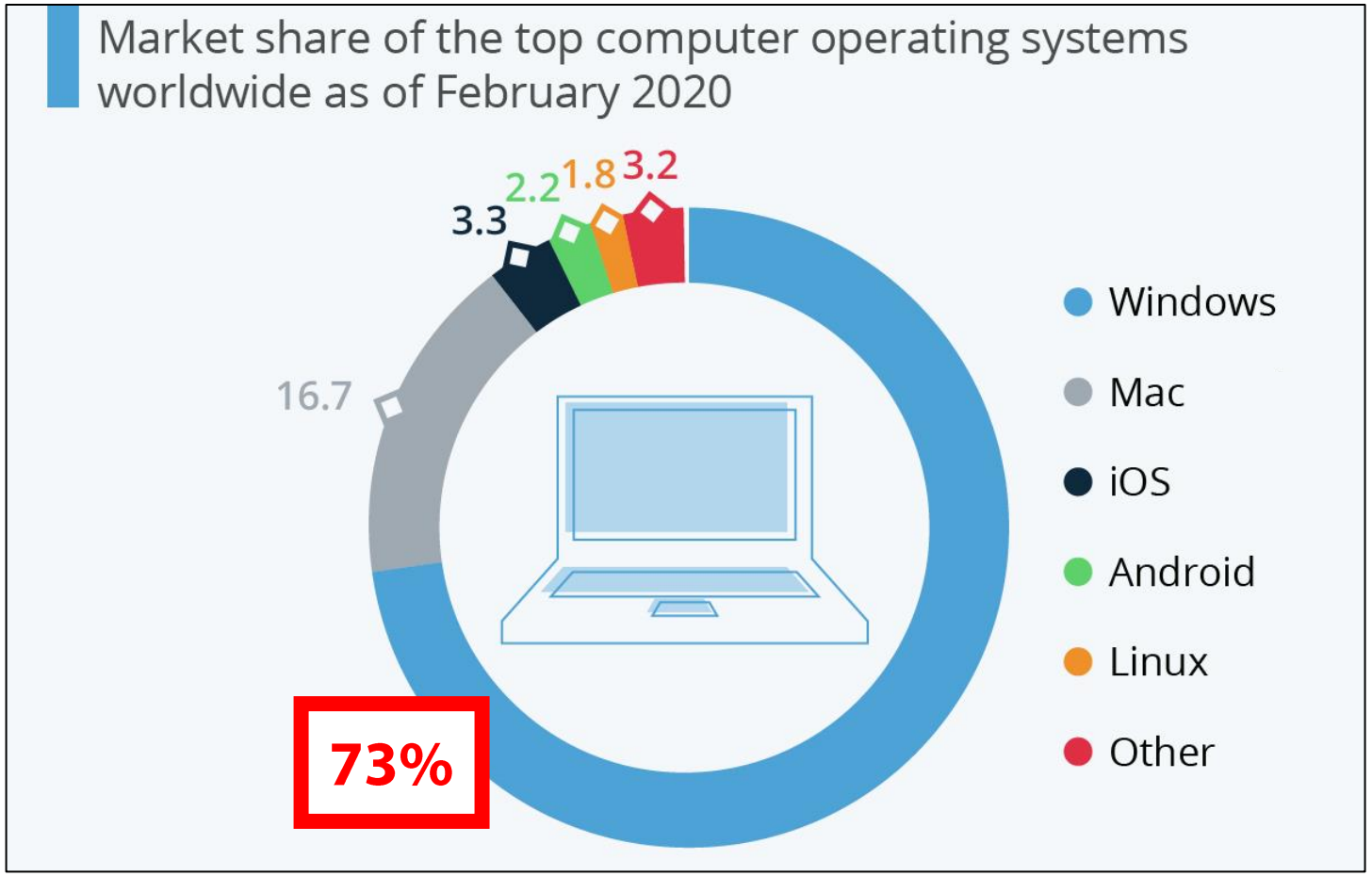
# WINNIE: Fuzzing Windows Applications with Harness Synthesis and Fast Cloning

Jinho Jung, Stephen Tong, Hong Hu*,
Jungwon Lim, Yonghwi Jin, Taesoo Kim


PennState*

# WINDOWS OS STILL DOMINATES MARKET SHARE




https://www.statista.com/chart/21244/global-market-share-of-operating-systems/

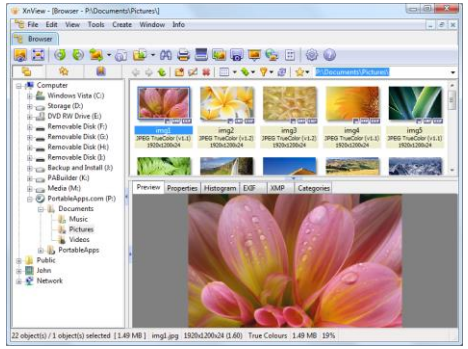
# MANY APPS ARE WAITING FOR BEING TESTED!

