Software Obfuscation with LLVM

(Ab)using the compiler to “protect” code

Bio

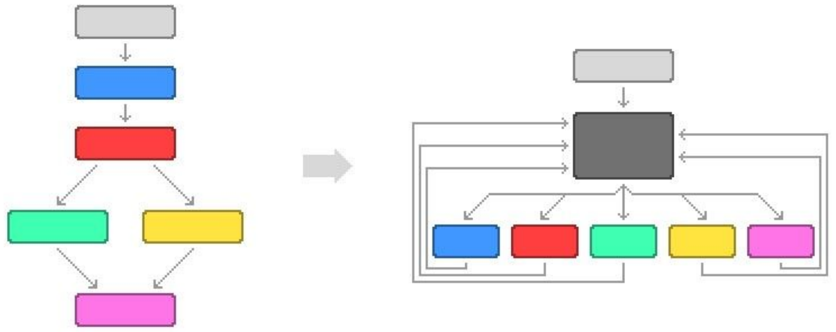
- Carl Svensson
- Head of Security, KRY/LIVI
- CTF: HackingForSoju
- Twitter: @zetatwo
- Email: calle.svensson@zeta-two.com
- Website: https://zeta-two.com

Agenda

- Software obfuscation
- Compilers
    - LLVM
- LLVM for obfuscation
- Testing
- Counter attacks

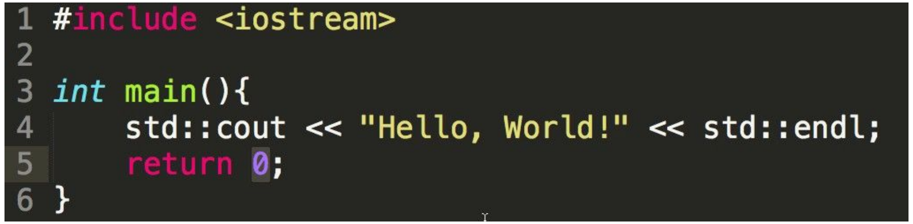
Software obfuscation

- Level
  - Source
  - Intermediate
  - Machine code
- Categories
  - Control flow flattening
  - Self-modifying code
  - Dead code
  - Packers
  - Droppers
  - Anti-debugging
  - VM

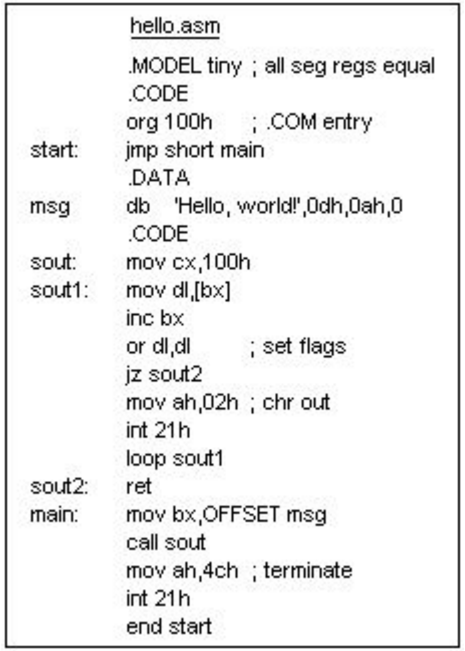


Compilers

- Transform language
- Human readable to machine
- Example: C to x86
- Example: Rust to ARM



