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
How does code probing work?
How does code probing works?

- Vector 1: arbitrary read.
  - Variants of heart bleed
  - JIT-ROP

- 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
How does MProbe defend against code probing?
How does MProbe work?

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

How does MProbe works?

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.**

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 work?

- 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

![Diagram: Hide the function address in GOT](image)

**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

![Diagram showing the process of MProbe](image)

**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.

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
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**

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

![Lmbench test results.](image)

**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.
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