DIFUZE: Interface Aware Fuzzing for Kernel Drivers

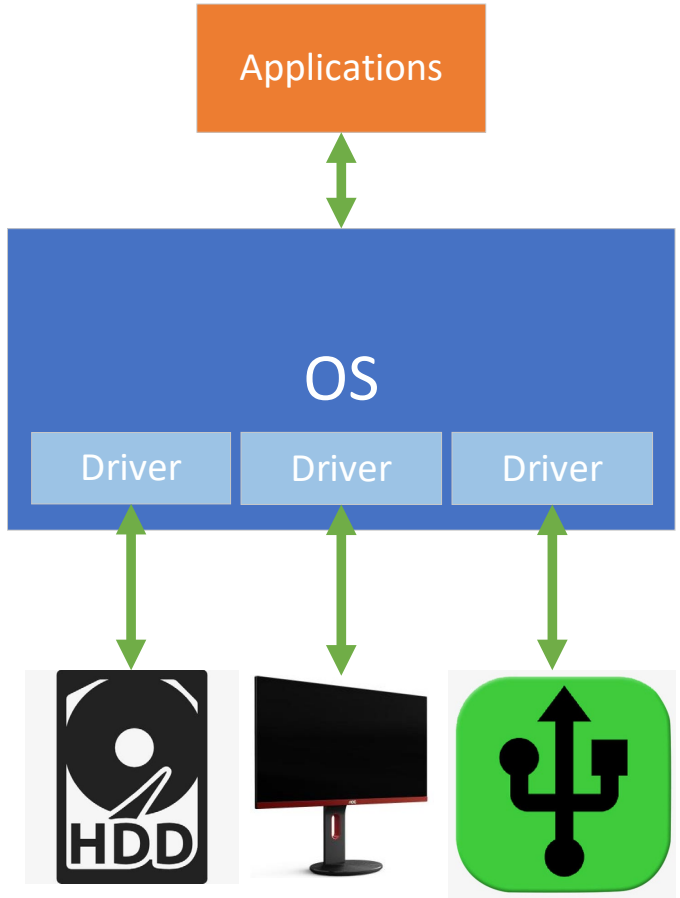
ACM CCS 2017
Oct 30th–Nov 3rd, Dallas, USA

Jake Corina Aravind Machiry Christopher Salls Yan Shoshitaishvili
Shuang Hao Christopher Kruegel Giovanni Vigna

