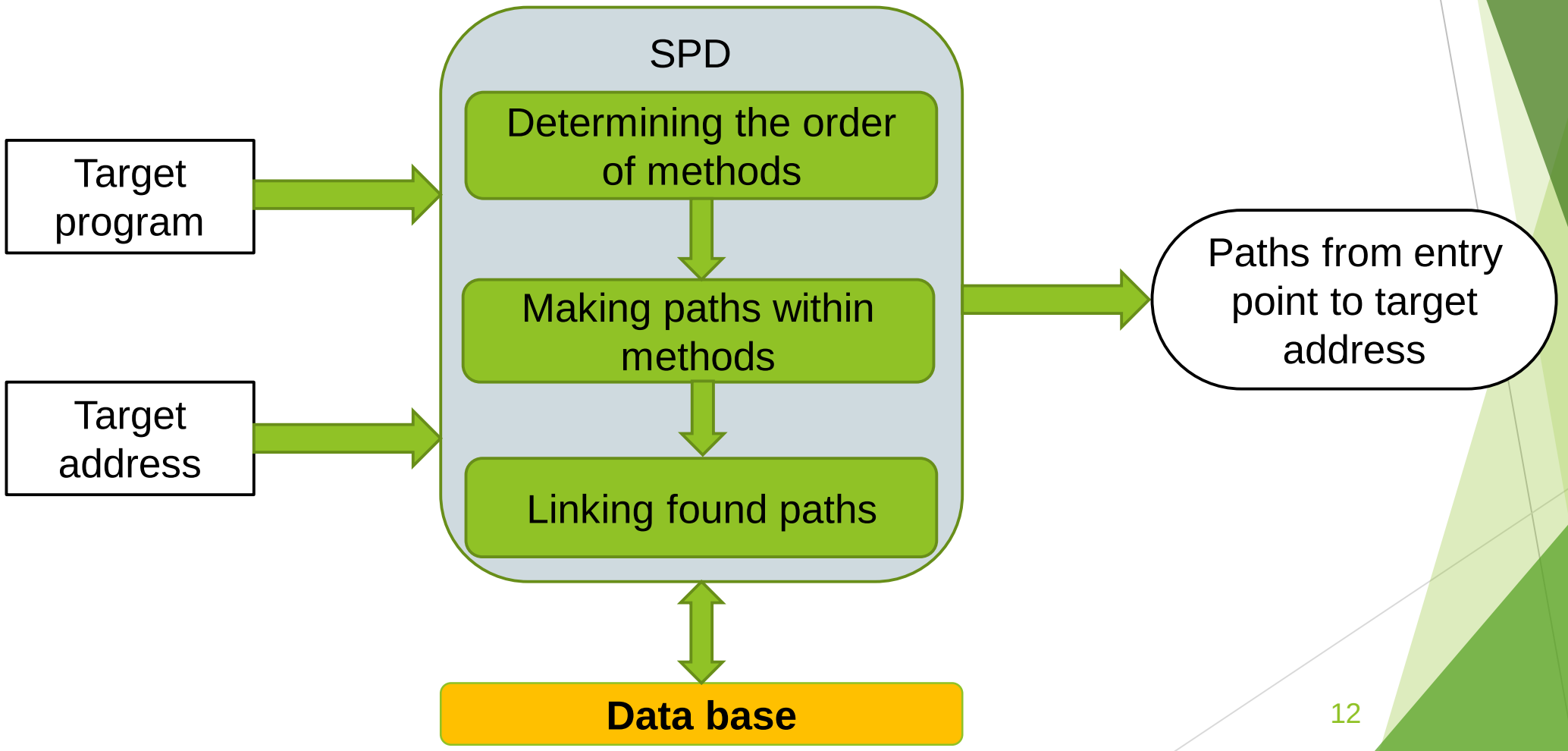


Next generated data

Result of ZC-DSE Invocation Plugin

Program name	Paths on ISP - Fuzzer	Paths on ISP-Fuzzer + ZC-DSE	Difference
pdftk	405	473	+68
tiff2pdf	86	121	+35
tiffdump	193	211	+18
tiffsplit	34	36	+2
jasper	30	33	+3
readelf	1045	2081	+1036 (1 crash)
gif2png	291	366	+75 (37 crash)
djpeg	40	51	+11 (1 hang)