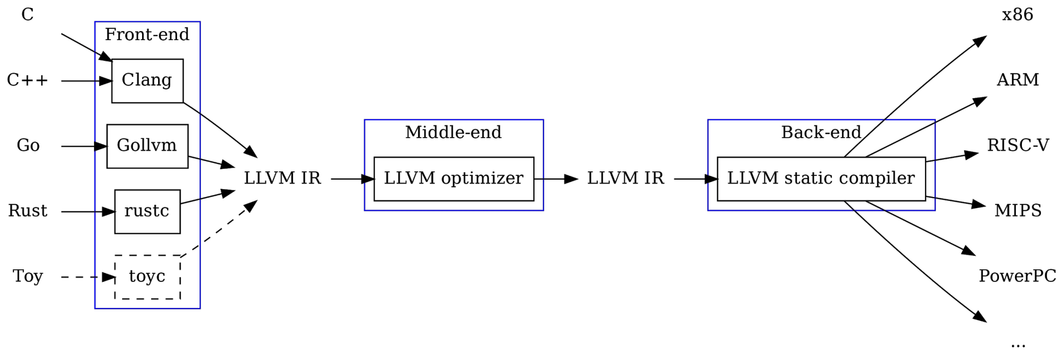
```cpp
#include <iostream>

int main(){
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

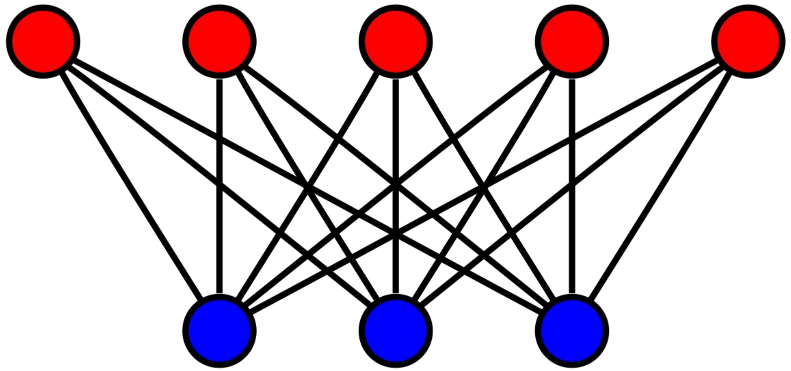


```
          hello.asm
          .MODEL tiny ; all seg regs equal
          .CODE
          org 100h      ; .COM entry
start:    jmp short main
          .DATA
msg       db   'Hello, world!',0dh,0ah,0
          .CODE
sout:     mov cx,100h
sout1:    mov dl,[bx]
          inc bx
          or dl,dl        ; set flags
          jz sout2
          mov ah,02h ; chr out
          int 21h
          loop sout1
sout2:    ret
main:     mov bx,OFFSET msg
          call sout
          mov ah,4ch  ; terminate
          int 21h
          end start
```

LLVM

- Compiler framework
- L+A instead of L*A
  - L number of languages
  - A number of architectures
- Single target
- A lot of tools exist
  - Manticore

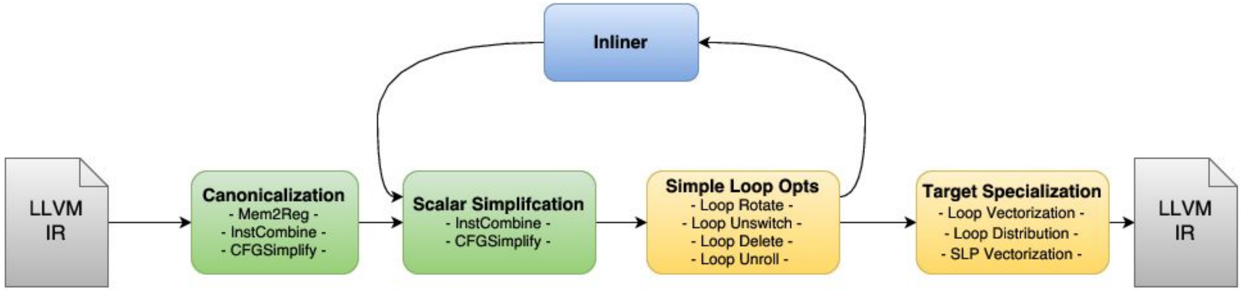




Writing an LLVM pass

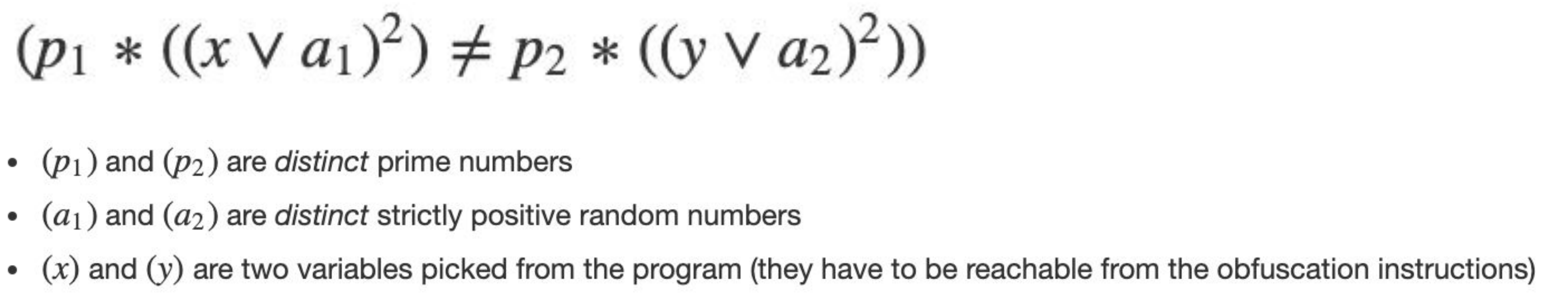
- Simple
- “Constrained” to LLVM API
- Example: Quarkslab’s

