MProbe: Make the code probing meaningless

Presenter：YongGang Li

Authors：YongGang Li, Ye-ching Chung, JinBiao Xing, Yu Bao, GuoYuan Lin

Affiliation: The China University of Mining and Technology

CONTENTS

01 How does code probing works?

02 How does MProbe defend against code probing?

03 What is the effect and efficiency of MProbe?

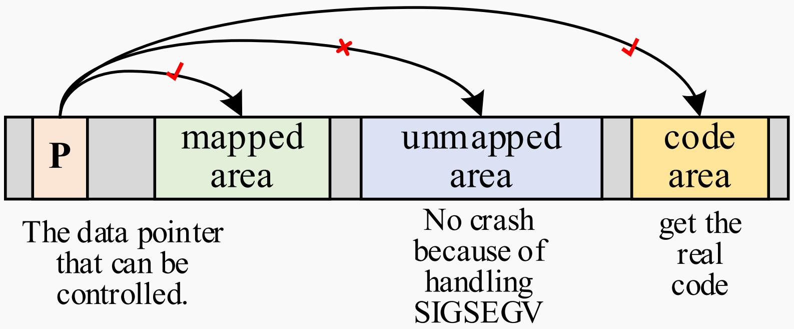
04 Conclusion

01

PART

How does code probing works?

How does code probing works?

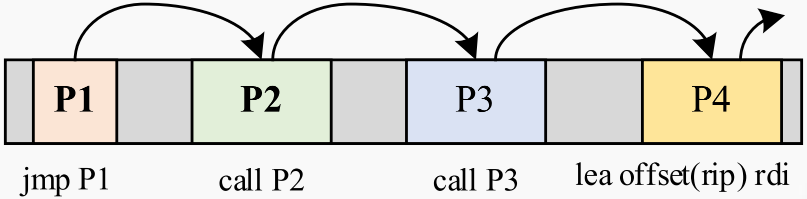

P
mapped area
unmapped area
code area
✓
✗
✓
The data pointer that can be controlled.
No crash because of handling SIGSEGV
get the real code

- ➢ Vector 1: arbitrary read.

  Variants of heart bleed

  JIT-ROP

Read code recursively



- ➢ Vector 2: arbitrary jump.

  BROP—Blind ROP

- ➢ Vector 3: Side-channel Probing.

  Analyze the TLB hit and miss.

- ➢ Vector 4: Data leakage.

  DOP—read the PLT and GOT where store the code addresses

02

PART

How does MProbe defend against code probing?

How does MProbe works?

The main idea of MProbe is to make the probed code lose its execution permission in the original address space.