# WINDOWS APP IS NOT FUZZING-FRIENDLY

**Install**

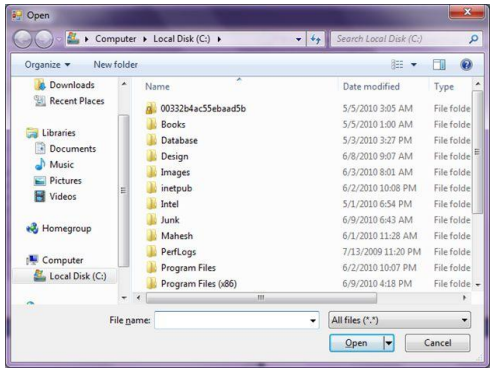
Installer.exe

**Main Application**



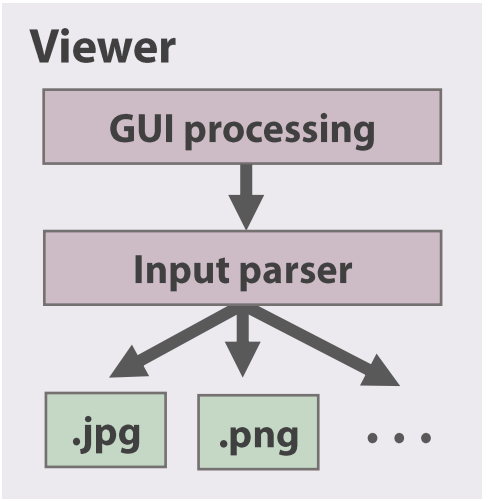
Viewer.exe

**User interaction**



Dialog Window

**Input processing**

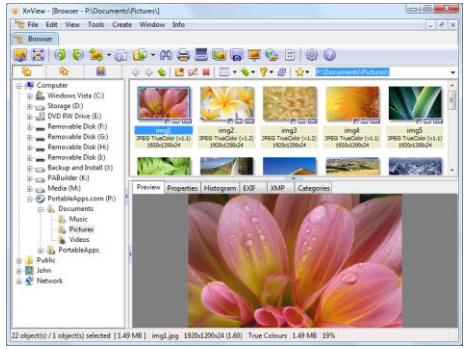


# WINDOWS APP IS NOT FUZZING-FRIENDLY

**Install**

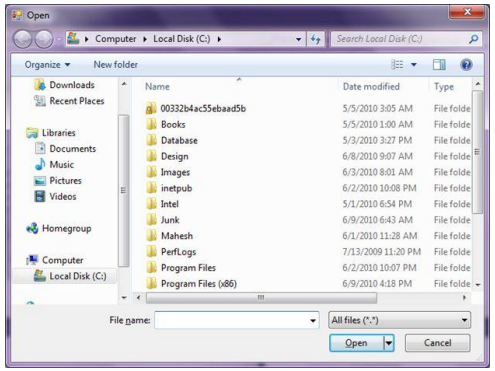
Installer.exe

**Main Application**



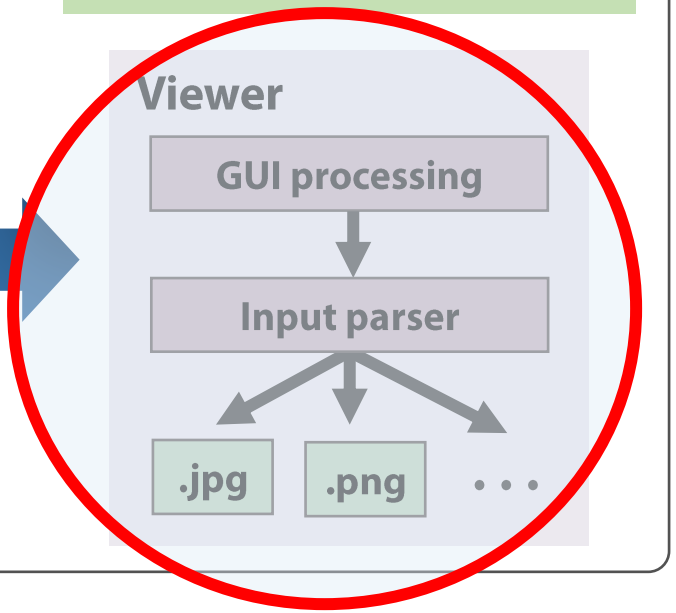
Viewer.exe

**User interaction**




Dialog Window

**Input processing**

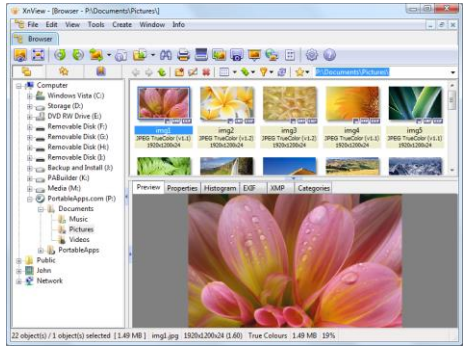



**This is what we want to test!**

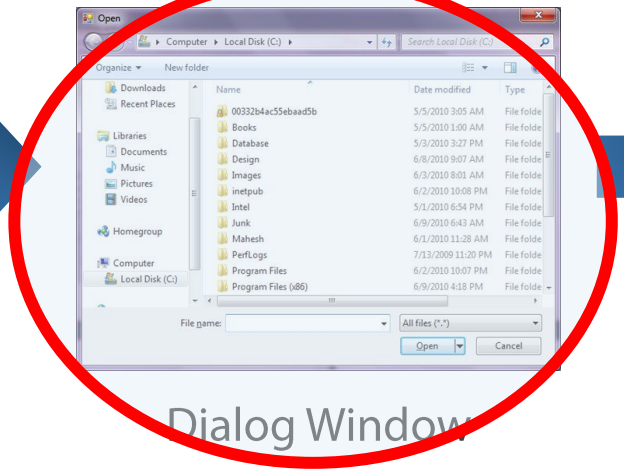
# WINDOWS APP IS NOT FUZZING-FRIENDLY

**Install** | **Main Application** | **User interaction** | **Input processing**

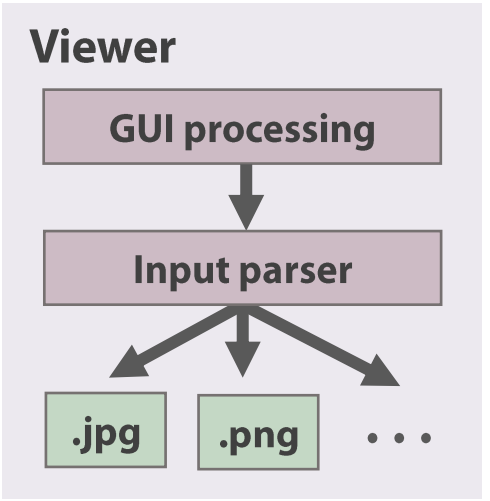
Installer.exe



Viewer.exe



Dialog Window

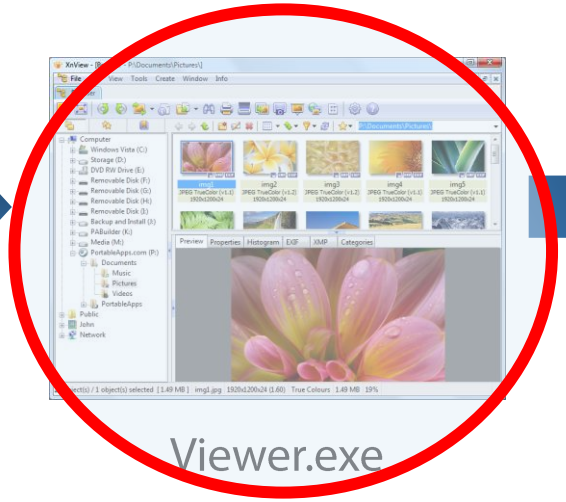



**GUI: user interaction, non-terminated**

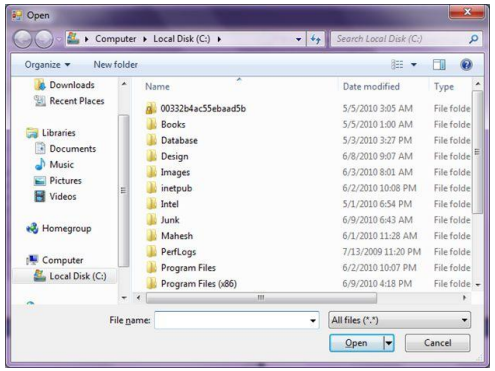
# WINDOWS APP IS NOT FUZZING-FRIENDLY

**Install** | **Main Application** | **User interaction** | **Input processing**

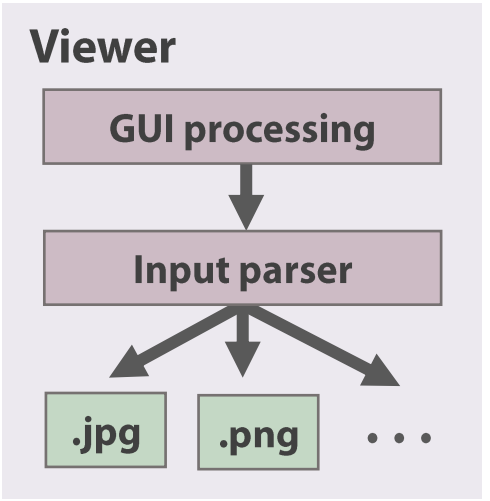
Installer.exe



Viewer.exe



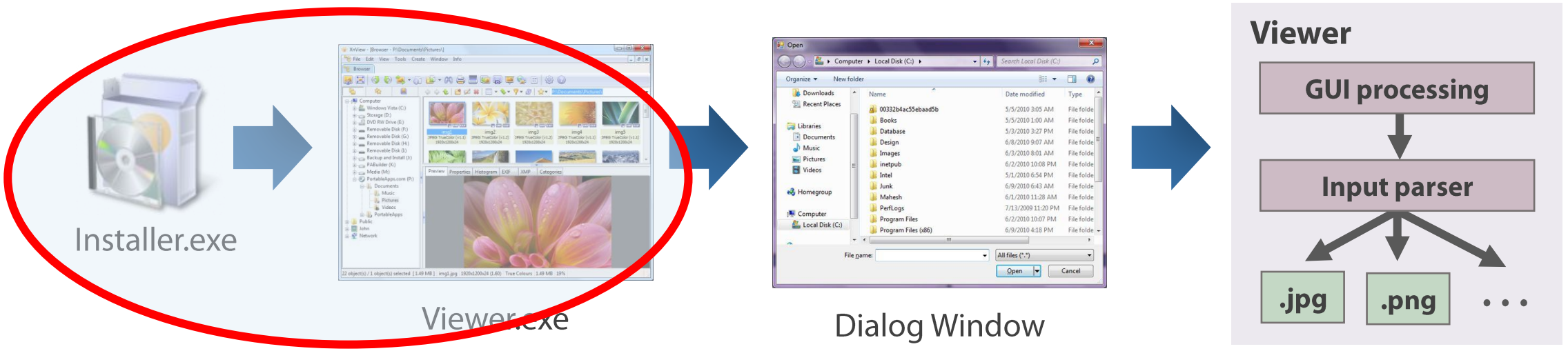
Dialog Window




**Slow speed**: heavy GUI, lack of fast cloning

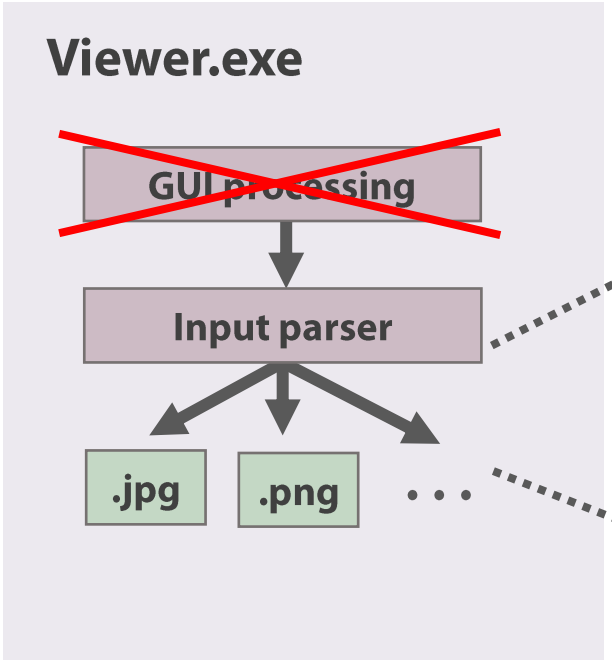
# WINDOWS APP IS NOT FUZZING-FRIENDLY

**Install** | **Main Application** | **User interaction** | **Input processing**




**Closed-source: difficult to infer internal context**

# EXISTING SOLUTION: HARNESS GENERATION



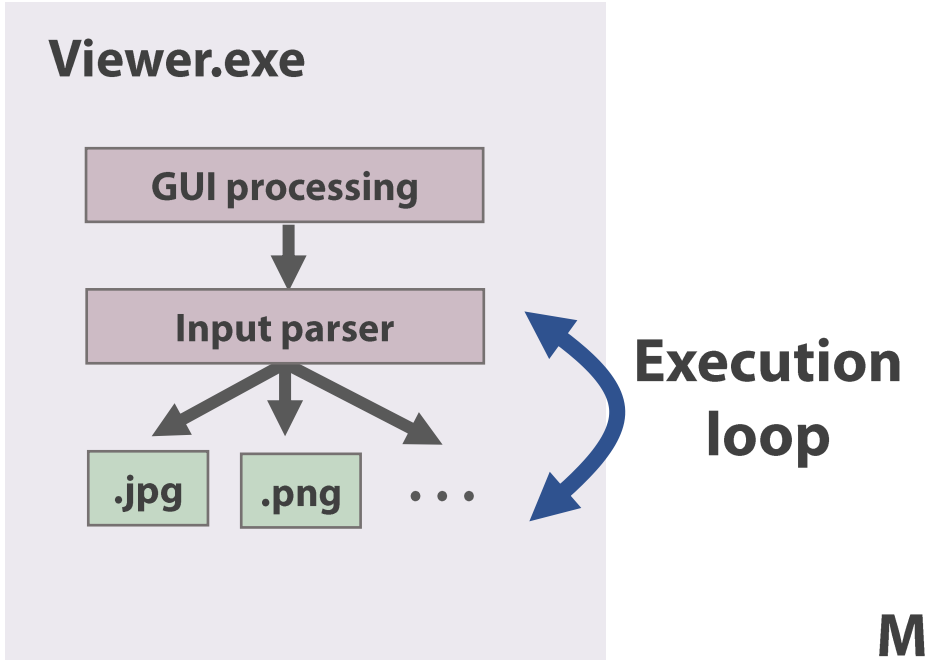

Extract parsing routine

Harness.exe

```
image_obj = IMAGE_Init (filename);
metadata = IMAGE_GetMetaData(&image_obj);
IMAGE_OpenImage(&image_obj);
IMAGE_GetPageInfo(&image_obj, &metadata);
IMAGE_CloseImage(&image_obj);
```