Jingtang Zhang
2019.09

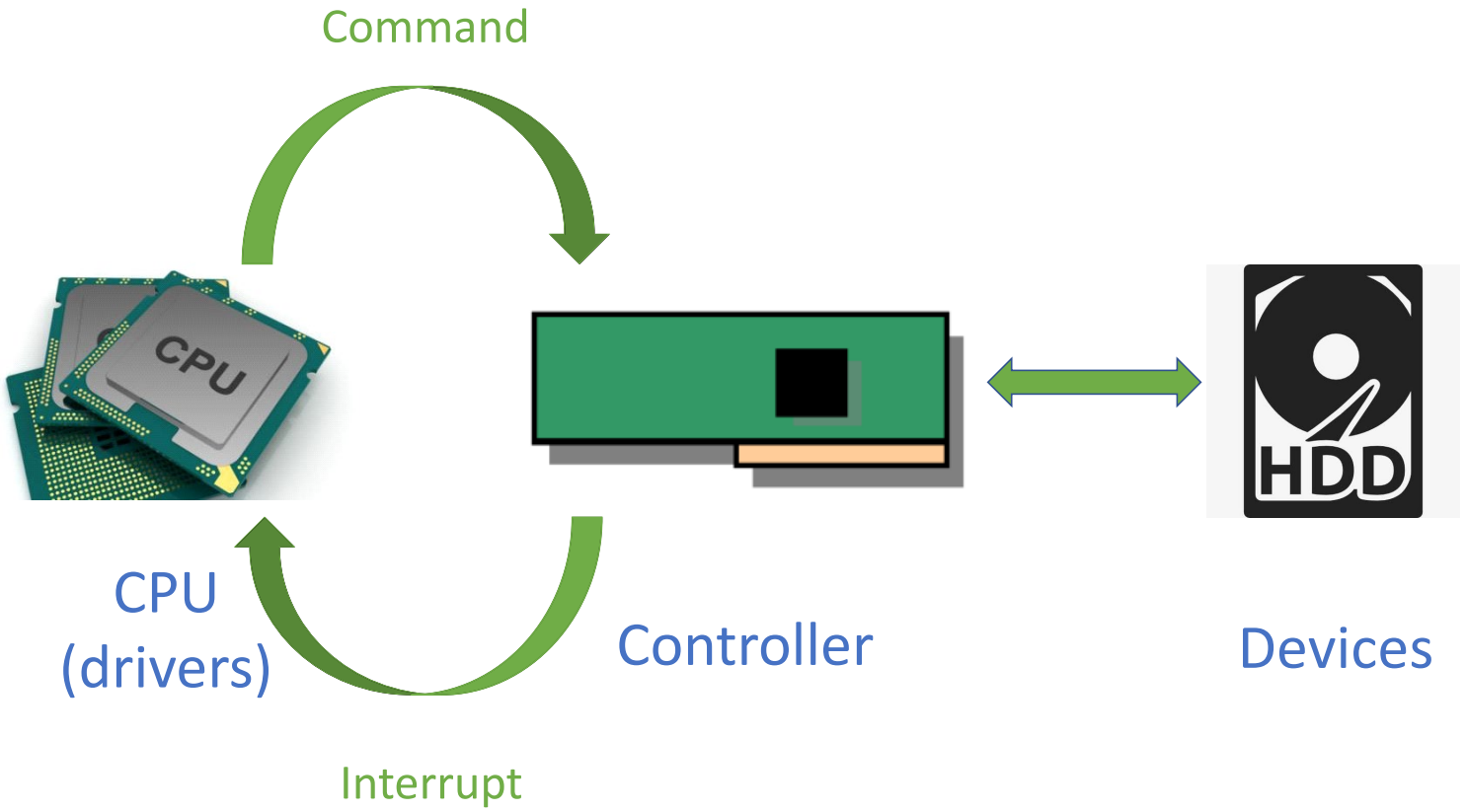
Drivers

- Part of OS kernel
- Operate or control a particular type of device that is attached to a computer



Drivers

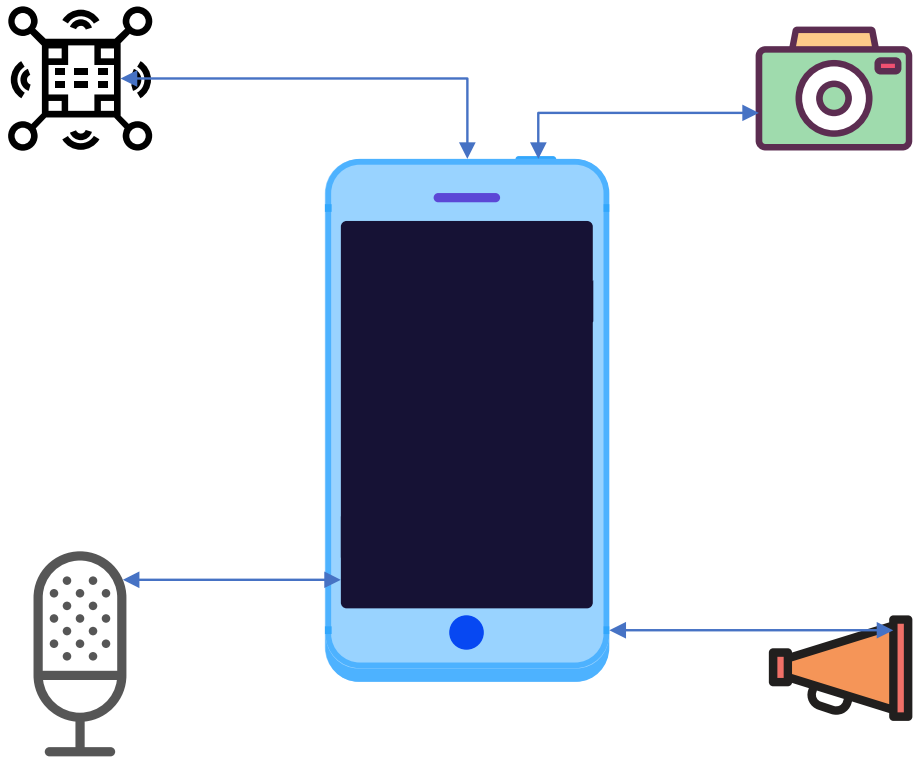
- Part of OS kernel
- Operates or controls a particular type of device that is attached to a computer