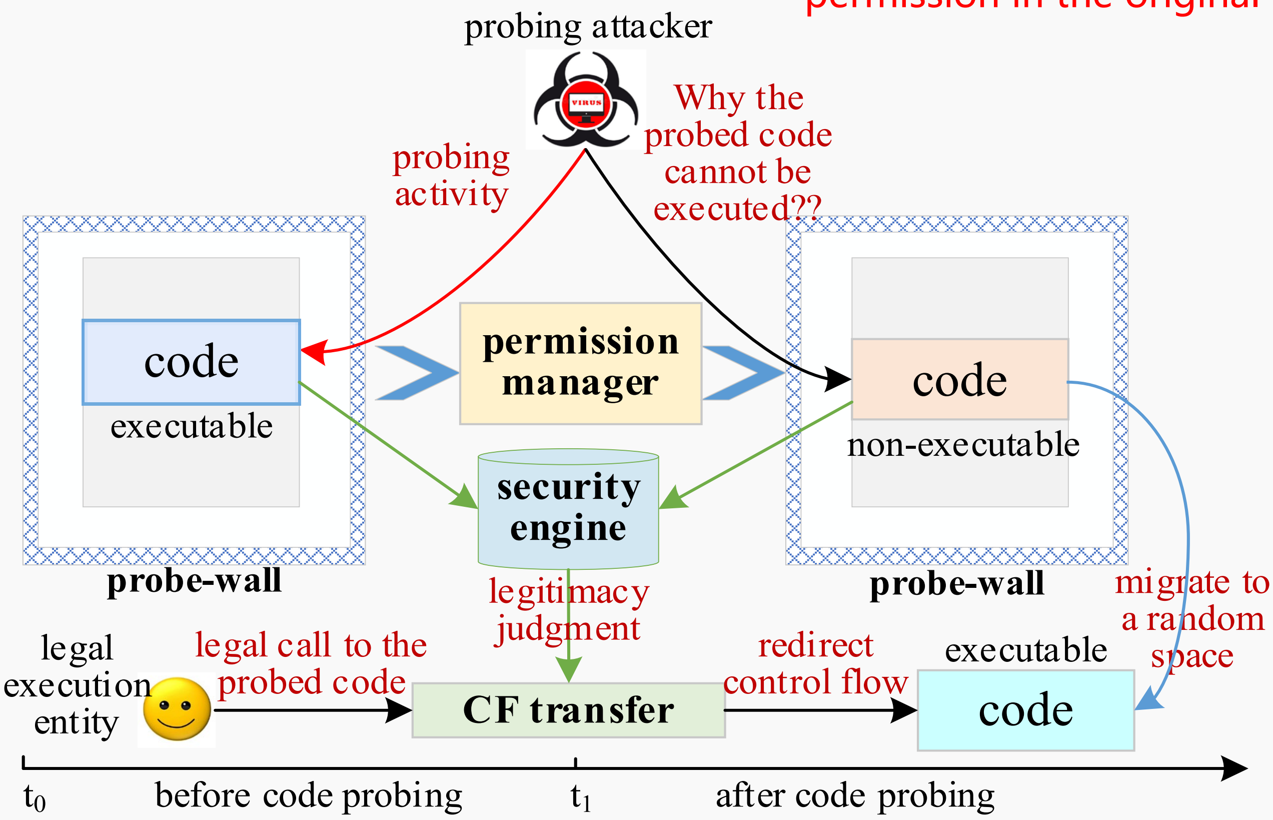
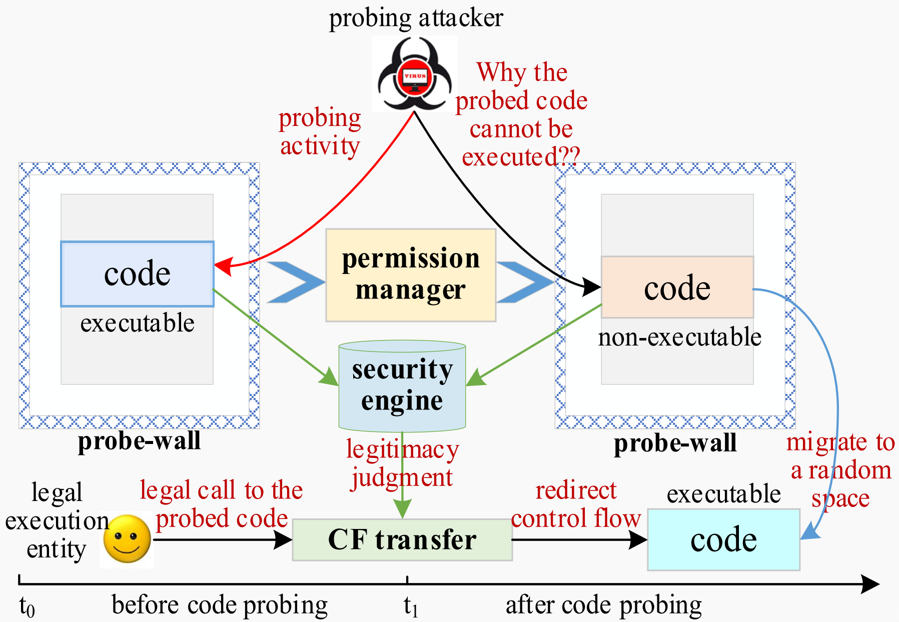


Figure 1: The overall architecture of MProbe

How does MProbe works?