**UnScalable: significant manual effort (w/o src)**

# EXISTING SOLUTION: PERSISTENT FUZZING

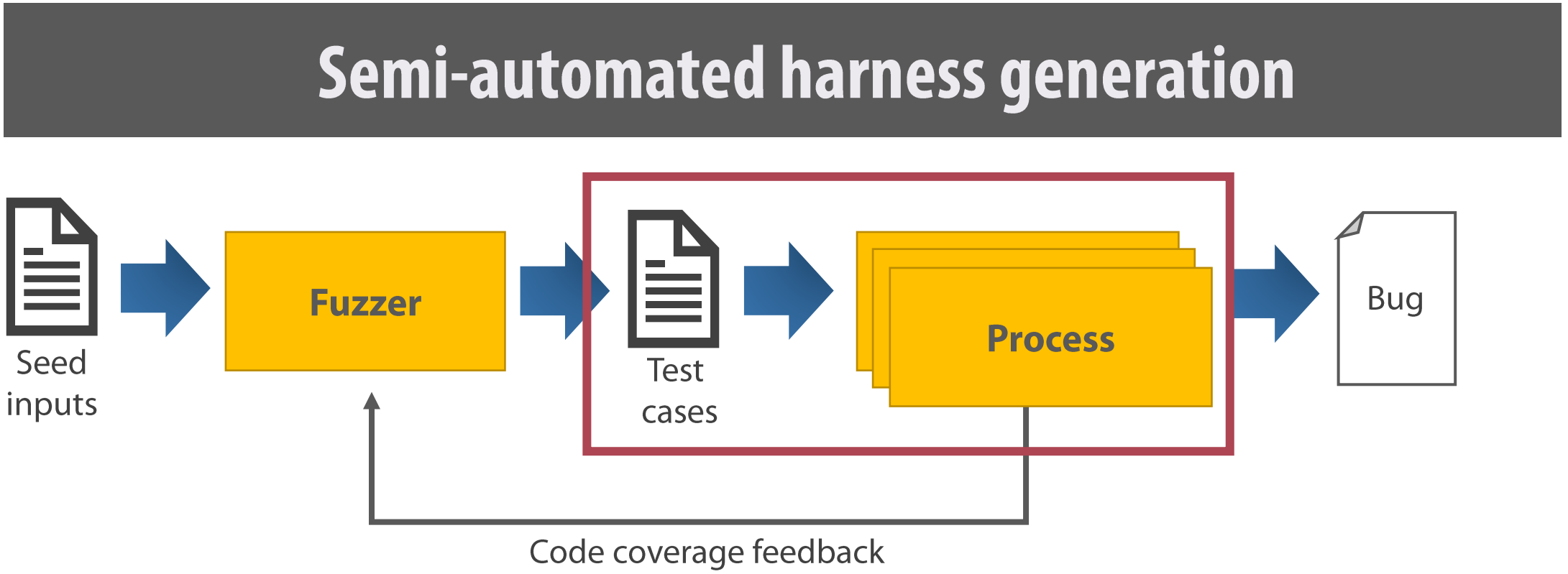



Missing fork() mechanism on Windows OS

**UnStable**: execution may corrupt global program state

# SOLUTION: GRAPHICAL INTERFACES

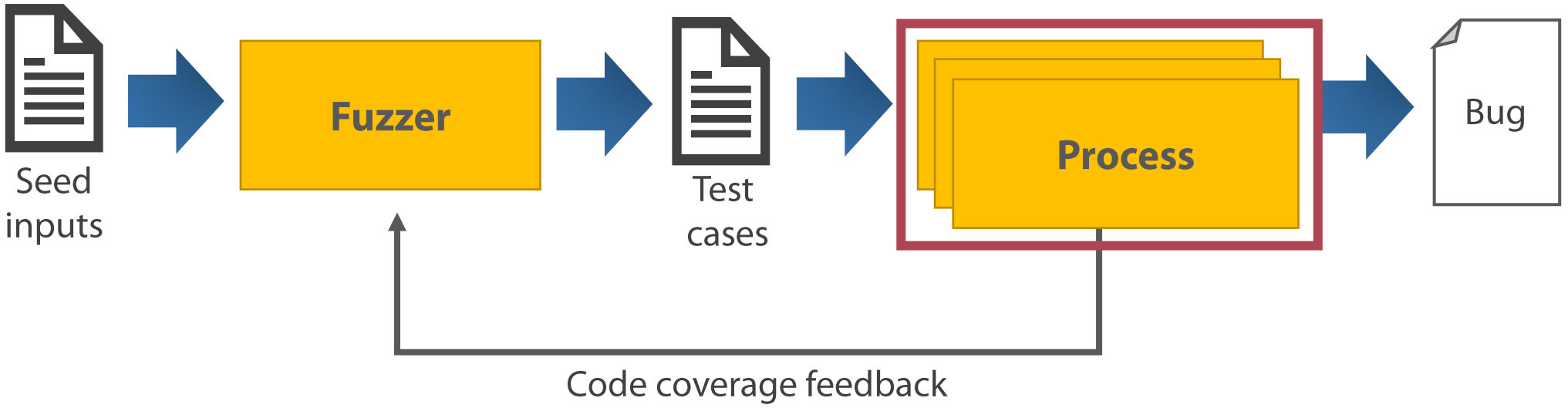
## 1 How to address user interaction and termination?



