GYMNÁZIUM

JURA HRONCA

(White hat) Hacking

#outline

* Key words & explanation
* background
* tools
* competitions
* progress
* references

Btw. English ?

... simply because majority* of good sec material,
and new sec research is in English ...

* but ofc you can find top security research in other lang as well,
Chinese, French & Russian as notable examples!

program

Computer programming (often shortened to programming) is a process that leads from an original formulation of a computing problem to executable computer programs.

...

The purpose of programming is to find a **sequence of instructions that will automate** performing a specific task or ***solving a given problem***

http://en.wikipedia.org/wiki/Computer_programming

hacker

Hacker (term), is a term used in computing that can describe several types of persons

1. Hacker (computer security) someone who seeks and exploits weaknesses in a computer system or computer network
2. Hacker (hobbyist), who makes innovative customizations or combinations of retail electronic and computer equipment
3. Hacker (programmer subculture), **who combines excellence, playfulness, cleverness and exploration in performed activities**

http://en.wikipedia.org/wiki/Hacker

vulnerability

In computer security, a vulnerability is a weakness which allows an attacker to reduce a system's information assurance. Vulnerability is the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw.[1] To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerability is also known as the **attack surface**

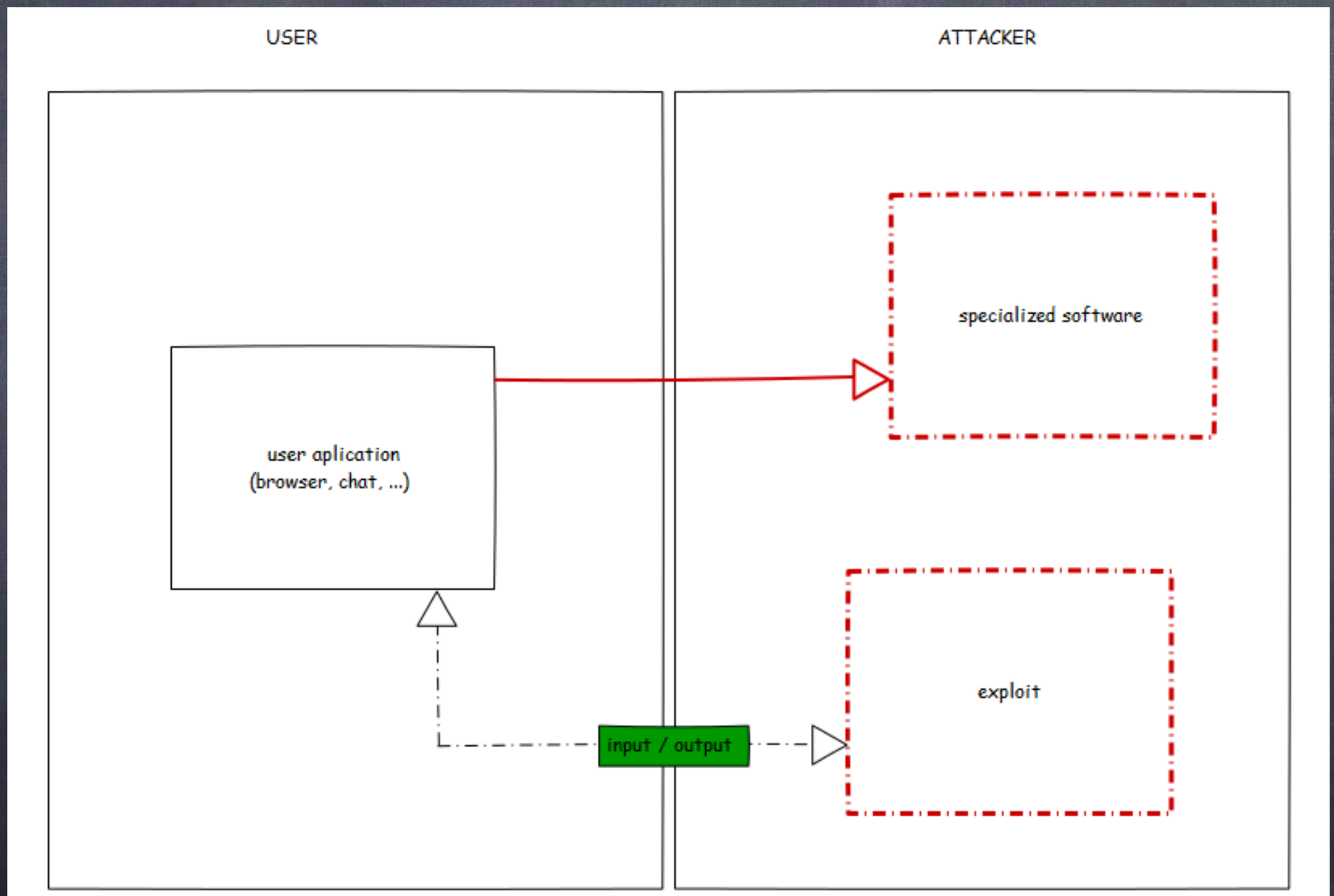
http://en.wikipedia.org/wiki/Vulnerability_(computing)