- probe-wall
  - Detect the probing activities.
- permission manager
  - Disable the execution permission of the probed code.
  - Migrate the probed code to a random address space
- security engine
  - Judge the legitimacy of the control flow
- CF transfer
  - Redirect the control flow to the random space



1. perceive the code probing activities
2. prevent the probed code snippets from being used as gadgets
3. ensure the probed code to be called legally

How does MProbe works?

- Perceive the probing attacks

The buddy system is modified to create a memory pool, which is the source of code page allocation. Pages in this pool are pre-set as unreadable. **Perceive Vector 1**.

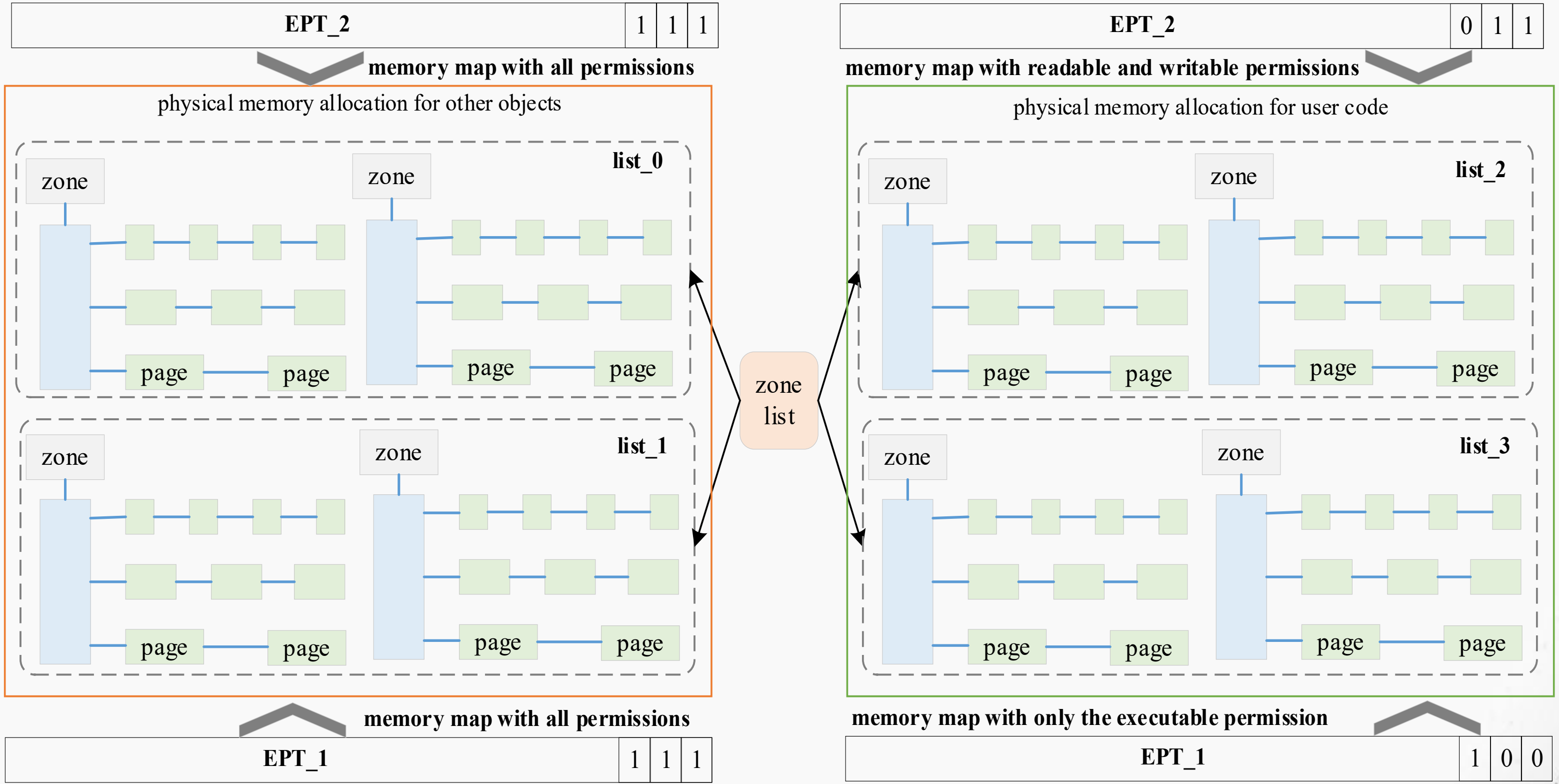

EPT_2
1 1 1
memory map with all permissions
physical memory allocation for other objects
list_0
zone
zone
page
page
page
page
list_1
zone
zone
page
page
page
page
memory map with all permissions
EPT_1
1 1 1
zone
list
EPT_2
0 1 1
memory map with readable and writable permissions
physical memory allocation for user code
list_2
zone
zone
page
page
page
page
list_3
zone
zone
page
page
page
page
