

A blue parallelogram and a light green parallelogram are positioned in the upper-left corner of the slide. The background features several dark gray diagonal stripes.

Level 0x08

Reverse Engineering



Topics

- Events
- Shirts



Code Quest



- Saturday, February 24
- 2.5 hours. Free breakfast and lunch
- Teams are 2-3 students
 - 1 laptop per person
 - Novice division
 - Advanced division (1 programmer with 1 year programming exp)
- High school students, ages 13-18 years old
- Up to 4 teams per school

Upcoming Events



LOCKHEED MARTIN
CYBERQUEST®
COMPETITION

- Saturday, March 23rd
- [Lockheed Martin Cyber Quest](#)
- 3 hours. Free Breakfast and Lunch
- Teams are 3-5 students
 - 3 laptops per team
- No team limit given



What About the Challenges?

Lockheed Martin CYBERQUEST® challenges are

Hacker History



John Carmack / id Software

- Father of the modern FPS
 - Wolfenstein 3D
 - Doom (Dec 11 was 30th anniversary)
 - Quake



As reported in David Kushner's *Masters of Doom*, when Carmack was 14, he broke into a school with other children to steal Apple II computers. To gain entry to the building, Carmack concocted a sticky substance of thermite mixed with Vaseline that melted through the windows. However, an overweight accomplice struggled to get through the hole and instead opened the window, setting off a silent alarm and alerting police. Carmack was arrested and sent for psychiatric evaluation. He was sentenced to a year in a juvenile home.

Inverse Square Root

Discovered in Quake 3 source

Went viral on /.

Not written by Carmack though...