exploitation

An exploit (from the English verb to exploit, meaning "using something to one's own advantage") is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability in order to cause ***UNINTENDED OR UNANTICIPATED BEHAVIOR*** to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service attack.

http://en.wikipedia.org/wiki/Exploit_(computer_security)

exploitation



Exploitation - ??? guys

- Hunt vulnerabilities
  - Write fuzzers, checkers, support tools ...
    - Use odays for their own reasons, cyber weapons, spying..
- Invent / copy methodologies
  - Misuse hole in protection mechanism for attack!
  - Do oday business with 3rd party
  - Keep their research private

Exploitation - good guys

- Hunt vulnerabilities
  - Write fuzzers, checkers, support tools ...
    - Report to vendors & Cooperate on fix
- Invent new methodologies
  - To uncover weakness of current protection mechanism
  - Cooperate on effective mitigations
  - Share research with community for faster improvement

background

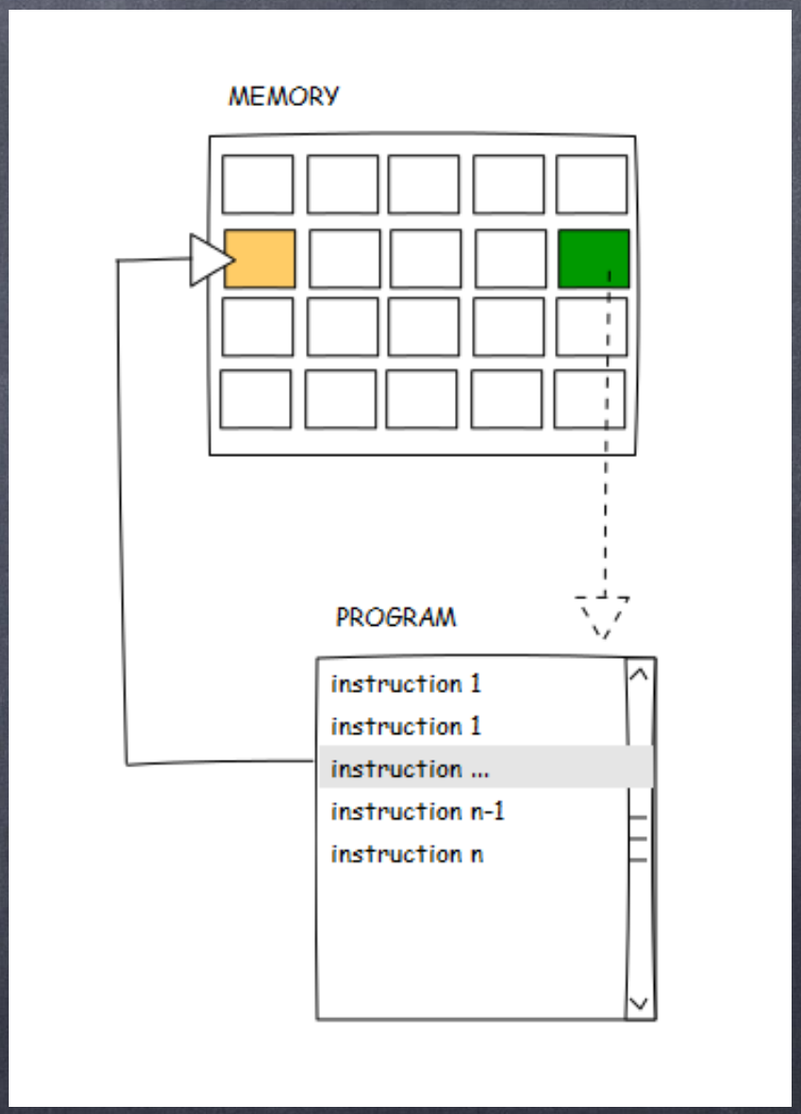
... program, code, reverse engineering ...