memory map with only the executable permission
EPT_1
1 0 0

Figure 2. The overall design of user code memory allocation

How does MProbe works?

◆ Perceive the probing attacks

- **Perceive Vector 2 (arbitrary jump)**

  1. Capture SIGSEGV and SIGILL
  2. Record the code triggers SIGSEGV and SIGILL
  3. Identify the restated process.

- **Perceive Vector 3 (side-channel probing)**

  1. Map the space at V+n*xGB (n<8, x=1 or 512)
  2. Set the page tables of the space to be unreadable

How does MProbe works?

- Perceive the probing attacks

- **Perceive Vector 4 (data leakage)**

  1. Set GOT to be unwritable
  2. Store the library function address in a non-readable code snippet
  3. Fill the code snippet’s starting into GOT

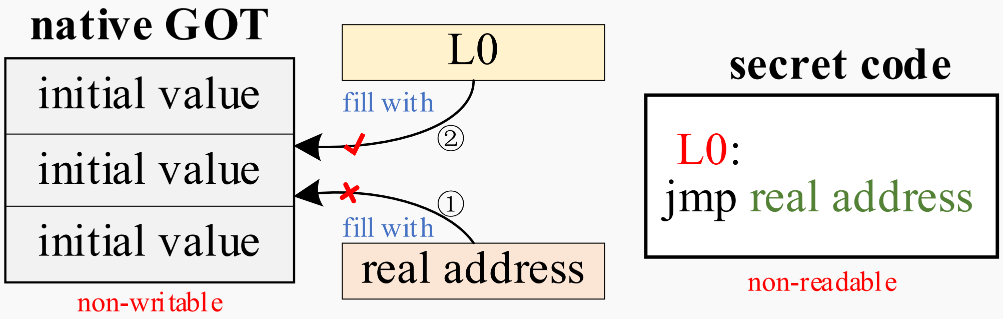


Figure 3. Hide the function address in GOT

How does MProbe works?

◆ Prevent the probed code is used as a gadget

- **Vector 1 (arbitrary read)**

1. When the Vector 1 is perceived, MProbe directly prevent the current code reading.

- **Vector 4 (side-channel probing)**

1. When the Vector 2 is perceived, MProbe directly prevent the current memory access.

How does MProbe works?