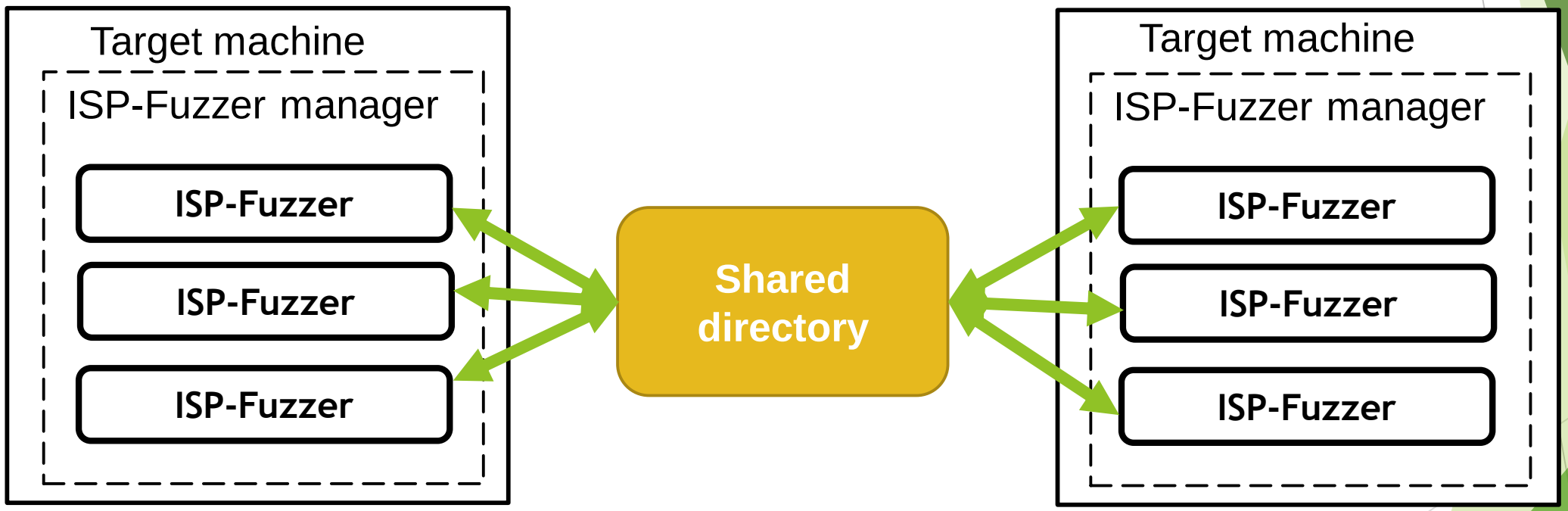
Directed Fuzzing Plugin – SPD



Result of Directed Fuzzing Plugin

Programs from DARPA challenges	AFL crashes or hangs	ISP-Fuzzer + SPD crashes or hangs
Personal_Fitness_manager	0	2
Humaninterface	0	1
H20Flowinc	0	1
Stream_vm2	2	0
3D_Image_Tool_kit	0	5
Flash_File_System	43	13
SAuth	1	1
ASL6parse	0	1
Single-Sign-On	0	2
ECM_TCM_Simulator	0	2

Parallel and Distributed Run



Results

Designed and developed framework for fuzzing which is:

- ✓ Extendable
- ✓ Supports multiply platforms
- ✓ Ability to fuzz through differ types of inputs (file, network protocol, standard input, environment variable and etc.)
- ✓ Distributed execution.

I will be glade to answer your questions!

Thank you for attention!