Drivers

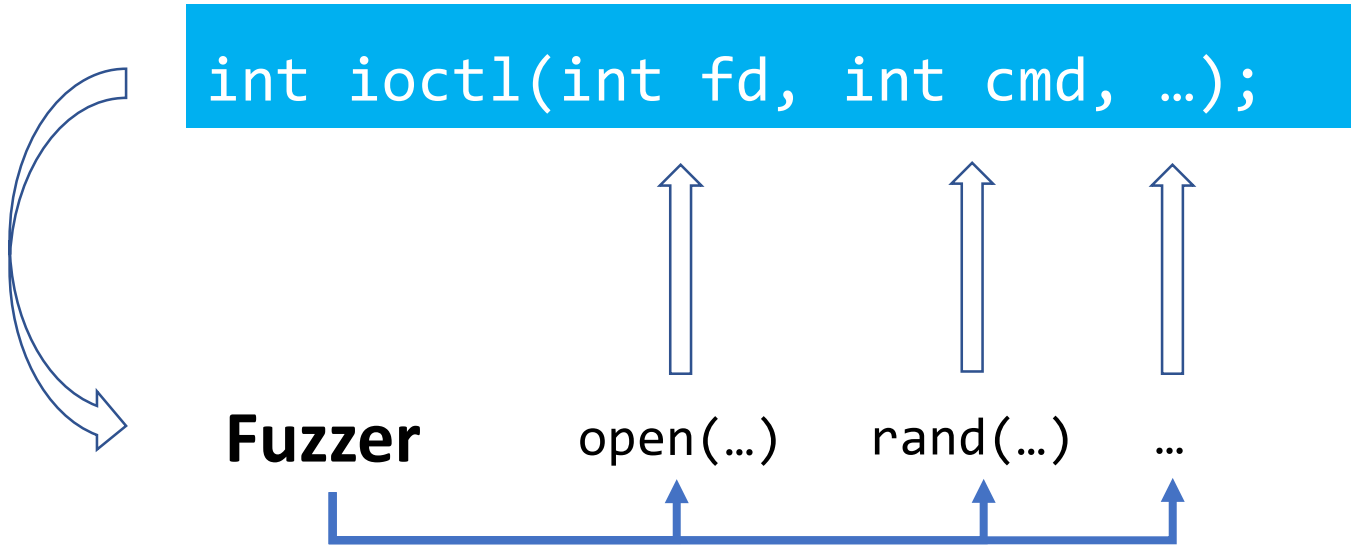
- Part of OS kernel
- Customization
- Monolithic OS

- 80% bugs of Android kernel



POSIX standard

- Portable OS Interface for Computing Systems



Issues

- Complex
- Non-structured
- Non-standard

```
int ioctl(int fd, int cmd, …);
```

