esse

Advanced Security for Systems Engineering – VO 01: Advanced Attacks on Applications 1

Clemens Hlauschek, Christian Schanes

Capture-the-Flag Team defragmented.brains

- Take part in many international hacking competitions
- Diverse bunch, different skills and skill levels
- Join our mailinglist: `ctf-join@inso.tuwien.ac.at`
- Next CTF: Hack.lu (Fluxfinger/Bochum) 28.-30.10.

Memory Corruption Bugs: Basics

Memory Corruption Bugs

After decades of research, problem still prevalent.

- Morris Worm (11/2/1988) exploited a buffer overrun in `fingerd`
- 33 years, and thousands of research papers later: memory corruptions bugs still relevant
- 2015: More than 1 billion devices affected by Stagefright vulnerability
- 2017: WanaCry ransomware infects 200.000 computers within a day via EternalBlue (SMB heap exploit, stockpiled by the NSA, leaked by The Shadow Brokers)
- Hard to exploit due to mitigation techniques nowadays
- Exploits traded for serious money on the black market

Memory Corruption Bugs: Vulnerability Categories

- Stack overflow
- Heap overflow
- Double free
- Use after free
- Buffer underrun
- Format string vulnerability
- Integer overflow
- Signed/unsigned conversion error
- Type confusion error
- NULL pointer dereference

Memory Corruption Bugs: Main Culprits

- **Computer and OS Architecture**

  Memory space contains control flow information as well as user data

  - Function return address stored on stack
  - Heap management info interleaved with user data
  - Function pointers: exception handler pointer, virtual function table, etc.

- **Type-unsafe Programming Languages**

  C or C++ allow arbitrary writes in process' memory space

  - Programmer responsible for validating user input
  - Some programming errors difficult to catch

Memory Corruption Bugs: Results of successful exploits

- Denial of Service
  - Induce process crash, prevent clients from accessing service
- Information Disclosure
  - Leaking private information (e.g., passwords, private keys)
  - Often 1st step in circumventing mitigation techniques (e.g., leaking process space address information)
- Control Flow Hijacking
  - Maliciously alter the process' behaviour: "Arbitrary Code Execution"

Memory Corruption Bugs: Control Flow Hijacking

1. Modify control flow data / metadata with user input
   - Function return address
   - Function pointer
   - Virtual method table
   - Heap metadata
   - Global Offset Table (GOT) or Import Address Table (IAT)
2. Redirect Control Flow
   - to injected (machine) code
   - or to existing code in the process' memory space

Literature / Links

- Aleph One (1996): Smashing the Stack for Fun and Profit, Phrack 49.
- Bulba and Kil3r (2000): Bypassing Stackguard and Stackshield, Phrack 56.
- Richarte (2002): Four different tricks to bypass Stackshield and Stackguard protection.
- Corelan Team: Exploit writing tutorial part 6 : Bypassing Stack Cookies, SafeSeh, SEHOP, HW DEP and ASLR.
- Meer (2010): Memory Corruption Attacks. The Almost Complete History. BlackHat.

Literature / Links

- Veen (2012): Memory Errors: The Past, the Present, and the Future.
- Szekers (2013): SoK: Eternal War in Memory.
- Llorente-Vazquez (2022): The Neverending Story: Memory Corruption 30 Years Later.
- Chris Anley (2007), The Shellcoder's Handbook: Discovering and Exploiting Security Holes.
- Intel 64 and IA-32 Software Developers Manual Combined Volumes.
- Metasploit Framework, www.metasploit.com

Thank you!

https://security.inso.tuwien.ac.at/