Programing ?

* Program :
Transformation of question / task to math-logic problem

* Code :
Smart calculator based on sequences of reads and writes

* Performance
how smart you build logic of your calculator



coding

- Understand basics
  - Arrays, conditions, loops
- Choose language
  - I recommend : C++ / python / Go
- Learn algorithms
  - Binary trees, hash-tables, string search, divide & conquer, dijkstra, ...
  - Performance & memory complexity, O(notation)

CODE : Bubble sort ?

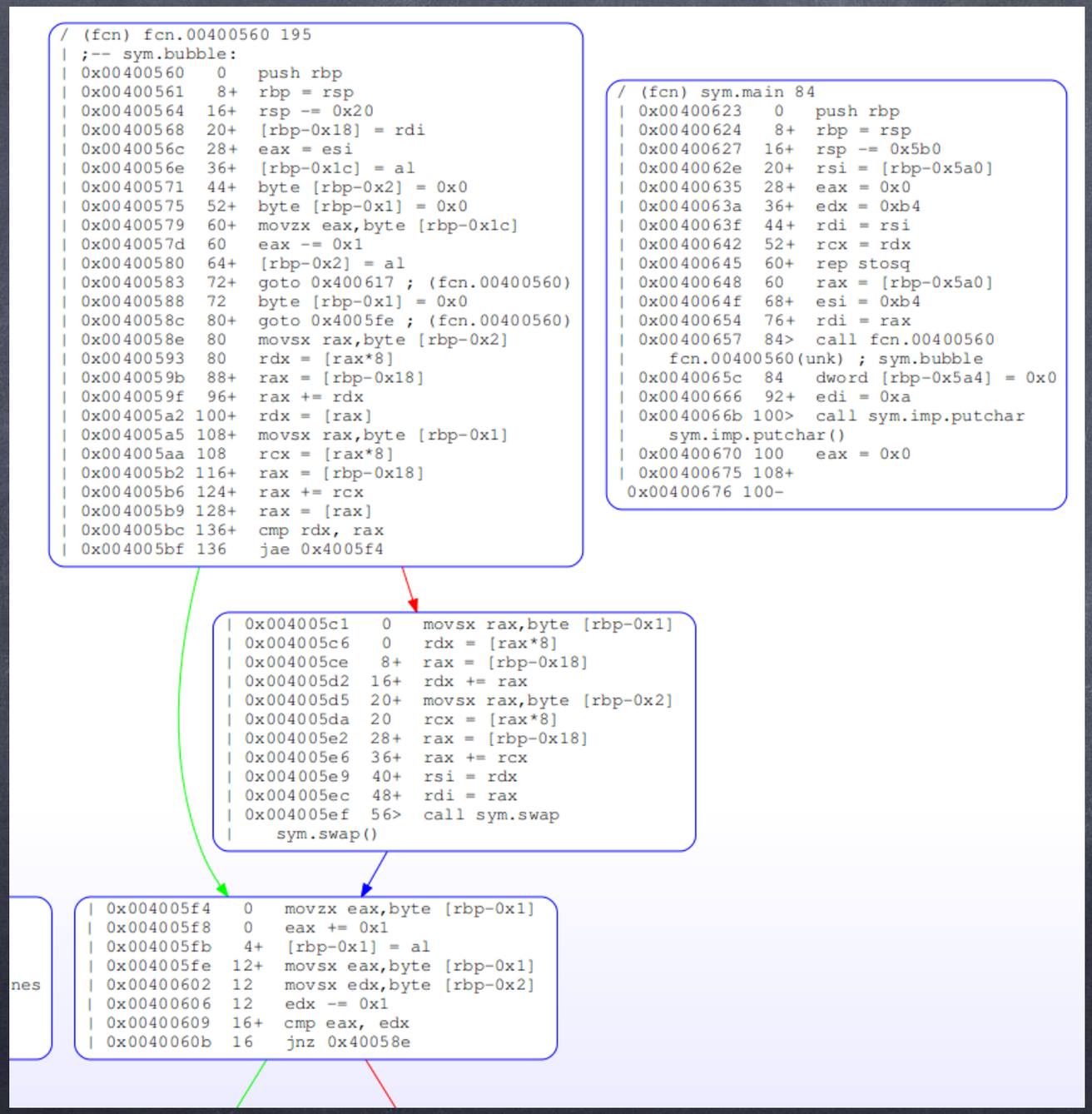
```c
void
swap(
  unsigned long long* a,
  unsigned long long* b
  )
{
  int temp = *a;
  *a = *b;
  *b = temp;
}

void
bubble(
  unsigned long long* arr,
  unsigned char len
  )
{
  char i = 0, j = 0;
  for (i = len - 1; i != -1; i--)
    for (j = 0; j != i - 1; j++)
      if (arr[i] < arr[j])
        swap(arr + i, arr + j);
}

int
main()
{
  unsigned long long a[180] = { 0 };
  //read data to buffer
  bubble(a, 180);
```