- Prevent the probed code is used as a gadget

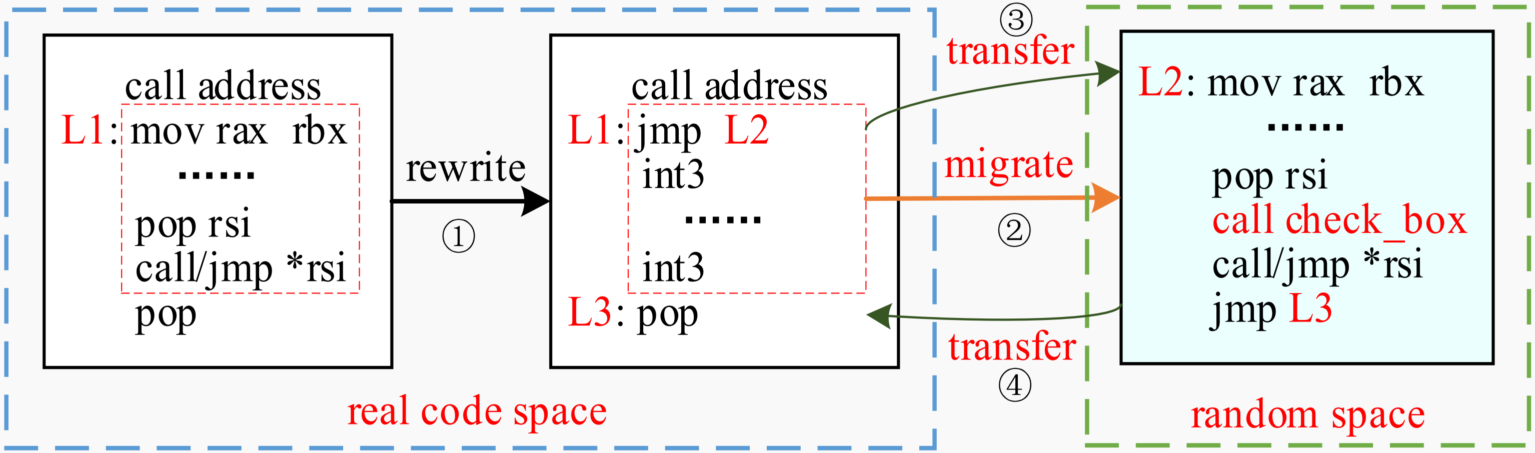


Figure 4. Migrate a code block to a random space

For the Vector 2 or Vector 3, the probed code block containing an ICT instruction will be migrated to a new random space

How does MProbe works?