# SOLUTION: LACK OF CLONING MACHINERY

## ❷ How to achieve fast execution on windows?

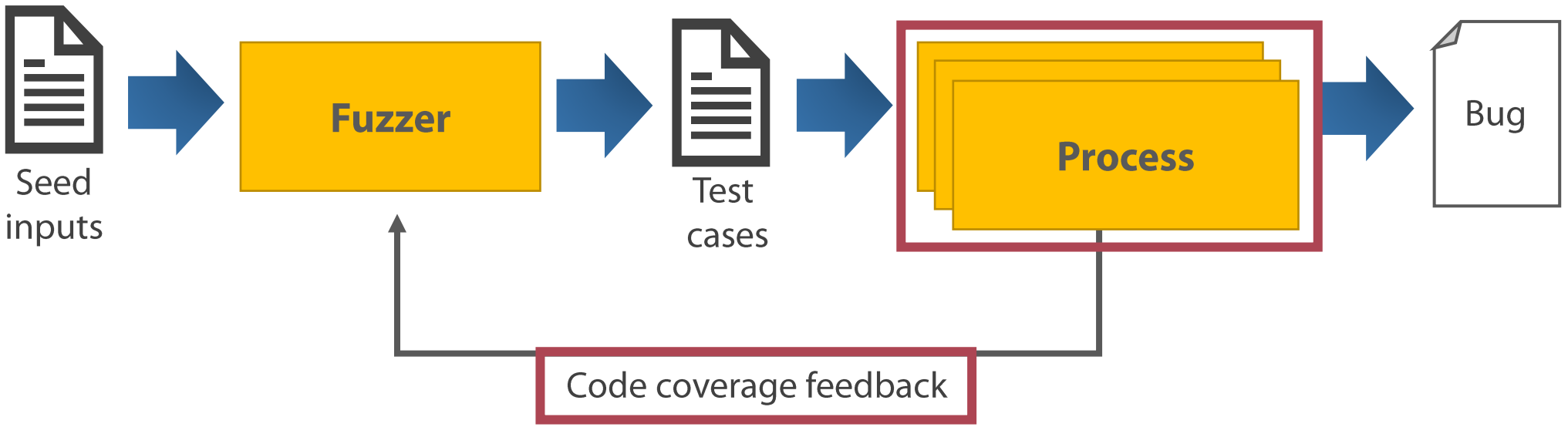
**Windows version of fork() mechanism**



# SOLUTION: CLOSED-SOURCE ECOSYSTEM

## 3 How to collect internal context of program?

**Hybrid analysis and Fullspeed fuzzing**

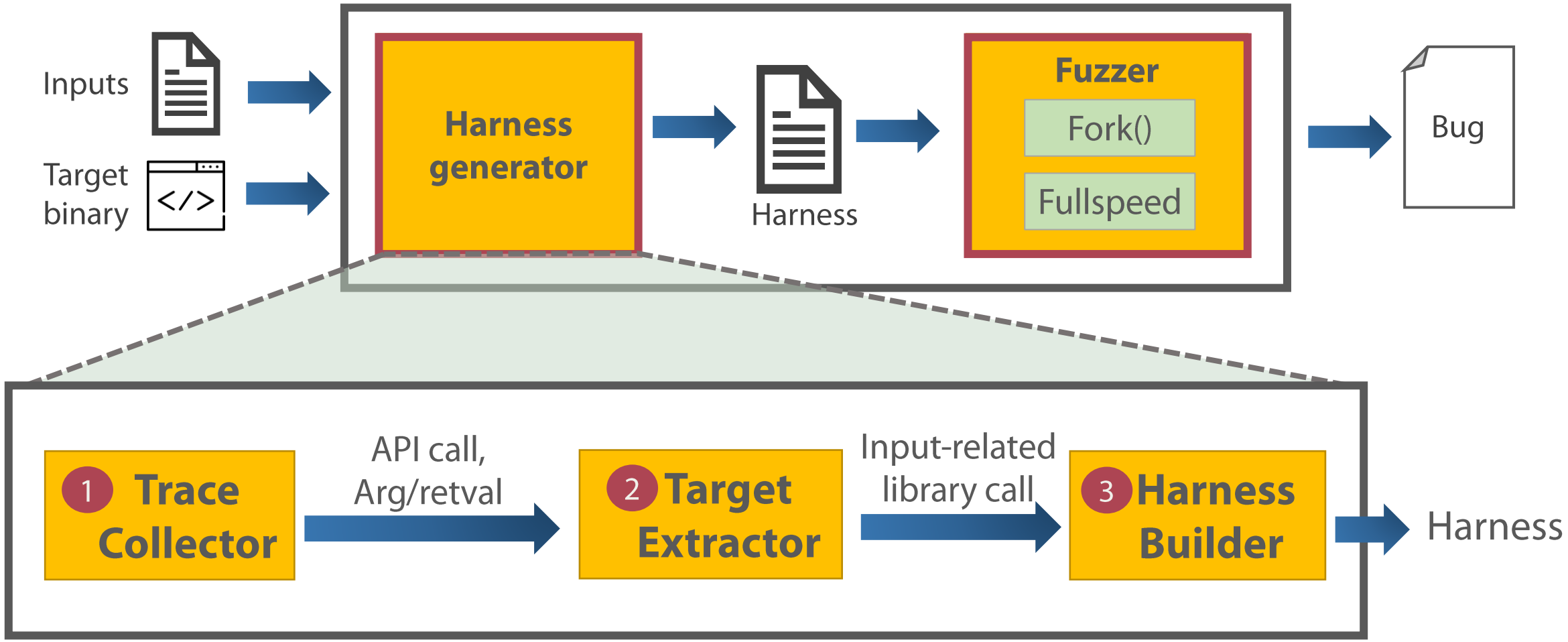


# WINNIE SYSTEM

1. Semi-automated fuzzing harness generator

2. A Practical Windows fuzzer

# WINNIE TOOLCHAIN OVERVIEW



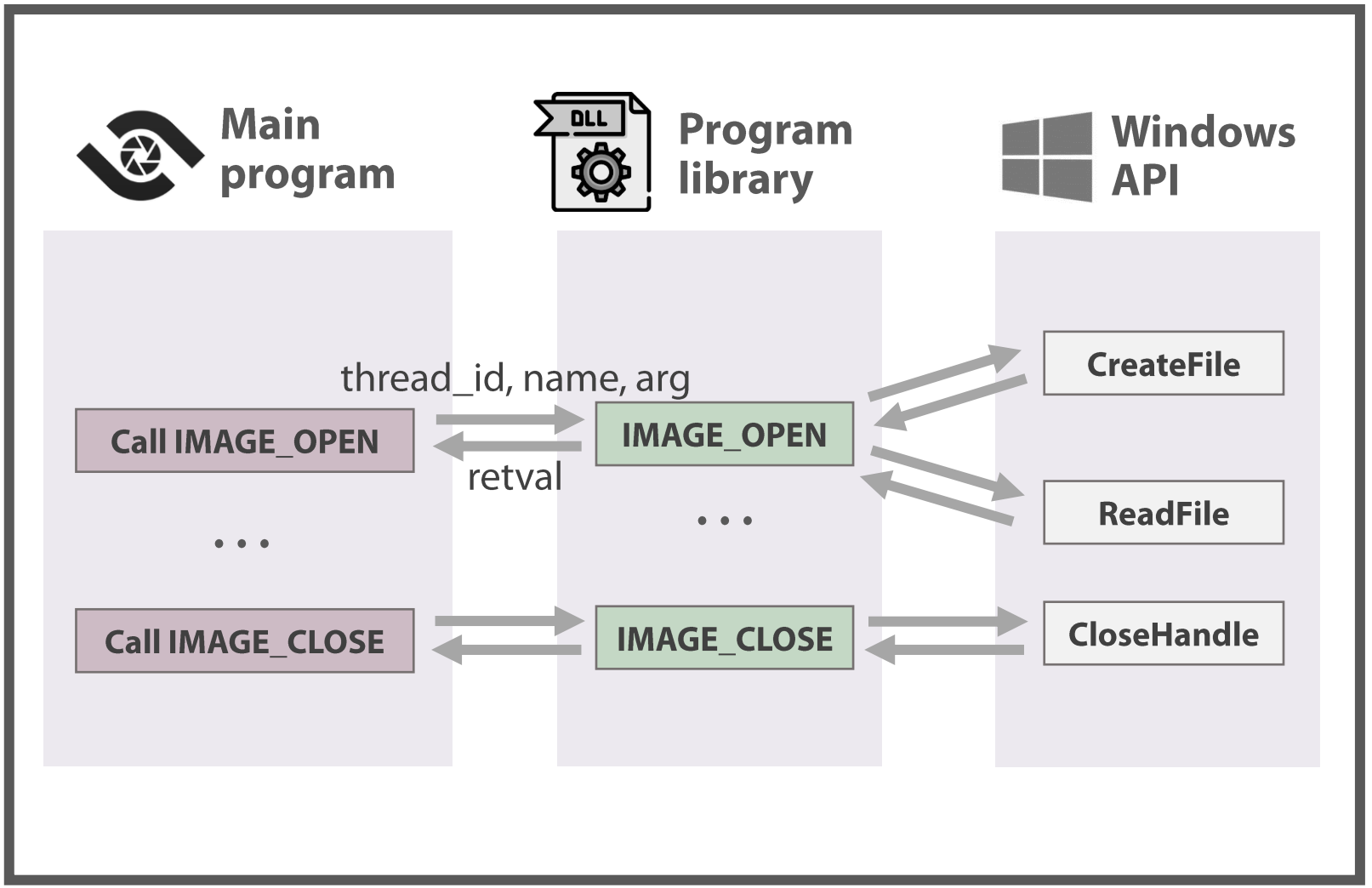
# WINNIE: SEMI-AUTOMATED HARNESS GENERATOR

**Harness generator**

- **TRACE COLLECTOR**
- TARGET EXTRACTOR
- HARNESS BUILDER

**Fuzzer**

- WINDOWS FORK()
- FULLSPEED FUZZING



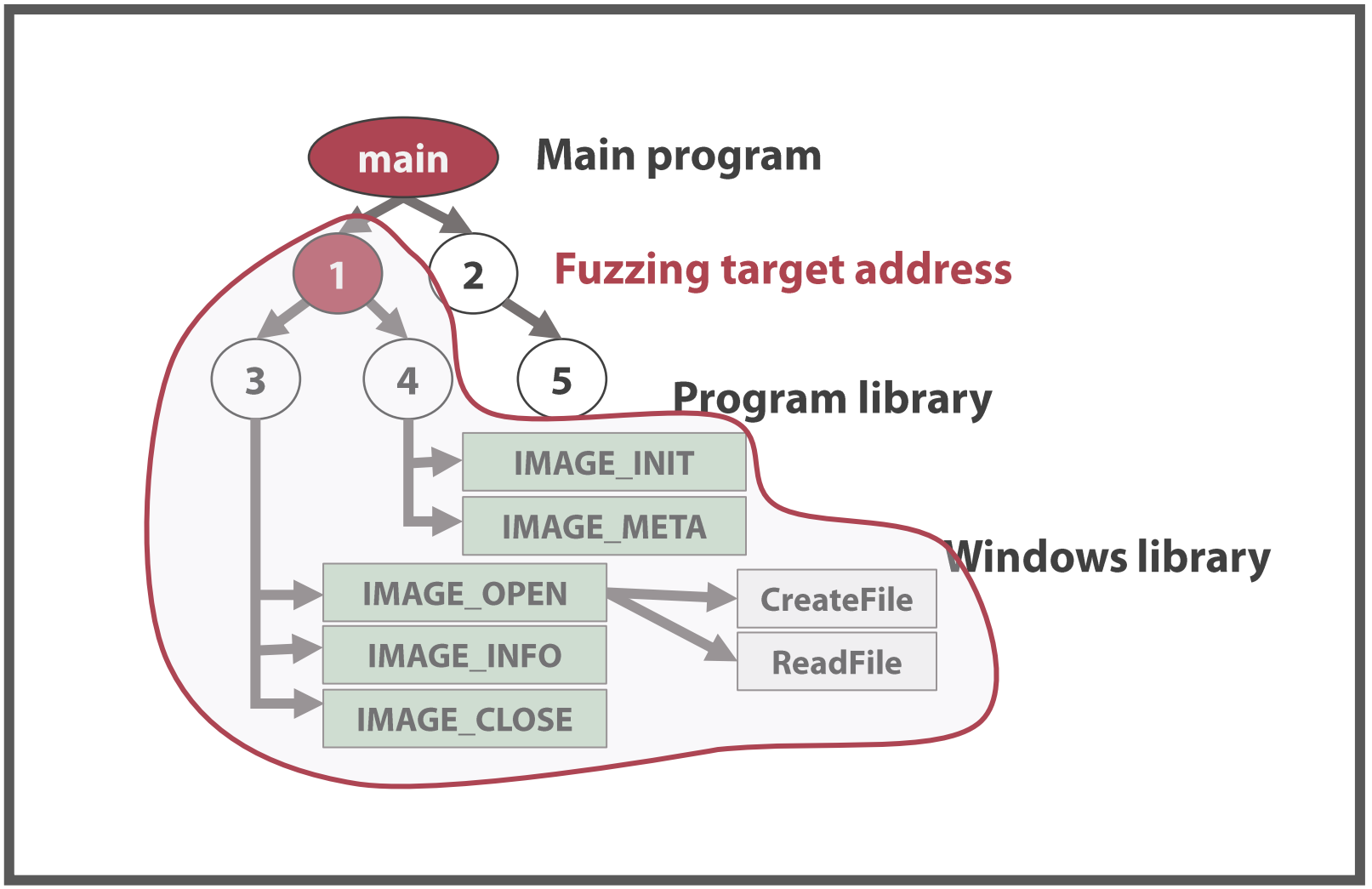
# WINNIE: SEMI-AUTOMATED HARNESS GENERATOR

