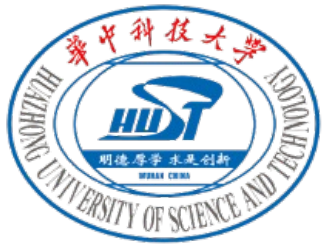


华中科技大学
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

网络空间安全学院

内存安全漏洞/内存错误漏洞

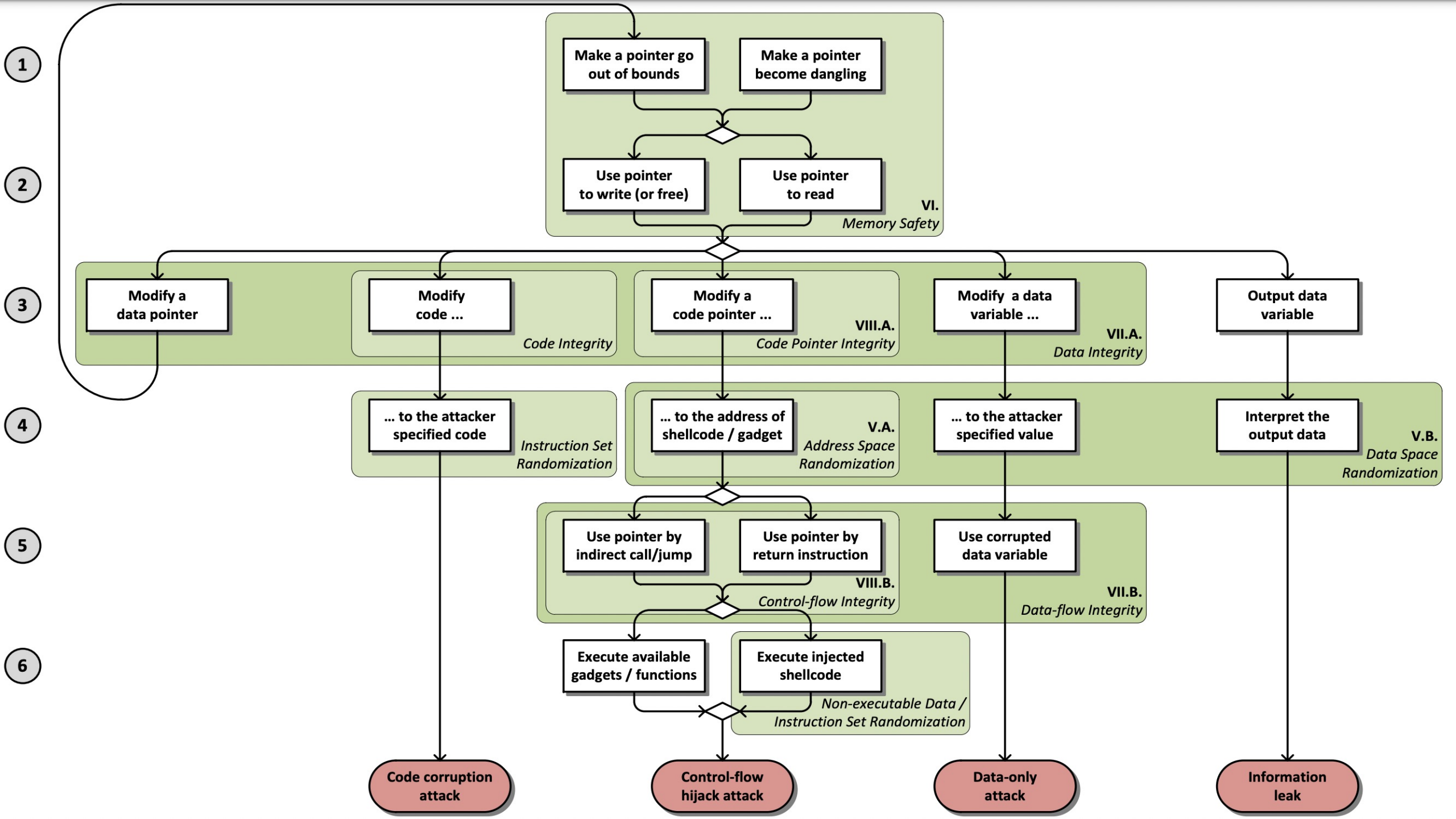
SoK: Eternal War

László Szekeres†, Mathias Paye
†Stony Brook
‡University of Cali
*Peking U

SoK: Eternal War in Memory

内存安全漏洞/内存错误漏洞

- 内存安全漏洞/内存错误漏洞
  - 内存破坏漏洞：更突出内存内容的破坏或篡改
  - C/C++ 等非内存安全语言漏洞
  - 攻击技术
    - Shellcode
    - Return-to-libc
    - ROP
  - 防御技术
    - Stack Canary
    - Data Execution Prevention
    - Address Space Layout Randomization


1
2
3
4
5
6
Make a pointer go out of bounds
Make a pointer become dangling
Use pointer to write (or free)
Use pointer to read
VI.
Memory Safety
Modify a data pointer
Modify code ...
Code Integrity
Modify a code pointer ...
VIII.A.
Code Pointer Integrity
Modify a data variable ...
VII.A.
Data Integrity
Output data variable
... to the attacker specified code
Instruction Set Randomization
... to the address of shellcode / gadget
V.A.
Address Space Randomization
... to the attacker specified value
Interpret the output data
V.B.
Data Space Randomization
Use pointer by indirect call/jump
Use pointer by return instruction
VIII.B.
Control-flow Integrity
Use corrupted data variable
VII.B.
Data-flow Integrity
Execute available gadgets / functions
Execute injected shellcode
Non-executable Data / Instruction Set Randomization
Code corruption attack
Control-flow hijack attack
Data-only attack
Information leak

内存安全漏洞分类

- 越界写/读 (Out-of-Bound Write/Read)
- 缓冲区溢出 (Buffer Overflow)
  - 栈、堆、全局数据缓冲区溢出
- 整数溢出 (Integer Overflow or Wraparound)
- 释放后使用 (Use After Free)
- 失控的资源消耗 (Resource Consumption)
- 空指针 (Null Pointer Dereference)
- 未初始化变量 Uninitialized Variable
- ……

空间类内存安全漏洞

- 越界写/读 (Out-of-Bound Write/Read)
- 缓冲区溢出 (Buffer Overflow)
  - 基于栈的缓冲区溢出 Stack-based buffer overflow
  - 基于堆的缓冲区溢出 Heap-based buffer overflow
  - 基于全局的缓冲区溢出 Global-based buffer overflow
- 整数溢出 (Integer Overflow)
- 格式化字符串 (Format String)
- 空指针 (Null Pointer Dereference)
- 未初始化变量 (Uninitialized Variable)

时间类内存安全漏洞

- Use-After-Free (UAF) 释放后重复
- Double-Free 双重释放
- Invalid Free 无效释放