How much can we learn in a single execution?

Dynamic reverse-engineering in one shot

December 8th, 2015

Roland Groz Franck de Goër Laurent Mounier

Université Grenoble Alpes - FRANCE

Motivation?

How much can we learn in a single execution?

This much

Function prototypes

This much

Function prototypes

- arity

This much

Function prototypes

- arity
- type of parameters

This much

Function prototypes

- arity
- type of parameters

Function coupling

This much

Function prototypes

- arity
- type of parameters

Function coupling

`malloc` ⟶ `free`

This much

Function prototypes

- arity
- type of parameters

Function coupling

`malloc` ⟶ `free`

`fopen` ⟶ `fclose`

How much can we learn in a single execution?

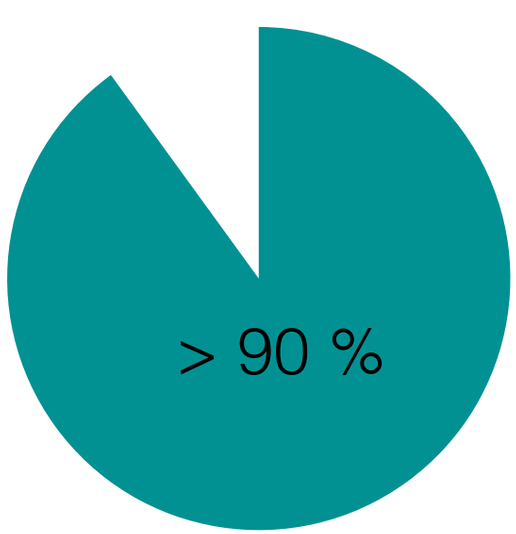
This much

Accuracy:

This much

Accuracy:

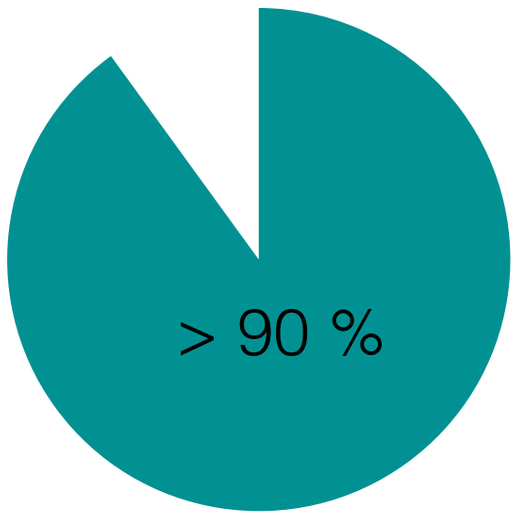
Function arity



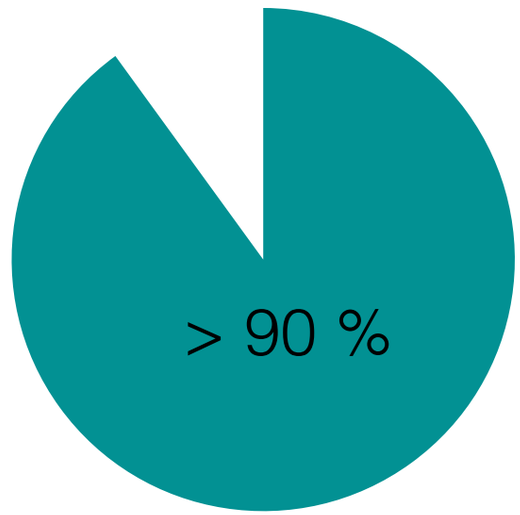
This much

Accuracy:

Function arity



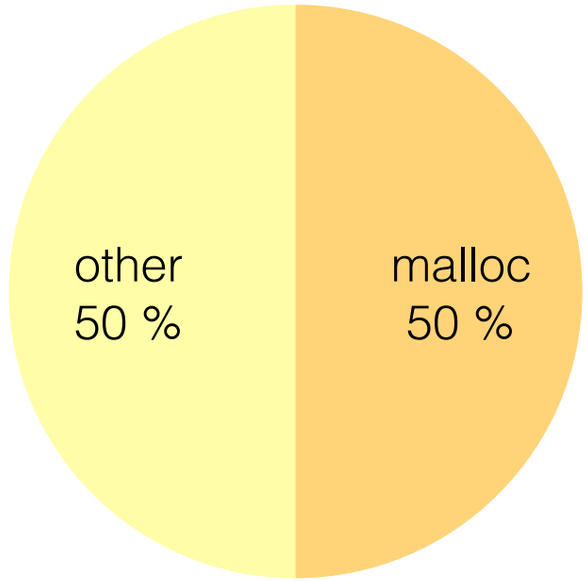
Parameter types



This much

Coupling (on MuPDF):