**Harness generator**

- TRACE COLLECTOR
- **TARGET EXTRACTOR**
- HARNESS BUILDER

**Fuzzer**

- WINDOWS FORK()
- FULLSPEED FUZZING



# WINNIE: SEMI-AUTOMATED HARNESS GENERATOR

**Harness generator**

TRACE COLLECTOR

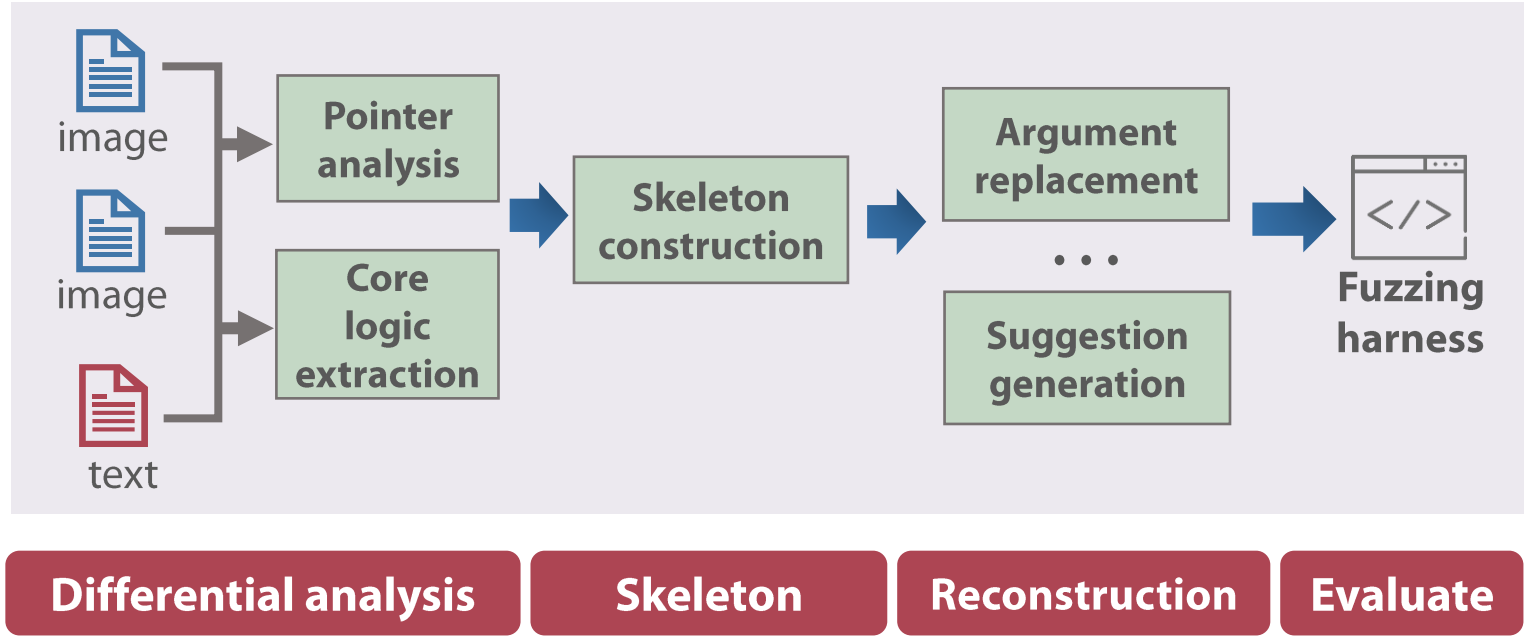
TARGET EXTRACTOR

**HARNESS BUILDER**

Fuzzer

WINDOWS FORK()

FULLSPEED FUZZING



# WINNIE: A PRACTICAL WINDOWS FUZZER

Harness generator

TRACE COLLECTOR

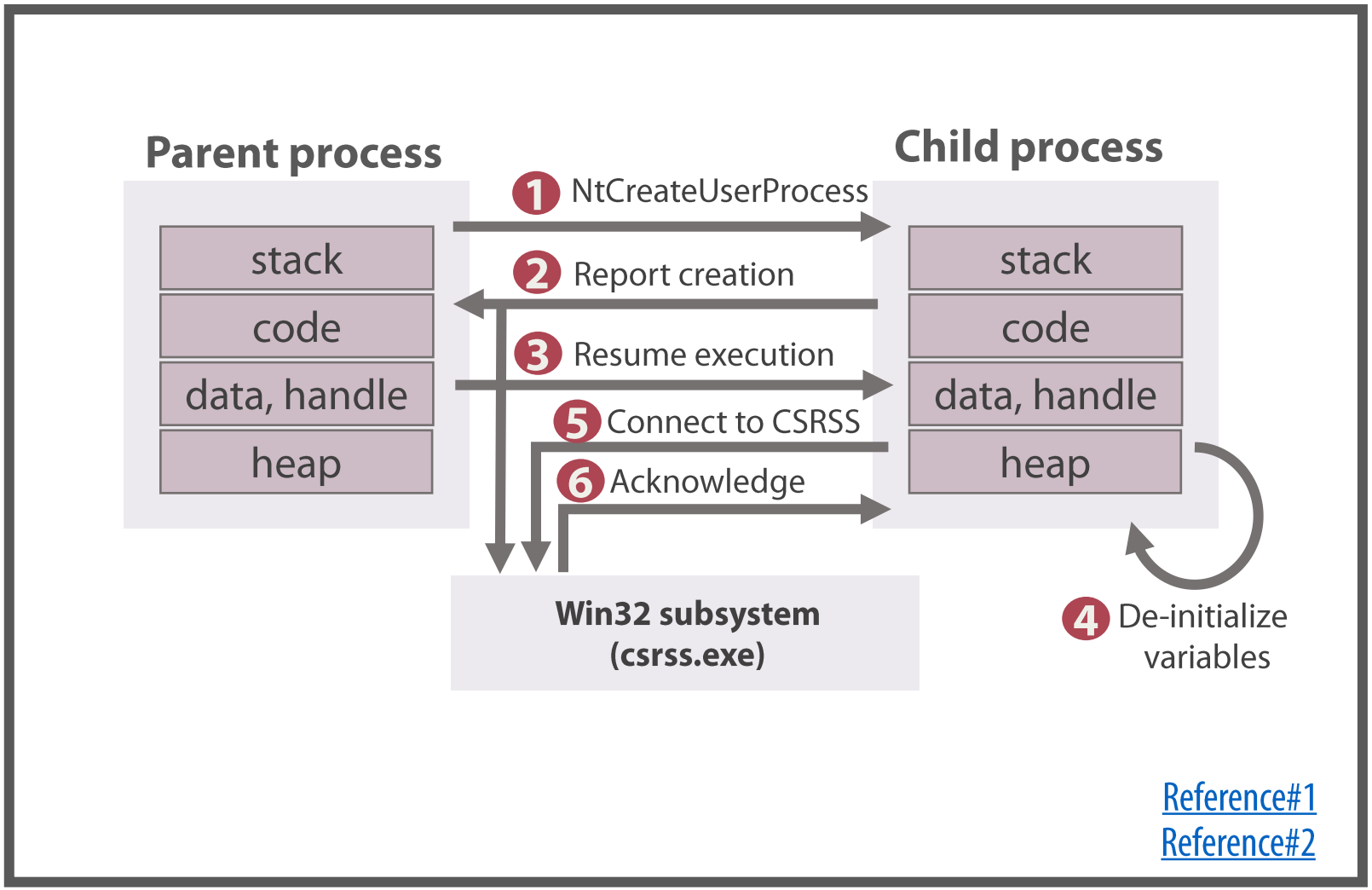
TARGET EXTRACTOR

HARNESS BUILDER

Fuzzer

**WINDOWS FORK()**

FULLSPEED FUZZING




Reference#1
Reference#2

# WINNIE: A PRACTICAL WINDOWS FUZZER

Harness generator

TRACE COLLECTOR

TARGET EXTRACTOR

HARNESS BUILDER

Fuzzer

**WINDOWS FORK()**

FULLSPEED FUZZING


- Reverse-engineered the process creation steps
- fork() *stably* runs complicated initialization *only once*