◆ Prevent the probed code is used as a gadget

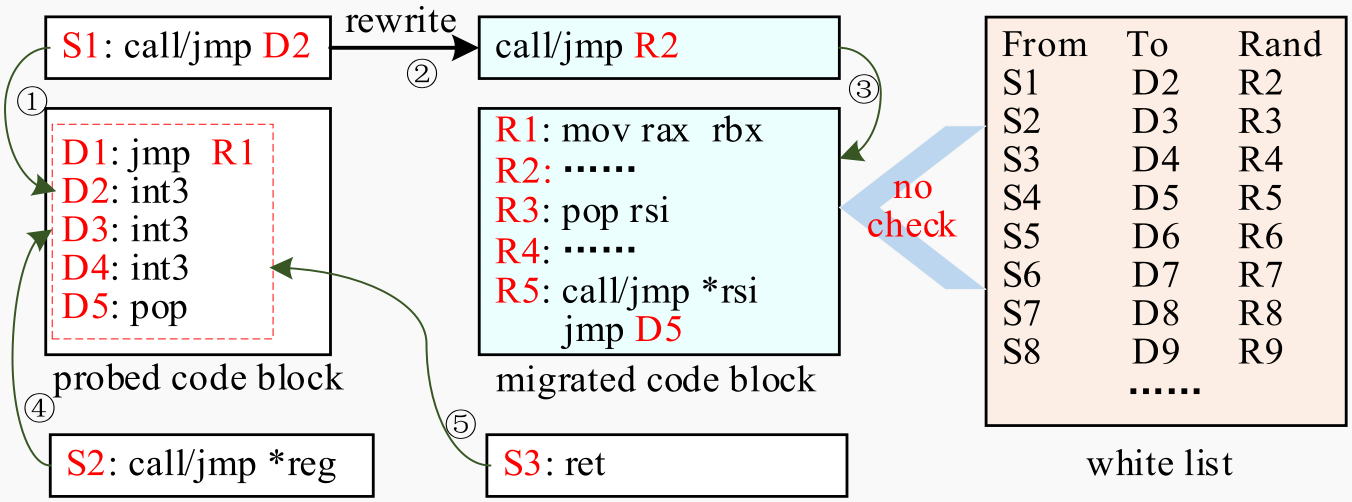
- **Security strategies:**

1. In the same mapped space, *jmp ** jumps to the inside of the current function; *call ** can only jump to the head of other functions.
2. If without going through PLT, *call* and *jmp* cannot transfer the control flow to a library from application code, nor can transfer it to any other libraries from the current library.
3. The jump targets of ICT instructions must conform to the code alignment forms in the ELF file.
4. The return address of the instruction ret cannot be changed before ret is executed.

……

How does MProbe works?

◆ Transfer the legal control flow.



```
S1: call/jmp D2
```

```
call/jmp R2
```

```
D1: jmp  R1
D2: int3
D3: int3
D4: int3
D5: pop
```

```
R1: mov rax  rbx
R2: ......
R3: pop rsi
R4: ......
R5: call/jmp *rsi
    jmp D5
```

```
S2: call/jmp *reg
```

```
S3: ret
```

From	To	Rand
S1	D2	R2
S2	D3	R3
S3	D4	R4
S4	D5	R5
S5	D6	R6
S6	D7	R7
S7	D8	R8
S8	D9	R9
	......	

Figure 5. Transfer the legal control flow

1. Rewrite the probed code block
2. The control flow transferred to the rewritten code in the real code space triggers a system trap
3. Check its legitimacy

03

PART

What is the effect and efficiency of MProbe?

What is the effect and efficiency of MProbe?

◆ Security

- **Arbitrary read:**

1. The modified HeartBleed can be detected due to the captured SIGSEGV.
2. The code reading of *memcpy(bp,pl,payload)* in openssl can be detected due to the unreadable code segment.

- **Arbitrary jump:**

1. Blind ROP can be detected due to the captured SIGSEGV or SIGILL.

What is the effect and efficiency of MProbe?

◆ Security

- **side-channel probing:**

1. The attacker cannot obtain the 30th~32nd bits and 39th~41st bits of the virtual address.
2. The typical side-channel attacks such as flush+reload, EVICT+TIME and PRIME+PROBE can also be detected and blocked by MProbe whenever they read the code.

- **GOT leakage:**

1. What the attackers obtain are not the real addresses of library functions.

What is the effect and efficiency of MProbe?

◆ Security

- **Control flow detection**

binary code	size	total gadgets	gadget chains	defense
libcodeblocks.so	4267	535758	70	√
libcapstone.so	869	109538	3	√
libfam.so	15	1969	1	√
libnetpbm.so.10	60	7704	1	√
libwxsmithlib.so	1719	187992	48	√
400.perlbench	877	100750	5	√
401.bzip2	45	3942	1	√
403.gcc	2285	254156	29	√
429.mcf	8	1079	1	√
471.omnetpp	401	56954	2	√

What is the effect and efficiency of MProbe?