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y;                                     // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 );                             // what the f ?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this can be removed
}
```

1.4 Accuracy

The first thing one might wonder is, “How good is this approximation?” The graph shown in Figure 1.2 shows the function $y = 1/\sqrt{x}$ and points taken from the `Q_rsqrt()` function at random points across all *floats*.

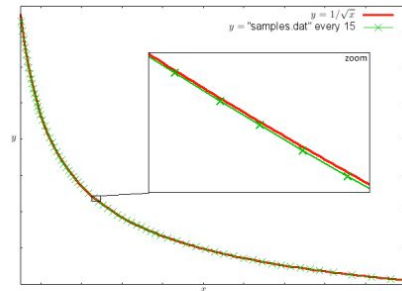
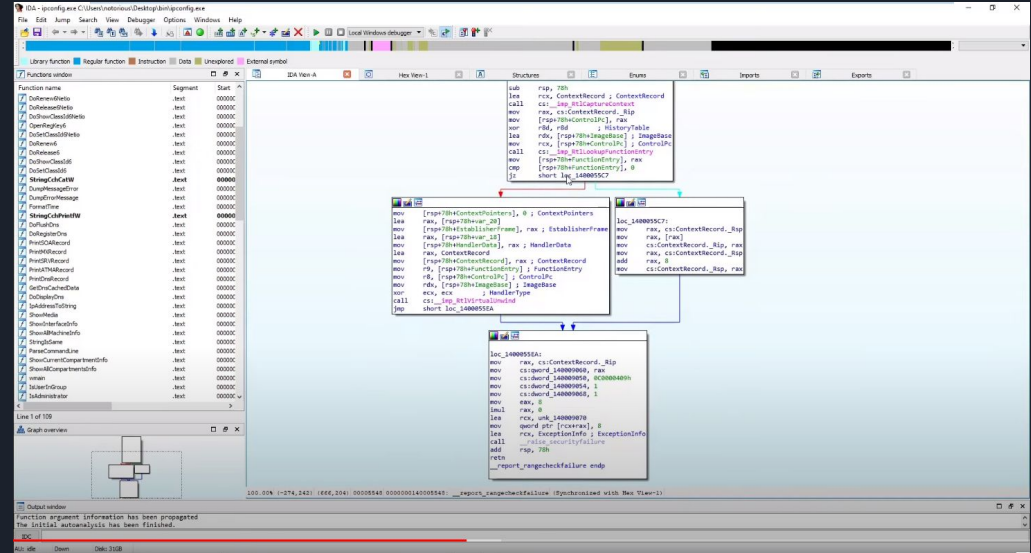


Figure 1.2: Graph of $y = 1/\sqrt{x}$ against sample data.

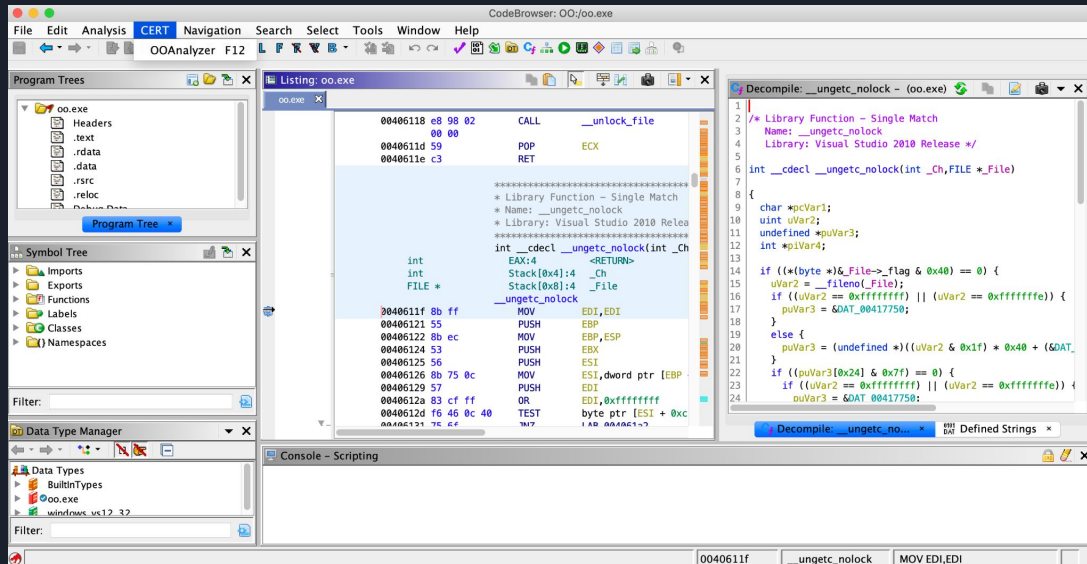
Static Analysis Tools - IDA Pro

- Original RE tool (1990)
- GUI released in 1999
- Supports many CPU archs
- Originally disassembler
- Hexrays decompilers
 - IDA Home \$400
 - IDA Pro \$2,000
 - Decompiler \$2,765
 - ARM32, ARM64, x86, AMD64, MIPS, etc (they are all separate)



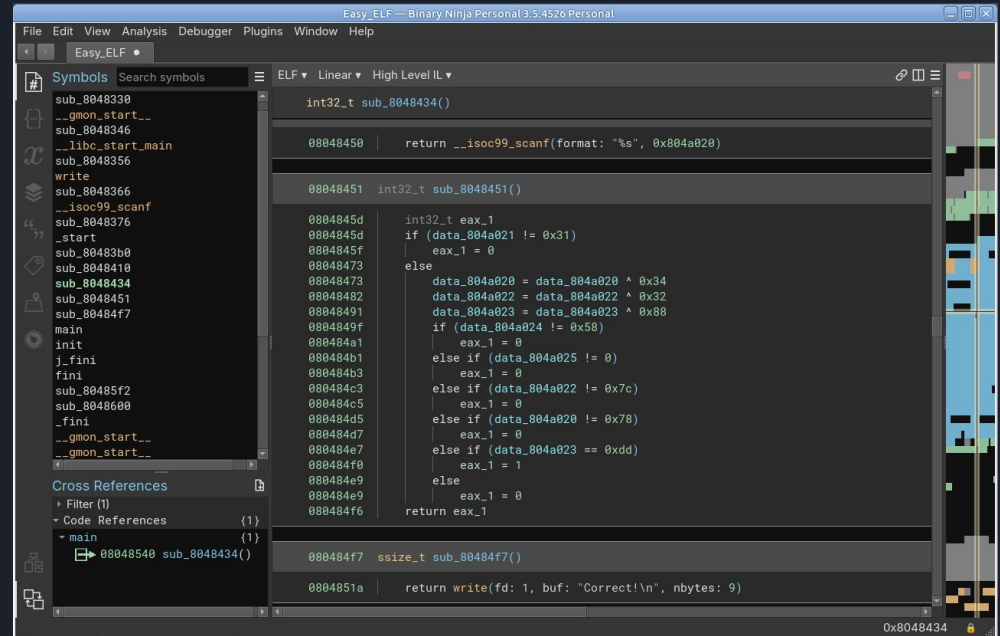
Ghidra

- Used by NSA internally since 2000s
- Released publicly 2019
- Runs in Java
 - Slow
 - Multiplatform
- Disassembler and decompiled code are side by side
- Comes with tutorial
- Free



Binary Ninja

- Originally tool for CTF players for easy patching
- Officially released 2016
- 4 levels of presentation
 - LLIL
 - MLIL
 - HLIL
 - Pseudo-C
- Made here in Melbourne, FL



The screenshot shows the Binary Ninja interface with the title bar 'Easy_ELF - Binary Ninja Personal 3.5.4526 Personal'. The menu bar includes File, Edit, View, Analysis, Debugger, Plugins, Window, and Help. The 'Symbols' panel on the left lists various symbols including sub_8048330, __gmon_start__, sub_8048346, __libc_start_main, sub_8048356, write, sub_8048366, __isoc99_scanf, sub_8048376, _start, sub_80483b0, sub_8048410, sub_8048434, sub_8048451, sub_80484f7, main, init, j_fini, fini, sub_80485f2, sub_8048600, _fini, __gmon_start__, and __gmon_start__. The 'Cross References' panel shows references to sub_8048434() and sub_80484f7(). The main window displays the assembly code for the function 'int32_t sub_8048434()'. The code includes a return statement for __isoc99_scanf, a call to int32_t sub_8048451(), and a series of conditional checks and assignments involving data_804a020, data_804a022, data_804a023, and data_804a024. The function ends with a return statement for eax_1. The status bar at the bottom right shows the address 0x8048434.