Georgia Tech

# WINNIE: A PRACTICAL WINDOWS FUZZER

Harness generator

TRACE COLLECTOR

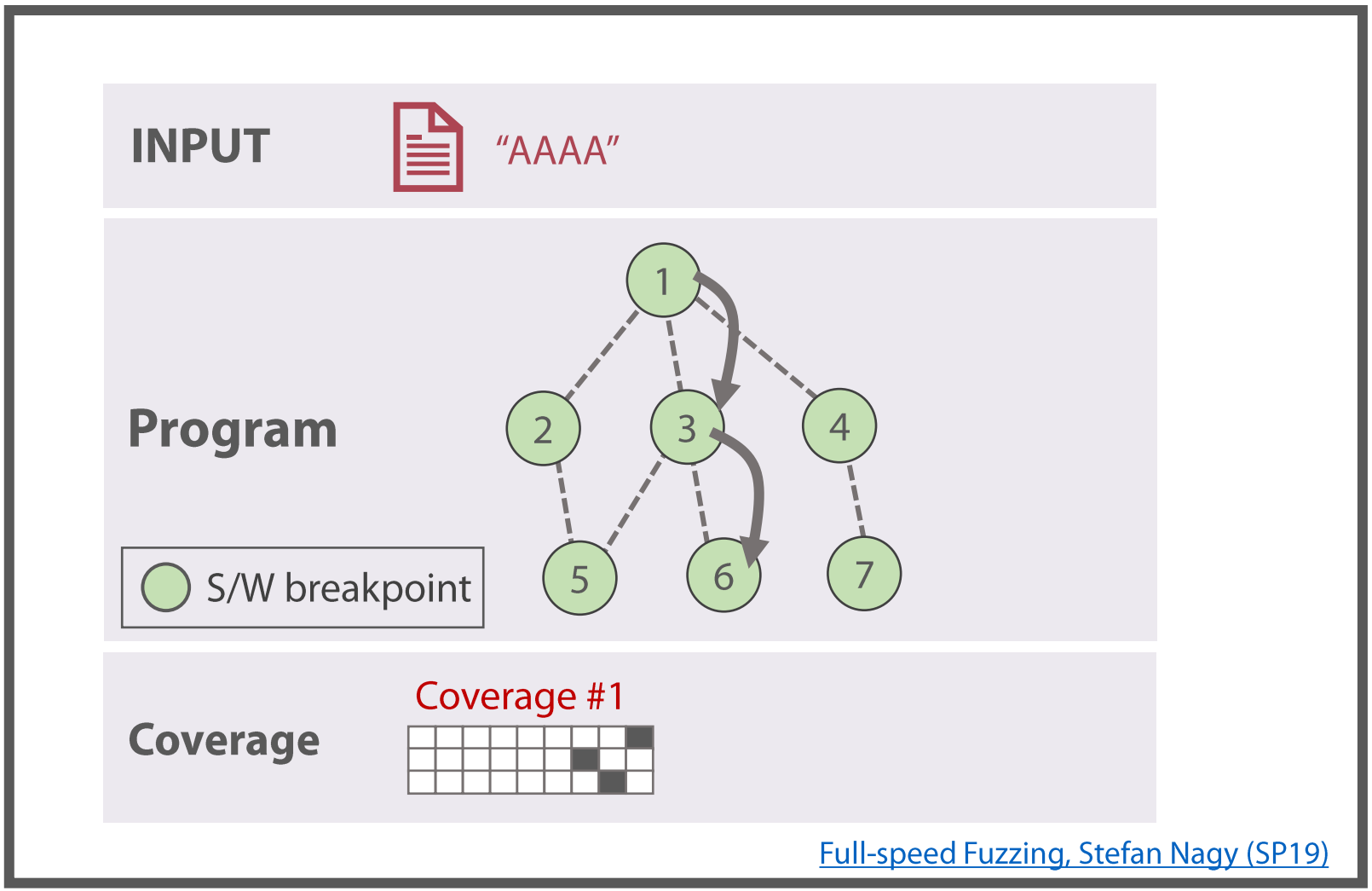
TARGET EXTRACTOR

HARNESS BUILDER

Fuzzer

WINDOWS FORK()

**FULLSPEED FUZZING**




Full-speed Fuzzing, Stefan Nagy (SP19)

# WINNIE: A PRACTICAL WINDOWS FUZZER

Harness generator

TRACE COLLECTOR

TARGET EXTRACTOR

HARNESS BUILDER

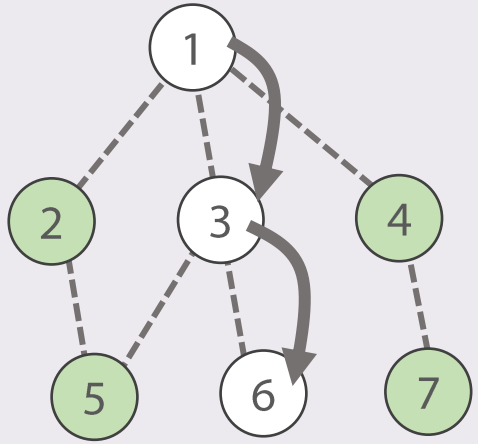
Fuzzer

WINDOWS FORK()

**FULLSPEED FUZZING**

**INPUT**  "AAAA"

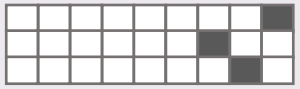
**Program**




S/W breakpoint

**Coverage**

Coverage #1



Full-speed Fuzzing, Stefan Nagy (SP19)

Georgia Tech

# WINNIE: A PRACTICAL WINDOWS FUZZER

Harness generator

TRACE COLLECTOR

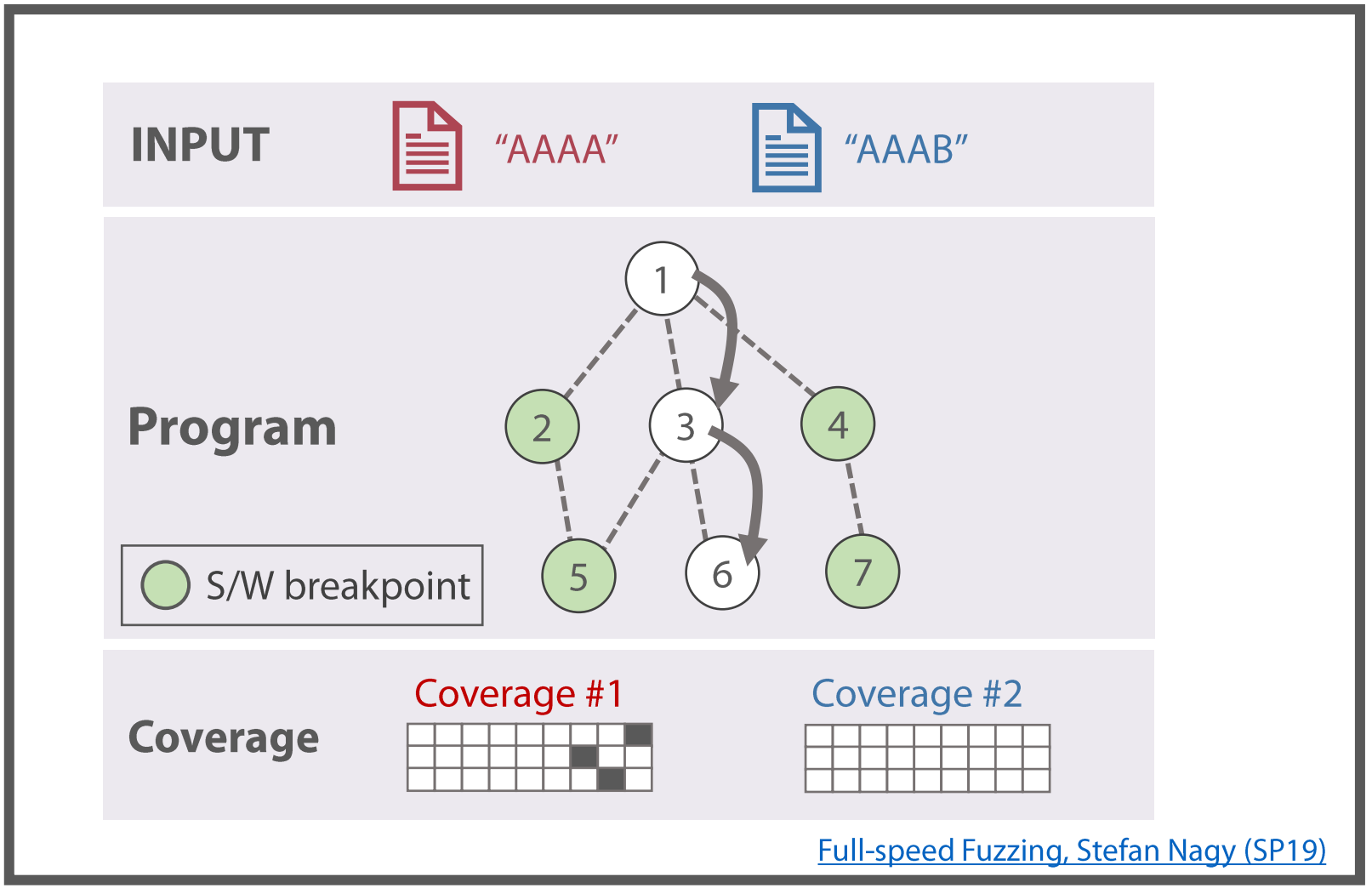
TARGET EXTRACTOR

HARNESS BUILDER

Fuzzer

WINDOWS FORK()