http://www.vim.org/

Reversing (RE)

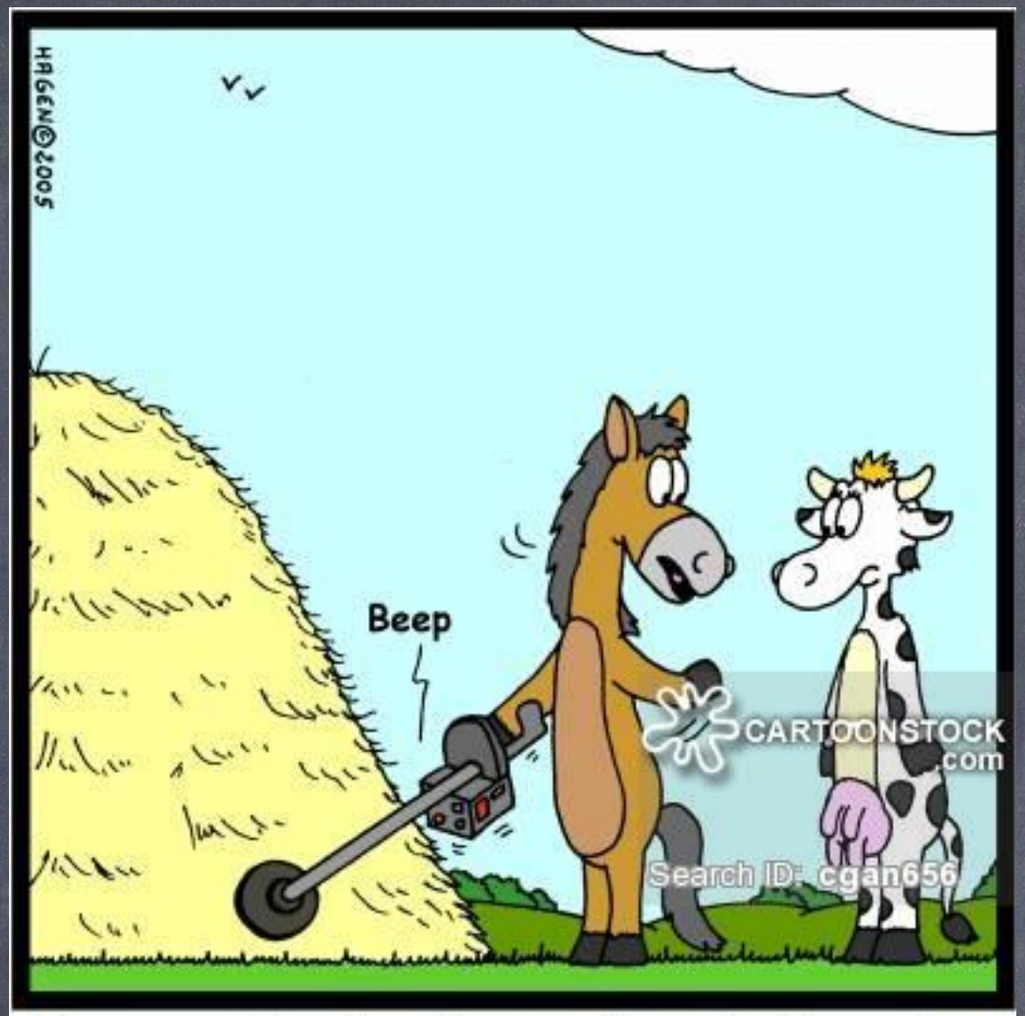
- Understand memory & cpu
  - How are data stored
  - Instructions – assembler
    - X86, arm
- Understand "program->compiler->assembly"
  - Variables
  - Functions
  - Loops & calls