```
int32_t sub_8048434()
{
    return __isoc99_scanf(format: "%s", 0x804a020)

    int32_t sub_8048451()
    {
        int32_t eax_1
        if (data_804a021 != 0x31)
            eax_1 = 0
        else
            data_804a020 = data_804a020 ^ 0x34
            data_804a022 = data_804a022 ^ 0x32
            data_804a023 = data_804a023 ^ 0x88
            if (data_804a024 != 0x58)
                eax_1 = 0
            else if (data_804a025 != 0)
                eax_1 = 0
            else if (data_804a022 != 0x7c)
                eax_1 = 0
            else if (data_804a020 != 0x78)
                eax_1 = 0
            else if (data_804a023 == 0xdd)
                eax_1 = 1
            else
                eax_1 = 0
        return eax_1
    }

    ssize_t sub_80484f7()
    {
        return write(fd: 1, buf: "Correct!\n", nbytes: 9)
    }
}
```

- Personal is \$299
- Commercial is \$1499



Static Analysis Process

- Identification of functions versus data / variables
 - Tools will do a great job of identifying functions - Recursive Disassembly
- Set names and types of global variables
- Rename functions to something more helpful
 - Identify type and names for function parameters
 - Part of a C++ class?
- Identify structures and arrays used by code
 - Determine the size of a structure
 - Determine the offsets of each member of a structure
- Comment, document, and understand what application is doing
- Deobfuscate code and variables
 - Obfuscation typical for malware or computer virus



Dynamic Analysis / Debugging

- Start application in debugger
 - Can be painful to pass in arguments
 - You have to be able to run the whole time in the debugger
 - Debugging and application use same window when CLI debugging
- Attach to existing process
 - Debugger running in separate process than application
 - Linux by default won't let you attach debugger to other process for security
 - `cat /proc/sys/kernel/yama/ptrace_scope`
 - `echo "0" | sudo tee /proc/sys/kernel/yama/ptrace_scope`
 - GDB remote
 - Can debug over network or serial connection
 - Can debug code running in QEMU
 - Can debug code running on separate hardware



Links

- https://github.com/id-Software/Quake-III-Arena/blob/master/code/game/q_math.c
- https://en.wikipedia.org/wiki/John_Carmack
- <https://mrober.io/papers/rsqrt.pdf>
- Photo of Carmack by Josh Edelson/AFP via Getty Images