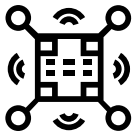
record

setting

```
struct1 {
    ……
};
```

……

```
struct2 {
    ……
};
```



……

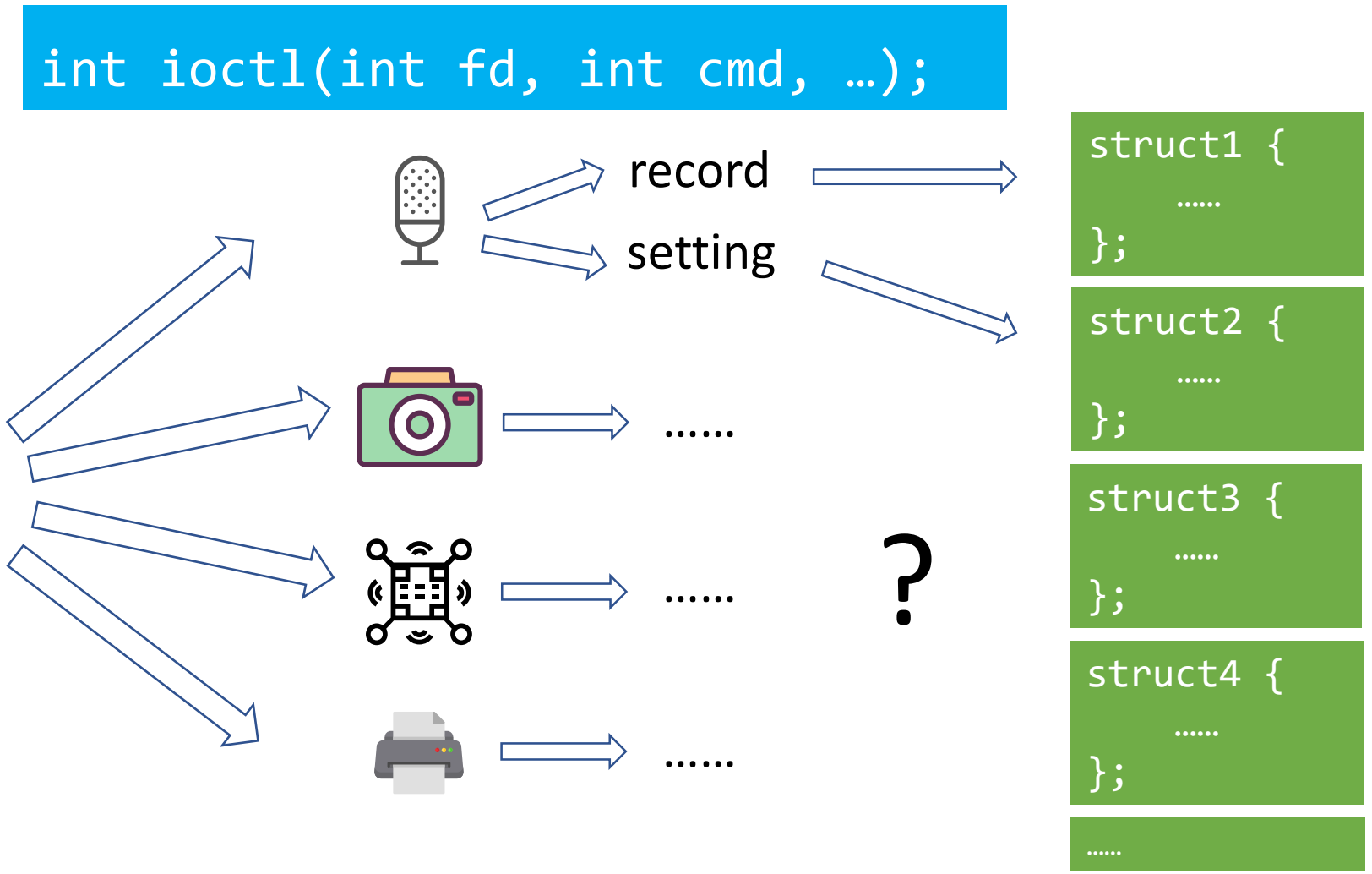
```
struct3 {
    ……
};
```

……

```
struct4 {
    ……
};
```

DIFUZE

Kernel source code