RE : Bubble sort ?



https://inguma.eu/projects/bokken

http://www.radare.org/r/



You were right: There's a needle in this haystack...

Here we are!

... did you spot something ? ...

VULNERABILITY
Bubble sort !

```
void
bubble(
    unsigned long long* arr,
    unsigned char len
    )
{
    for (char i = len - 1; i != -1; i--)
        for (char j = 0; j != i - 1; j++)
            if (arr[i] < arr[j])
                swap(arr + i, arr + j);
}
```

https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx

As signed numbers can represent **NEGATIVE** numbers, they lose a range of positive numbers that can only be represented with unsigned numbers of the same size (in bits) because roughly half the possible values are non-positive values (so if an 8-bit is signed, positive unsigned values 128 to 255 are gone while -128 to 127 are present). Unsigned variables can dedicate all the possible values to the positive number range.

EXPLOITATION

Bubble sort !

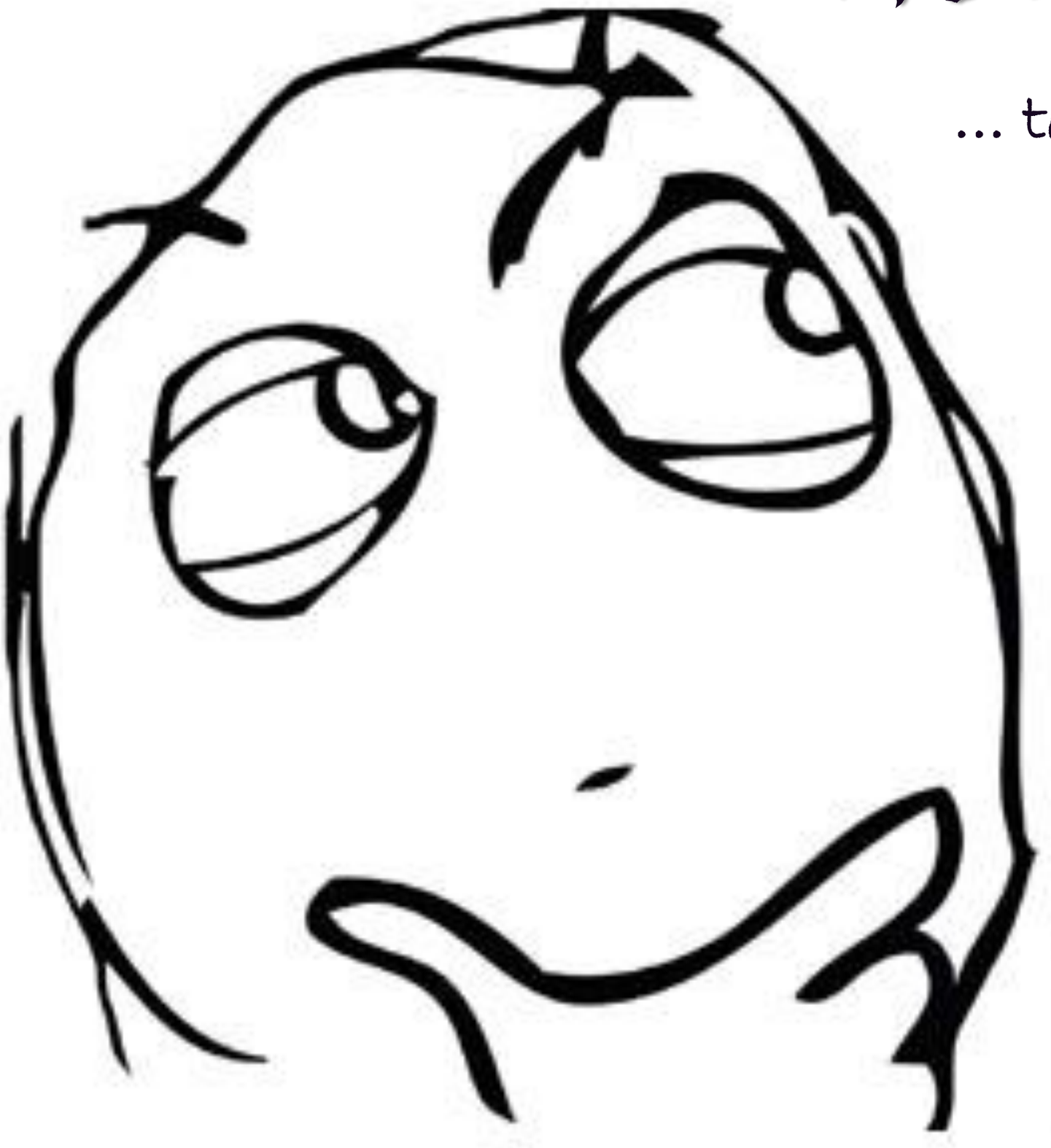
```
12  void
13  bubble(
14    unsigned long long* arr,
15    unsigned char len
16    )
17  {
18    for (char i = len - 1; i != -1; i--)