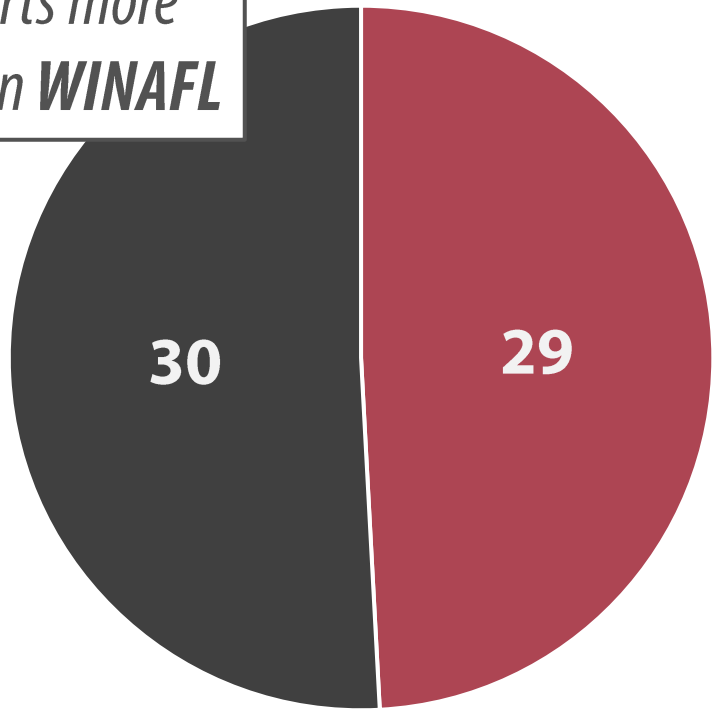
**FULLSPEED FUZZING**




Full-speed Fuzzing, Stefan Nagy (SP19)

# WINNIE SUPPORTS MORE APPLICATIONS

***WINNIE** supports more applications than **WINAFL***




Fuzzable applications
(WINNIE can run all)