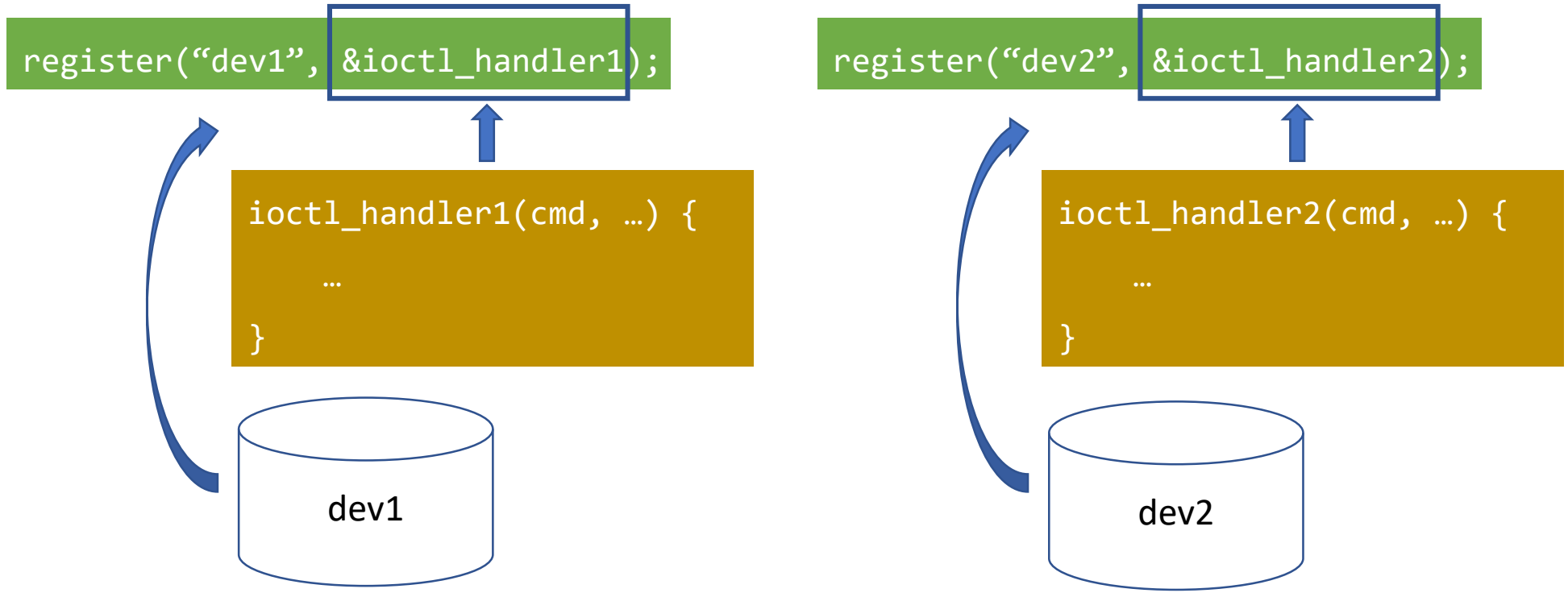


DIFUZE

User Space

```
int ioctl(int fd, int cmd, …);
```

Kernel

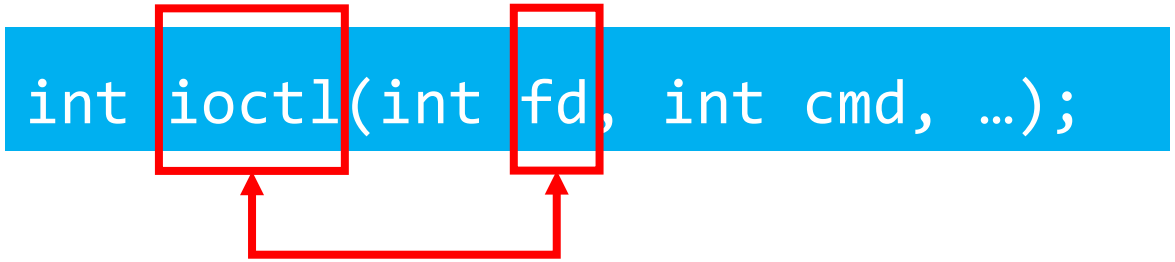


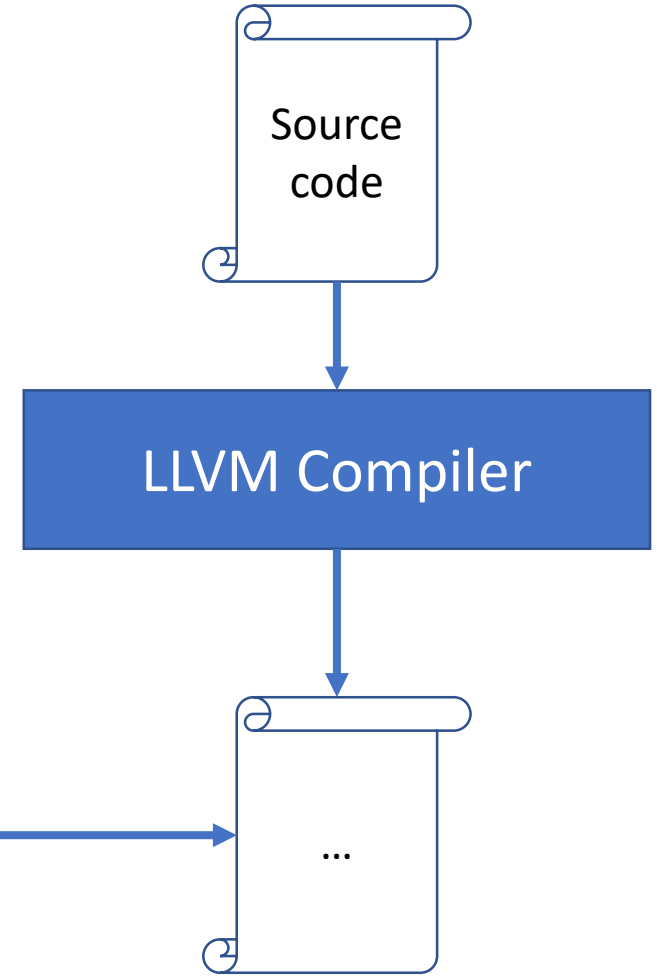
DIFUZE – Handler Identification

```
register("dev1", &ioctl_handler1);
```