◆ performance

- **SpecCPU 2006**

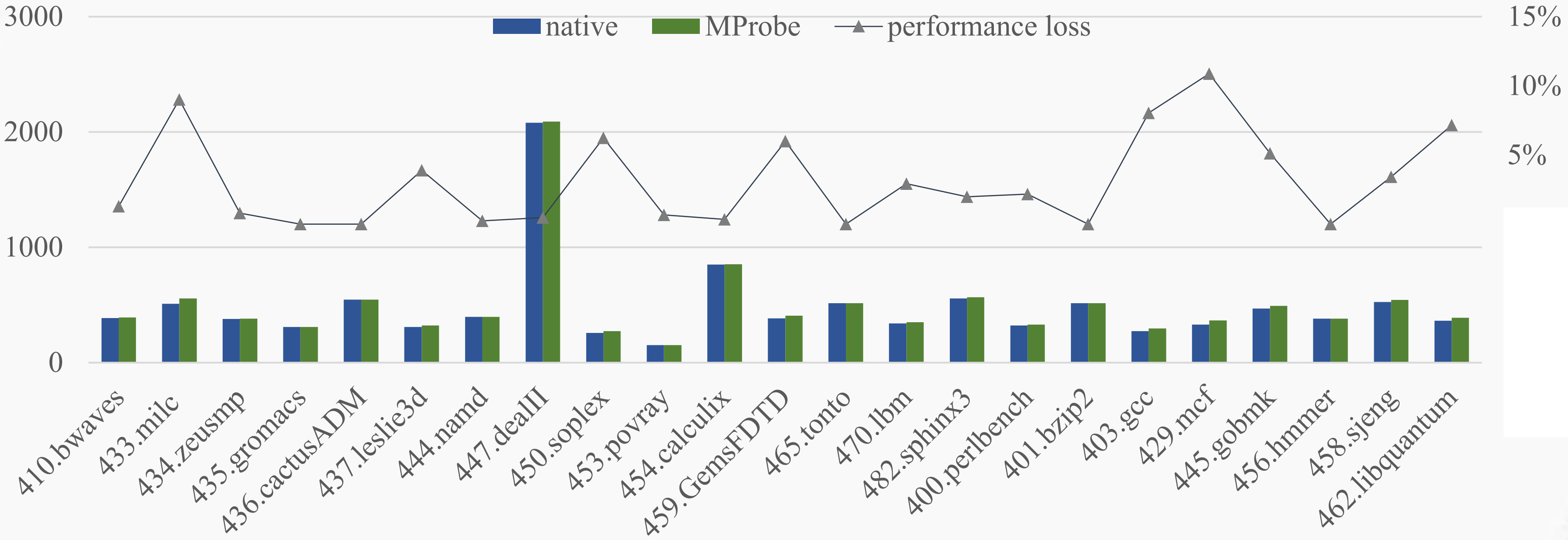


Figure 6. SpecCPU2006 test results.

What is the effect and efficiency of MProbe?

