Security for Hackers and Developers: Exploit Development

AUDITING, DEBUGGING, AND VULNERABILITIES

Dr. Jared DeMott
CHIEF HACKING OFFICER

@jareddemott www.vdalabs.com

Overview

Why Exploit?

Audit Binary

Audit Code

Debug

What Could be Hacked?

Computers

- Desktops, laptops, servers
- Mainframes, Minicomputers

Mobile

- Phones, tablets

Network Security

- SOHO too

Industrial Control Systems

- Buildings, factories, machines

Embedded systems of all kinds

- IoT, cameras

So What?

Privilege

- If the exploited process is remote to the attacker, or is locally running at a higher access privilege, this is interesting because the attacker is crossing a “privilege boundary”
- Perform unauthorized actions
  - Ransomware, theft, fraud, blackmail

Money for the Bug

- Odays have real monetary value
- White, grey, black markets

Agenda

- Defacement, DDoS, politics, fake news

Activates Relating to Exploit Development

Vulnerability Scanning

- Often involves running Nessus, etc.

Software Testing

- Software companies should test their software for robustness and security
  - Often lack hacking expertise

Penetration Test

- Often involves using exploits from a tools like Metasploit or Canvas

Methods of Exploitation

Supply Chain

- Embedded backdoors
  - Difficult to defend against

Social Engineering

- Often combined with technical attack
- Sometimes combined with a physical breach as well
- Employee's need to be training to defend against

Unknown Software Vulnerabilities

- Often called CNE

Specialization

People and Companies Tend to Specialize

- Forensic companies, pentest, software testers, blue team setup, web auditing, code review, etc.

Oday Hunting

- Often No/limited source code
  - Except with open source
- Usually involves reverse engineering and fuzzing
  - Exploit development if a critical bug is discovered

Assume we've found a bug, what do we know need next?

Bug → Proof of Concept (PoC) → Exploit

What knowledge is required to do this?

1. Operating System
   - Win/Linux most common
2. Architecture
   - x86 is most common

What do we do next?

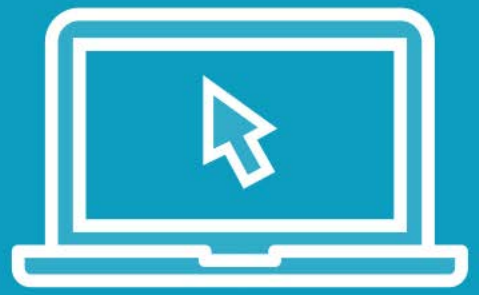
3. Software development
   - In order to RE/understand the bug, one must have a clue as to how it was written/compiled
4. Reverse Engineering (RE) techniques/tools
5. Exploit development tools and techniques

Tools and Techniques

Creative Process

- But there are common classes of bugs
  - Similar packaging/development tools can apply
- Debuggers are key
  - Gdb for *NIX, Immdbg/WinDbg/IDA for Windows
- Network programming and shellcoding often required
  - Shellcode is the attack code (payload) you wish to have execute via the software hole

Demo



Audit Code

- Identify the location and reason for the bug
  - What's the bug?
  - What's interesting about different inputs?
  - How do debug?
  - What's interesting about different builds?

Lab 1

- Practice, using same binaries
- Hone IDA and WinDbg skills

Summary

Importance of Safe Code

- Hackers will find implementation flaws
  - Runtime or manual audits
- They will find a way to exploit them
  - Architecture knowledge, Debugger, Shellcode
- Next:
  - Function pointer overwrite