PV204 Security technologies

Reverse engineering of binary applications

Petr Švenda svenda@fi.muni.cz
Faculty of Informatics, Masaryk University

CS
Centre for Research on
Cryptography and Security

Laboratory

1. Go to disassembly in IDE, understand basic principles
2. Practical disassembling and binary debugging tutorial
   - Lena tutorial 1 & 2
   - Open file in debugger
   - Basic operations, jumps
   - Patching
   - Understanding structures from assembler

MIXED MODE IN IDE

Mixed mode in IDE

- Use project 10_REMixedModeDemo
- Debugging session must be running
- Explanation of different parts
  - Assignment, addition, if, loop, switch, while
  - Local copy of variable inside switch
    - prevention of race condition issues
- Difference between Debug and Release
  - Removal of dead code
  - Removal of compile-time decidable conditions
  - Optimizations via reserved register (esi)
  - Faster instruction XOR esi, esi
- Difference between Visual Studio and QTCreator

BINARY DEBUGGING

Lena tutorials

- Tutorial 1: basics + binary patching
- Tutorial 2: reversing of algorithm

- Newer browsers may prevent Lena tutorial to run (Flash player required) – enable for use on this page only

OllyDbg - shortcuts

- **F3** ... Open binary file
- **F2** ... Toggle breakpoint (on opcodes, or double click)
- **F9** ... Run debugged program
- **Ctrl+F2** ... Restart program, all temporary changes are lost!
- **F8** ... Step over
- **F7** ... Step into
- **Spacebar** or double click ... allows to set new opcode. Use when you like to change program behaviour, e.g., replacing conditional jump (JGE) by unconditional jump (JMP) or to discard instruction (NOP).
- **Alt+BkSp** ... Undo change

OllyDbg - shortcuts

- ***Rightclick*->Search for->All referenced text strings** ... Constant text strings referenced in code. Use to find strings like hardcoded passwords, important messages ("Wrong license"). Double click on string will takes you to referencing instruction. Helps you to build mind model quickly.
- ***Rightclick*->Find references to->Address constant** ... will find references to particular memory elsewhere in the code – use when you like to know where in code the memory is set, changed or otherwise used.
- **Ctrl+F1** ... Help on Win32 API (WIN32 API help file already prepared in OllyDbg directory (WIN32.HLP)). Use to get meaning of the parameters pushed to stack just before the API function is called.
- **;** ... add or edit your comment for specific code line. Use to write down things you already understand. Use classic paper as well (program mind model)
- ***Rightclick*->Copy to executable->All modifications** (or **Selection**) … make changes permanent. New window with modified code is opened. ***Rightclick*->Save file** to write patched binary to disk.

Homework – Crackme disassembling

- Reverse engineer supplied crackme file pv204.exe
  - Obtain information about its behavior (OllyDbg)
  - Make crackme to continue successfully without error message by
    - a) Patching (modification of control routine, submit patched file)
    - b) Creating valid license info (submit license file)
- Produce short (1xA4) text description of solution
  - How you performed analysis, what you learned, how you solved
- Bonus: More principally different solutions for the same problem might be awarded by extra points
- Submit before: 19.4. 23:59 (full number of points)
  - Every additional started day (24h) means 1.5 points penalization