◆ performance

- **Lmbench**

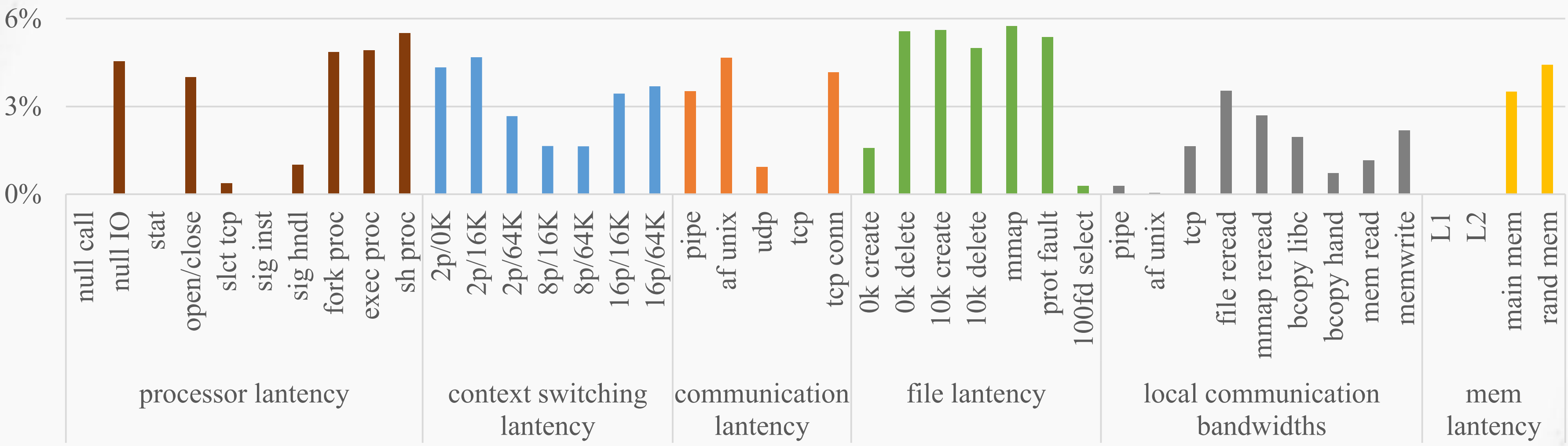


Figure 7. Lmbench test results.

What is the effect and efficiency of MProbe?

◆ performance

- **Probing test**

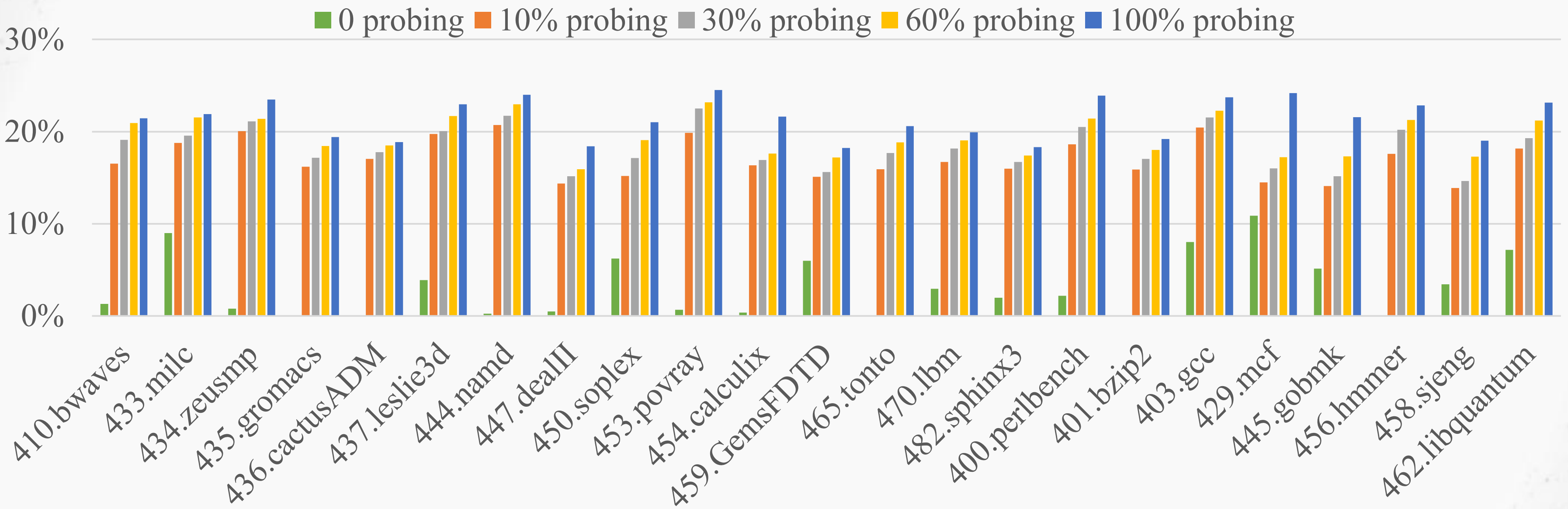


Figure 8. MProbe's impact on the process after a probing attack.

04

PART

Conclusion

What is the effect and efficiency of MProbe?

◆ Conclusion

(1) Propose a probing perception mechanism.

(2) Propose a protection mechanism to prevent the probed code from being used as gadgets.

(3) Implement the MProbe prototype in Linux.

THANKS