- Block device
- Character device
- Proc device
- …

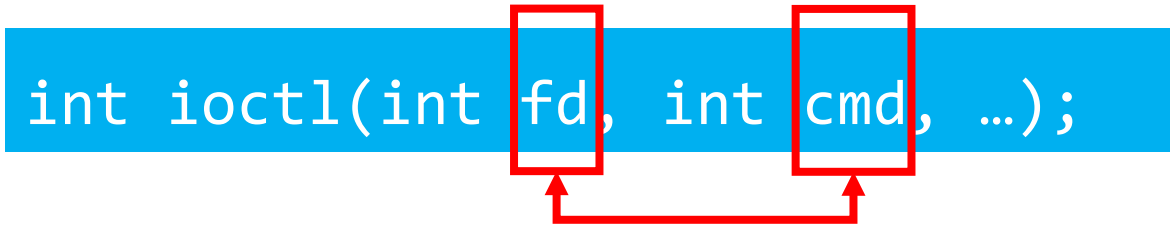
```
int ioctl(int fd, int cmd, …);
```





DIFUZE – Command Determination

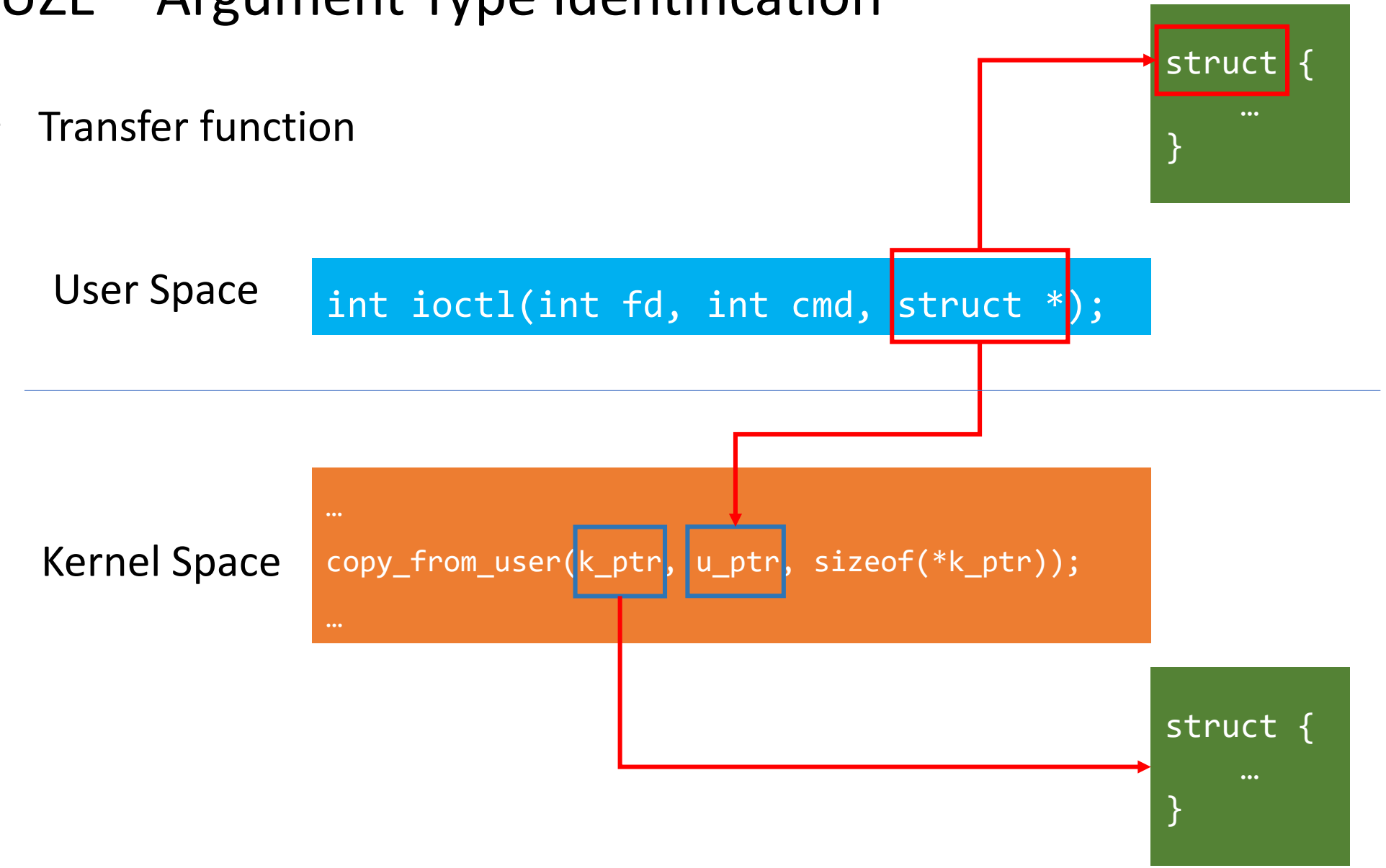
- Equality constraints




```
ioctl_handler(int cmd, …) {
    switch (cmd) {
        case 0x1001:
            …
            if (cmd == 0x1001) {
                …
            }
            …
        case 0x1002:
            …
    }
}
```