19      for (char j = 0; j != i - 1; j++)
20        if (arr[i] < arr[j])
21          swap(arr + i, arr + j);
22  }
```


```
/ (fcn) sym.main 84
| 0x00400623   0-   55                    push rbp
| 0x00400624   8+   4889e5                rbp = rsp
| 0x00400627  16+   4881ecb0050.          rsp -= 0x5b0
| 0x0040062e  20+   488db560faf.          rsi = [rbp-0x5a0]
| 0x00400635  28+   b800000000            eax = 0x0
| 0x0040063a  36+   bab4000000            edx = 0xb4
| 0x0040063f  44+   4889f7                rdi = rsi
| 0x00400642  52+   4889d1                rcx = rdx
| 0x00400645  60+   f348ab                rep stosq
| 0x00400648  60    488d8560faf.          rax = [rbp-0x5a0]
| 0x0040064f  68+   beb4000000            esi = 0xb4
| 0x00400654  76+   4889c7                rdi = rax
| 0x00400657  84>   e804ffffff            call fcn.00400560
|                       fcn.00400560(unk) ; sym.bubble
| 0x0040065c  84    c7855cfafff.          dword [rbp-0x5a4] = 0x0
```

STACK (32bit case)
VAR2 (i)
VAR1 (j)
☑ RET
ARG1 (buf_ptr)
ARG2 (len)
VAR-buf[0]
VAR-buf[1]
...
VAR-buf[n]

How to Start

... tools, competitions ...