```cpp
#include "llvm/Pass.h"
#include "llvm/IR/Function.h"
#include "llvm/Support/raw_ostream.h"

#include "llvm/IR/LegacyPassManager.h"
#include "llvm/Transforms/IPO/PassManagerBuilder.h"

using namespace llvm;

namespace {
class MyPass : public BasicBlockPass {
public:
  static char ID;

  MyPass() : BasicBlockPass(ID) {}

  bool runOnBasicBlock(BasicBlock &BB) override {
    errs() << "I m running on a block...\n";
    return false;
  }

};
}

char MyPass::ID = 0;
static RegisterPass<MyPass> X("MyPass", "Obfuscates zeroes",
                              false, false);

// register pass for clang use
static void registerMyPassPass(const PassManagerBuilder &,
                               PassManagerBase &PM) {
  PM.add(new MyPass());
}
static RegisterStandardPasses
    RegisterMBAPass(PassManagerBuilder::EP_EarlyAsPossible,
                    registerMyPassPass);
```



Writing an obfuscating LLVM pass

- Simple
- “Constrained” to LLVM API
- Example: Quarkslab’s



$$(p_1 * ((x \vee a_1)^2) \neq p_2 * ((y \vee a_2)^2))$$

- (p_1) and (p_2) are *distinct* prime numbers
- (a_1) and (a_2) are *distinct* strictly positive random numbers
- (x) and (y) are two variables picked from the program (they have to be reachable from the obfuscation instructions)

Forking LLVM

- More complicated
- Full control
- Example: Obfuscator-LLVM

“The aim of this project is to provide an open-source fork of the LLVM compilation suite able to provide increased software security through code obfuscation and tamper-proofing.”

Testing

- Write some unit tests
- Utilize an existing large project
    - Example: OpenSSL

Antidote?

- Static analysis
  - Build unpacker
- Symbolic execution
  - Generic
  - Specific
- Dynamic analysis
  - Tracing
  - Fuzzing
  - Manual

Sources

- Obfuscator-LLVM: https://github.com/obfuscator-llvm/obfuscator/wiki
- Quarkslab:
  - https://blog.quarkslab.com/turning-regular-code-into-atrocities-with-llvm.html
  - https://blog.quarkslab.com/deobfuscation-recovering-an-ollvm-protected-program.html
- https://yurichev.com/blog/llvm/
- https://github.com/0vercl0k/stuffz/blob/master/llvm-funz/kryptonite/llvm-functionpass-kryptonite-obfuscater.cpp
- https://doar-e.github.io/blog/2013/09/16/breaking-kryptonites-obfuscation-with-symbolic-execution/

Thanks for listening

Questions?

Email: calle.svensson@zeta-two.com