DIFUZE – Argument Type Identification

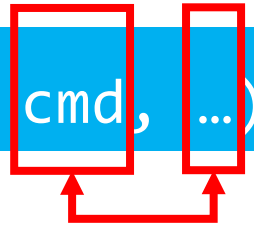
- Transfer function



DIFUZE – Argument Type Identification

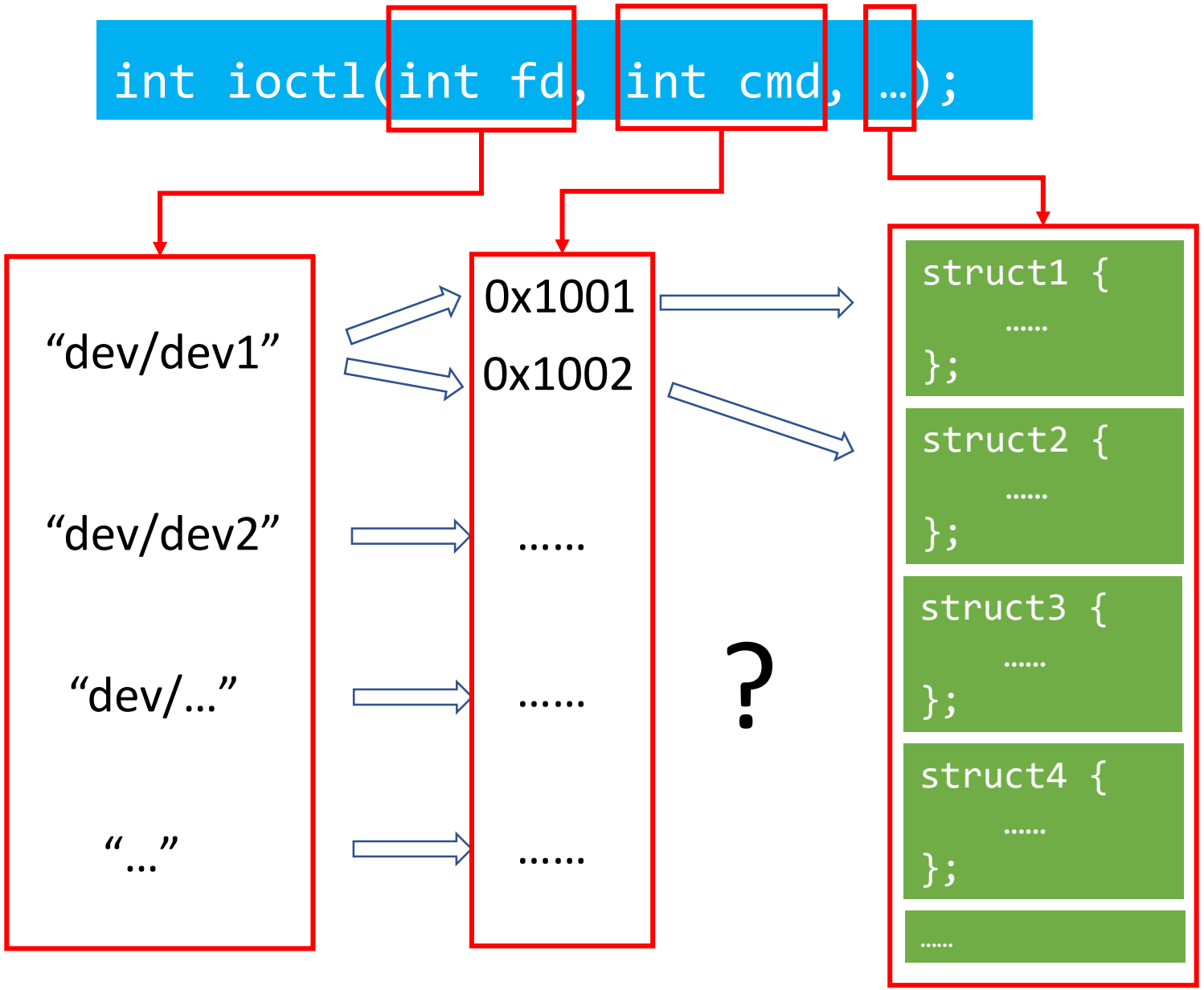
- Equal constraint on the path to transfer function

```
int ioctl(int fd, int cmd, …);
```