WINAFL CAN

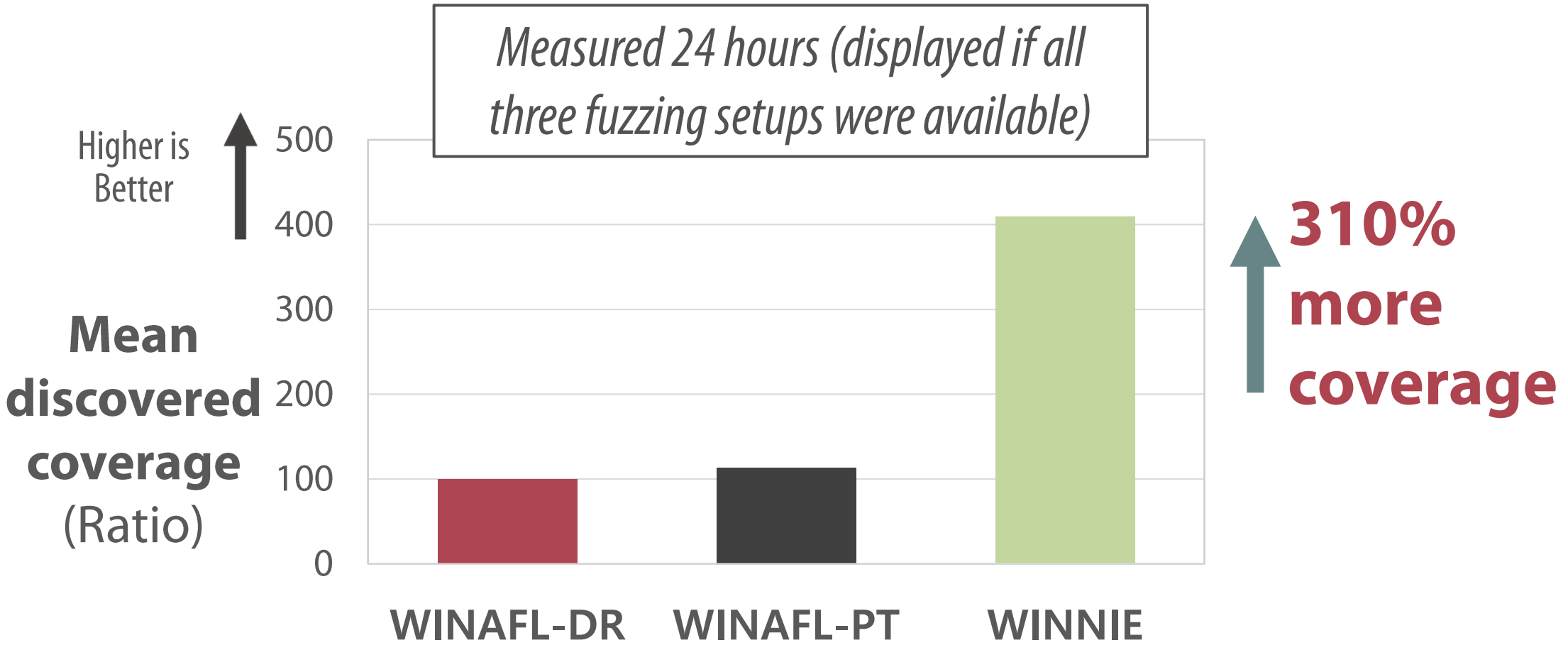
WINAFL CANNOT

**Why?**

- Global state corruption
- Intel-PT driver error

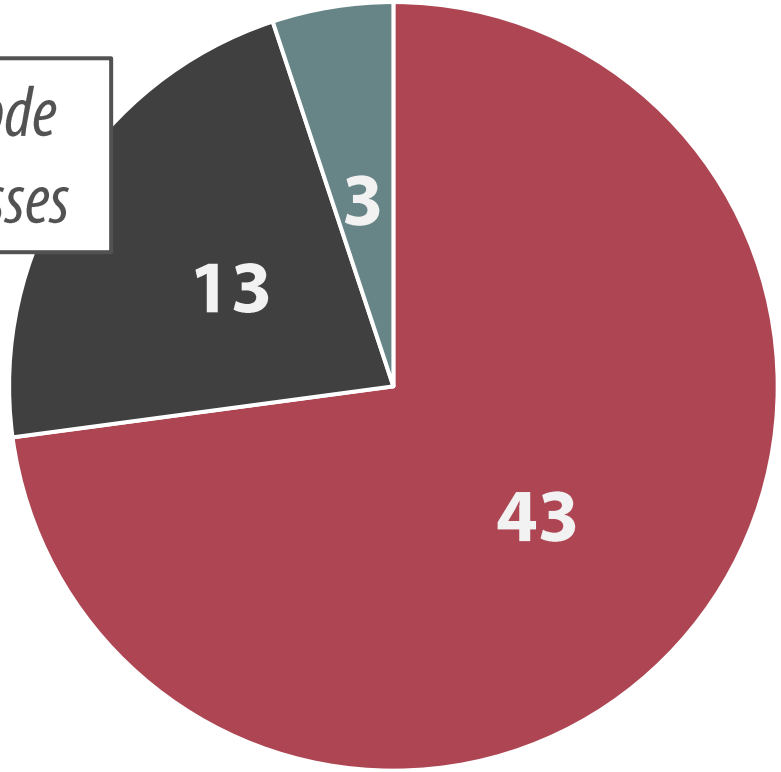
**103% more applications**

# WINNIE HAS BETTER CODE COVERAGE



# WINNIE EFFECTIVELY GENERATES HARNESSES

*Ratio of manually fixed code from the generated harnesses*




- **Fixed Avg. 4% code manually**

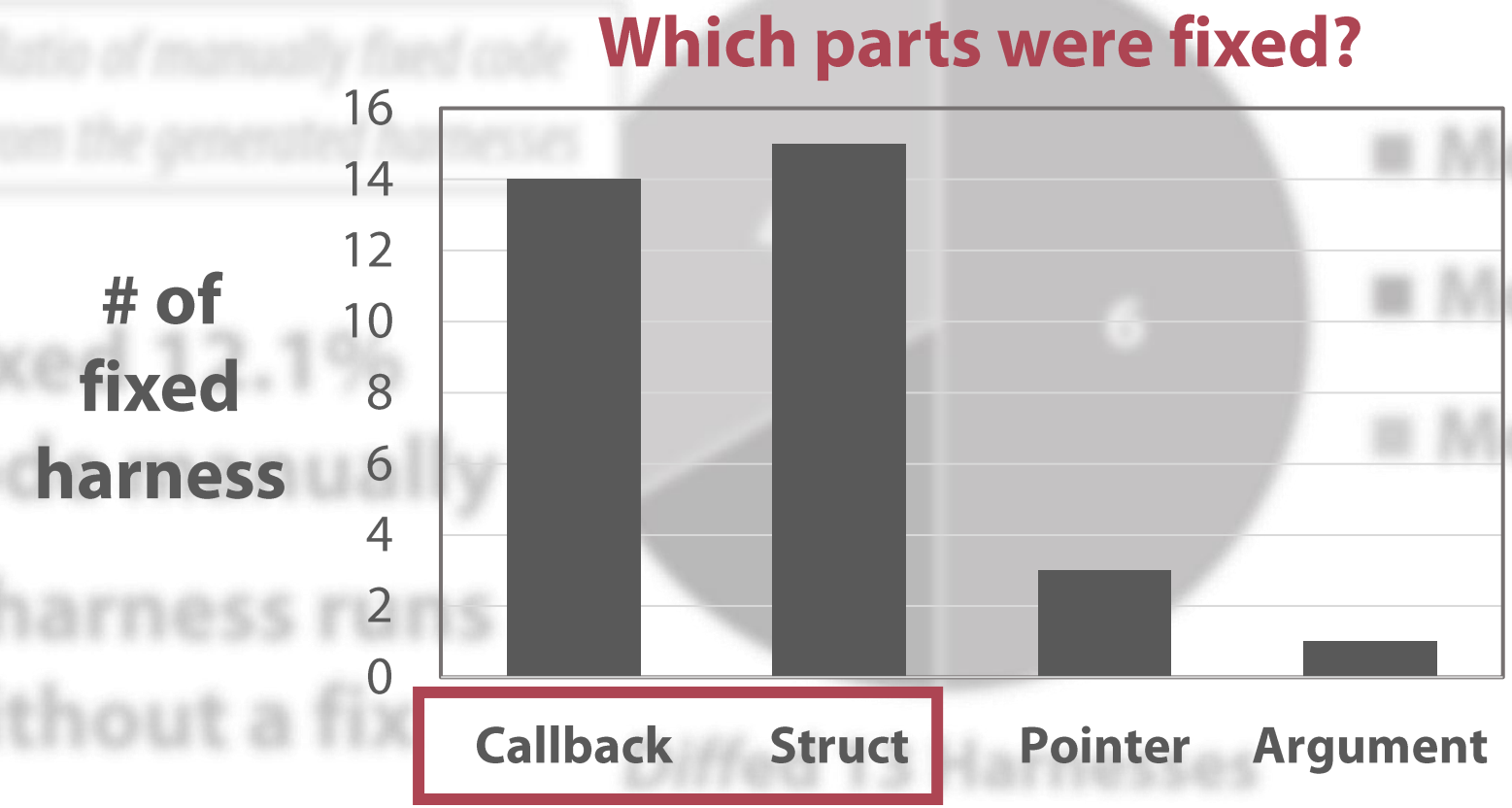
**Modify (0-5%)**

**Modify (5-10%)**

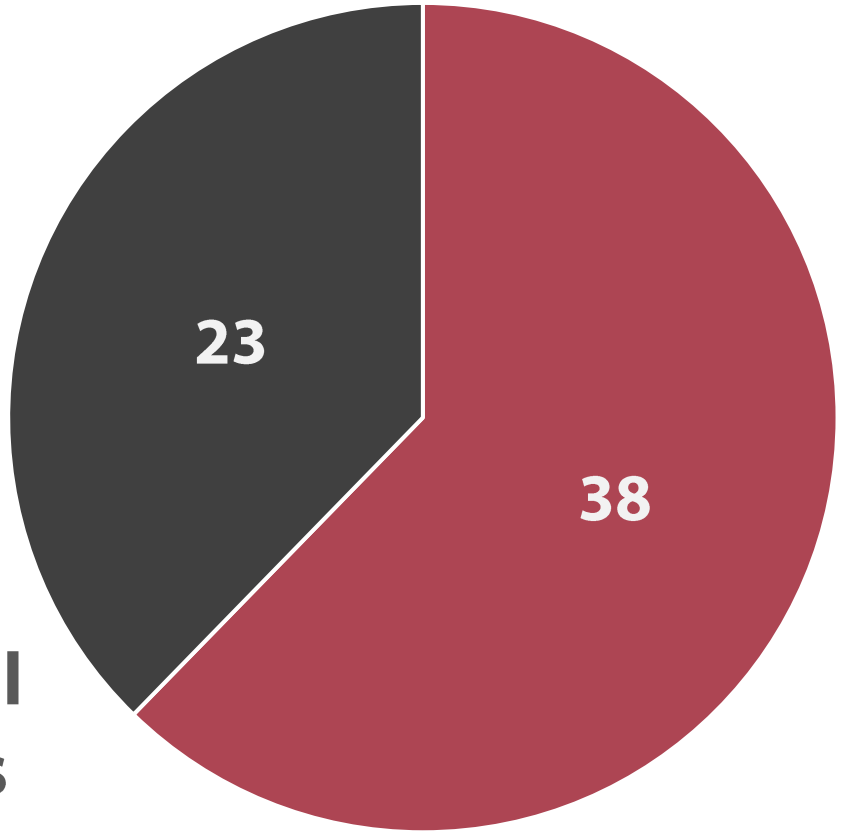
**Modify (10% >)**

**Compared 59 Harnesses**

# WINNIE EFFECTIVELY GENERATES HARNESSES



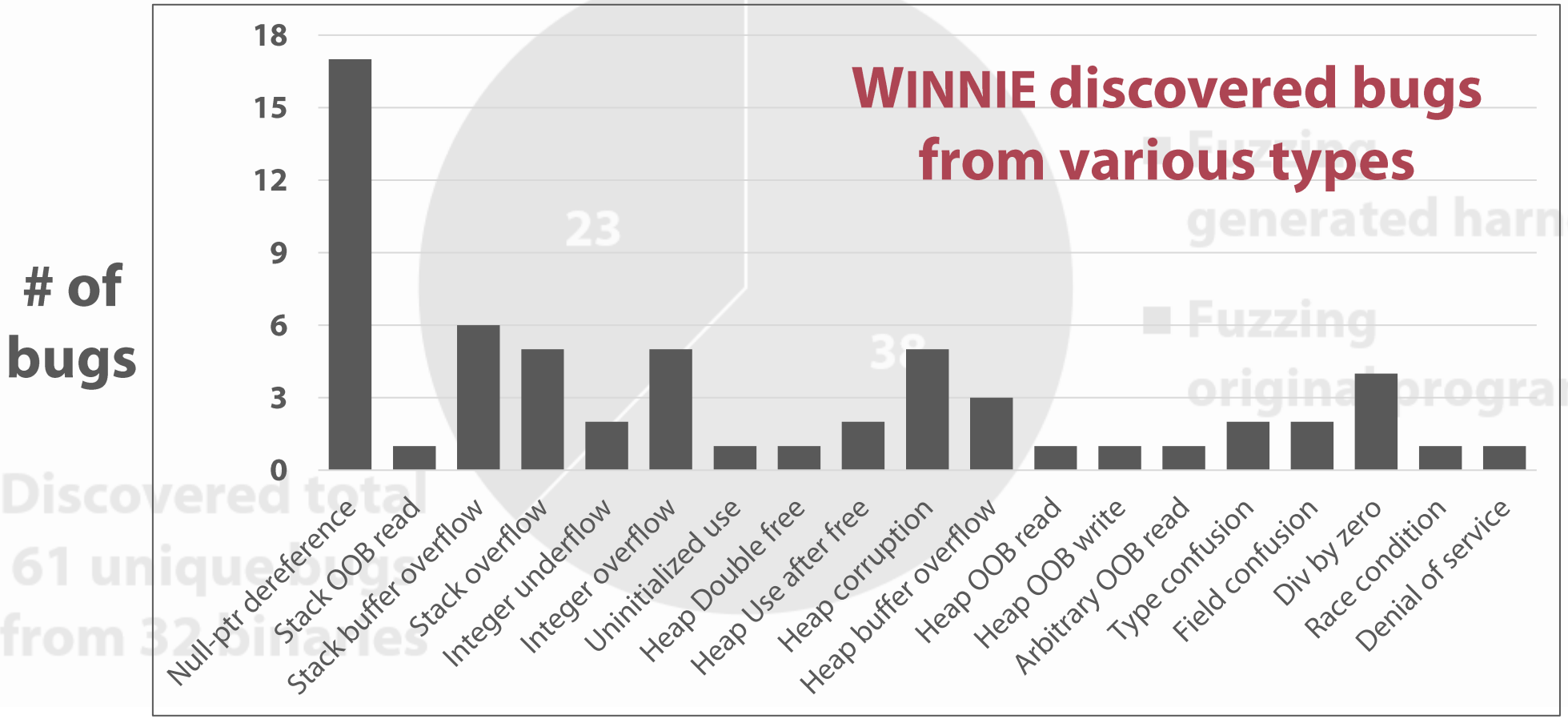
# WINNIE DISCOVERS REAL-WORLD BUGS




- Fuzzing generated harness
- Fuzzing original program

**Discovered total 61 unique bugs from 32 binaries**

# WINNIE DISCOVERS REAL-WORLD BUGS



# CONCLUSION

- WINNIE is a toolchain for fuzzing Windows applications
  - Semi-automated harness generator
  - A practical fuzzer with fast process cloning mechanism

- Open-source: https://github.com/sslab-gatech/winnie

# END

JINHO.JUNG@GATECH.EDU