HMM...

IDE (+ plugins!) programming environment

- Visual Studio 2013 (community edition)

- Vim

- Sublime

REVERSE ENGENEERING

- bokken

- windbg

- gdb (lldb)

GDB
The GNU Project
Debugger

Virtual machine + emulators

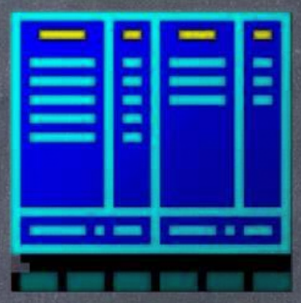
- Virtual Box

- Bochsd

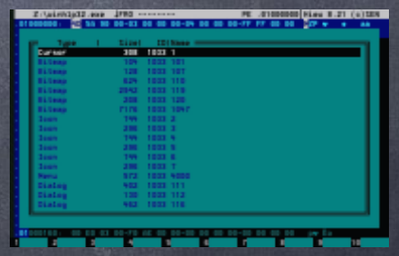
- Qemu

Additional tools (win)

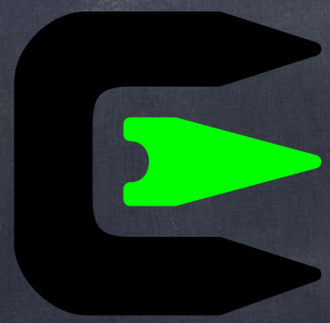
- ConEmu (far manager)



- Hiew



- cygwin



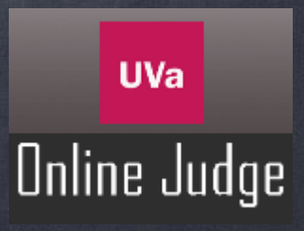
ALGO - COMPETITIONS

Slovakia

- ZENIT
- KSP
- Matfyz -RP
- nationals

international

- TOPCODER
- ACM
- UVA
- IOI



[TopCoder]®

CTF - COMPETITIONS

ghost in the shellcode

XCTF
全国网络安全技术对抗联赛

DEFCON

0CTF 2015
0OPS

ConfidEncE

DRAGON SECTOR

PCTF 2011

Final words

... advices, references ...

how to progress

- Self - learning
  - For ever and ever best approach
- *DO SPORT*
  - Keep balanced body and mind
  - essential for creative ideas ;)
- HARDwork
  - Push 110% to everything in your life (learning, sport, work, study, ...)

#whoami

* Peter Hlavaty - @zer0mem
* GJH (2004-2008)
* Matfyz (2008-2010)
* ESET (2010-2014)
* KEEN (2014-...)
* Conferences (...)
* Lectures (...)
* Pwn Events (...)

- Feel free to ContacT me
- I will try to help
(with some delay +- :)

References - tools

editor:
http://www.vim.org/
https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx
http://www.sublimetext.com/

virtual :
https://www.virtualbox.org/
http://bochs.sourceforge.net/
http://wiki.qemu.org/Main_Page

re :
https://inguma.eu/projects/bokken
http://www.radare.org/r/
http://www.capstone-engine.org/
http://www.windbg.org/
https://msdn.microsoft.com/en-us/library/windows/hardware/ff551063(v=vs.85).aspx
http://www.gnu.org/software/gdb/
http://lldb.llvm.org/

tools:
http://www.farmanager.com/
http://www.hiew.ru/
http://conemu.github.io/
https://www.cygwin.com/

References - events

http://uva.onlinejudge.org/

http://ctf.codegate.org/

https://www.topcoder.com/community/data-science/data-science-tutorials/

https://ctf.0ops.sjtu.cn/

https://arena.topcoder.com/#/a/home

https://legitbs.net/

http://ghostintheshellcode.com/

http://zenit.edu.sk/

http://play.plaidctf.com/

https://www.ksp.sk/

https://ctf.dragonsector.pl/

http://people.ksp.sk/~acm/welcome.php