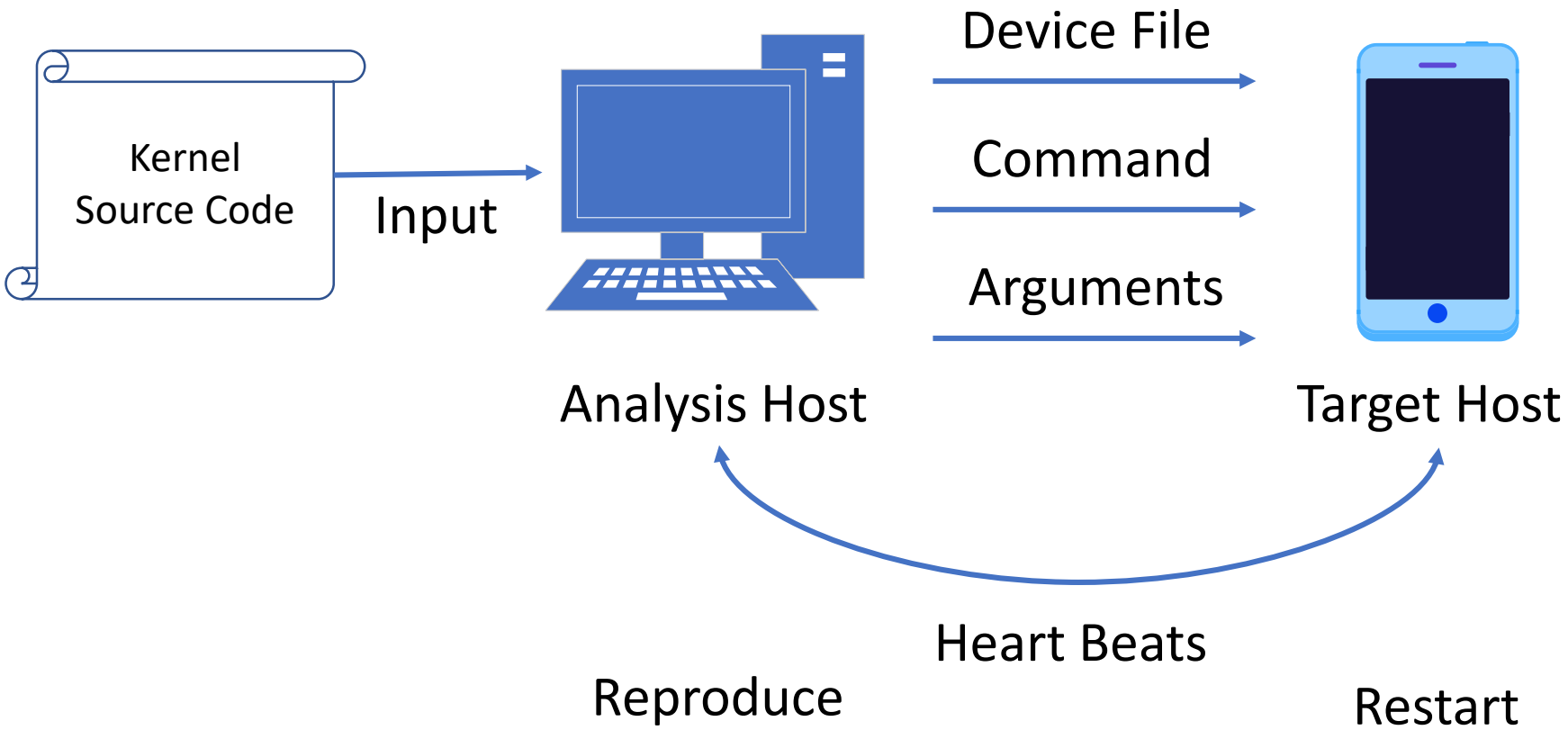
```
ioctl_handler(int cmd, …) {
    switch (cmd) {
        case 0x1001:
            …
            copy_from_user(…);
            …
        case 0x1002:
            …
    }
}
```

DIFUZE


int ioctl(int fd, int cmd, …);
“dev/dev1”
“dev/dev2”
“dev/…”
“…”
0x1001
0x1002
……
……
……
?
struct1 {
……
};
struct2 {
……
};
struct3 {
……
};
struct4 {
……
};
……

DIFUZE

```
int ioctl(int fd, int cmd, …);
```



Results

Syzkaller	DIFUZE
(without interface information)	(interface-aware)
0 bugs	36 bugs

Weaknesses

Dynamically generated device name

```
1   static struct cdev driver_devc;
2   static dev_t client_devt;
3   static struct file_operations driver_ops;
4   __init int driver_init(void)
5   {
6     // request minor number
7     alloc_chrdev_region(&driver_devt, 0, 1, "example_device");
8     // set the ioctl handler for this device
9     driver_ops.unlocked_ioctl = ioctl_handler;
10    cdev_init(&driver_devc, &driver_ops);
11    // register the corresponding device.
12    cdev_add(&driver_devc, MKDEV(MAJOR(driver_devt), 0), 1);
13  }
```

Weaknesses

Dynamically generated device name

```
1    VOS_INT __init RNIC_InitNetCard(VOS_VOID) {
2      ...
3      snprintf(pstDev->name, sizeof(pstDev->name),
4               "%s%s",
5               RNIC_DEV_NAME_PREFIX,
6               g_astRnicManageTbl[ucIndex].pucRnicNetCardName);
7      ...
8    }
```

Weaknesses

Inability to extract complex relationships between fields of structures

```
struct args {
    ……
    char *buffer;
    int buffer_len;
    ……
};
```

DIFUZE: Interface Aware Fuzzing for Kernel Drivers

ACM CCS 2017
Oct 30th–Nov 3rd, Dallas, USA

Thanks.