To LLVM Bytecode Obfuscation and Beyond

Serge ≪ sans paille ≫ Guelton

Unintentionally Developing an LLVM Fuzzer

~~To LLVM Bytecode Obfuscation and Beyond~~

Serge ≪ sans paille ≫ Guelton

Replace all constants with an opaque computation

vs.

Clang

```
vg_assert(cfsi.len < 5000000);
```

from coregrind/m_debuginfo/storage.c

```
assert(NumOperands == Ops.size() &&
       "NumOperands wasn't ...");
```

from llvm/CodeGen/SelectionDAGNodes.h

```
RandomNumberGenerator *
Module::createRNG(const Pass* P) const;
```

```
Function::Create(funcType,
    GlobalValue::InternalLinkage,
    oldFunction->getName() + "_"
    + header->getName(),
    M);
```

See https://sourceware.org/bugzilla/show_bug.cgi?id=18581

```
%div = sdiv i512 %a, %b
```

```
define i8 @f(i8 %a) {
 %1 = bitcast i8 %a to <8 x i1>
 %2 = insertelement <8 x i1> %1, i1 1, i8 2
 %3 = bitcast <8 x i1> %2 to i8
 ret i8 %3
}
```

```
% clang -mno-sse -m32 -O2
mov       al, [esp+arg_0]
or        al, 4
```

% clang -m32 -O2

```
push    ebx
push    edi
push    esi
sub     esp, 10h
mov     al, [esp+1Ch+arg_0]
mov     [esp+1Ch+var_14], al
movzx   edx, [esp+1Ch+var_14]
mov     esi, edx
mov     edi, edx
mov     eax, edx
shr     eax, 7
mov     [esp+1Ch+var_1C], al
mov     ebx, edx
mov     ecx, edx
mov     eax, edx
and     edx, 1
[...]
```

1. Design an obfuscation
2. Unittest your obfuscation
3. Compile CMake with the obfuscator
4. Run test suite
   → Validation fails
   → Debug obfuscated code

1. Design an obfuscation
2. Unittest your obfuscation
3. Compile CMake with the obfuscator
4. Run test suite
   $\rightarrow$ Validation fails
   $\rightarrow$ Debug obfuscated code

1. Design an obfuscation
2. Unittest your obfuscation
3. Compile CMake with the obfuscator
4. Run test suite
   → Validation fails
   → Debug obfuscated code

1. Design an obfuscation
2. Unittest your obfuscation
3. Compile CMake with the obfuscator
4. Run test suite
   $\rightarrow$ Validation fails
   $\rightarrow$ Debug obfuscated code

1. Design an obfuscation
2. Unittest your obfuscation
3. Compile CMake with the obfuscator
4. Run test suite
   → Validation fails
   → Debug obfuscated code