- 526 functions
- 578 couples
- 42 unique left-side functions



Key Idea

Lightweight

Lightweight dynamic

Key Idea

Lightweight dynamic instrumentation

Key Idea

Lightweight dynamic instrumentation

to recover

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and coupling

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and coupling from

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and coupling from binaries

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and coupling from binaries

in

Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

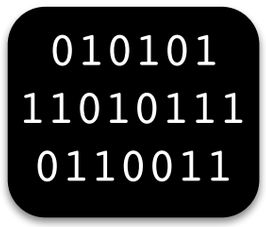
Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

Object we study:

Object we study:

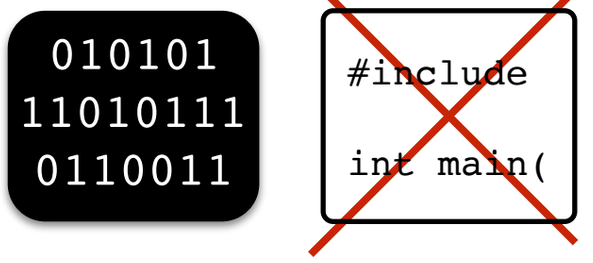
- binary program



binaries

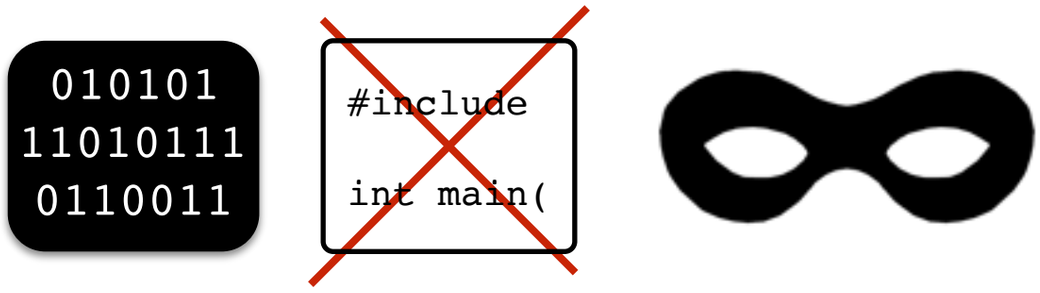
Object we study:

- binary program
- no source code



Object we study:

- binary program
- no source code
- possibly stripped
- possibly obfuscated



Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

We want to recover:

We want to recover:

- the number of parameters ⟶ function arity

We want to recover:

- the number of parameters ⟶ function arity
- the type of parameters ⟶ function prototype

Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

Intuition:

Functions that are semantically linked

- `malloc` ⟶ `free`
- `fopen` ⟶ `fclose`

Definition. ρ-coupling

Two functions `f` and `g` are ρ-coupled if for any execution `e`, there is a data-flow from the returned value of `f` to a given parameter of `g` in a proportion of calls to `g` greater than ρ.

Key Idea

Lightweight dynamic instrumentation

to recover function prototypes

and coupling from binaries

in one execution

#1 — arity

- Calling convention specifies the way parameters are passed to functions
- For example, `x86-64` calling convention:
  - Parameters passed through registers
  - For integers/addr: `%rdi`, `%rsi`, `%rdx`, `%rcx`, `%r8`, `%r9`
  - For floats: `%xmm0`–`%xmm7`

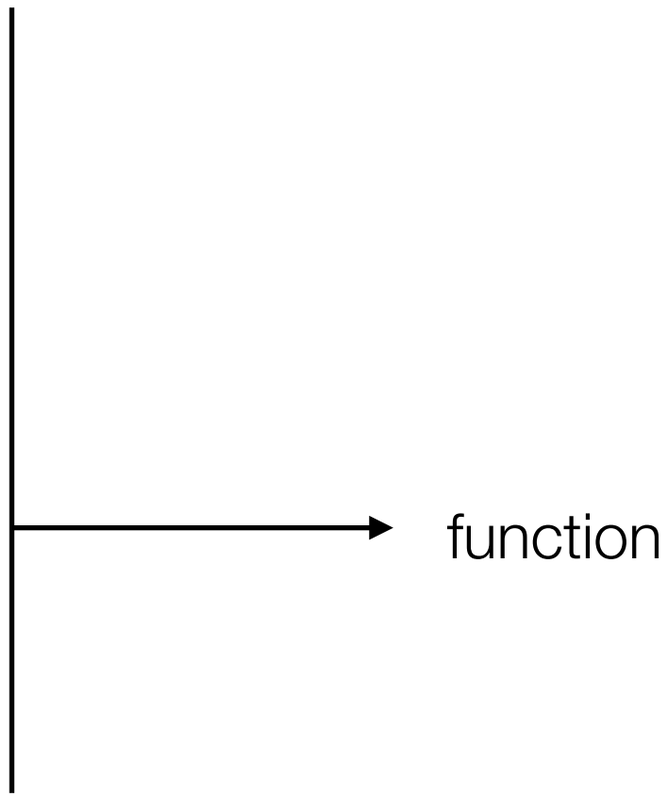
#1 — arity

Registers read before written are parameters.

to recover

#1 — arity

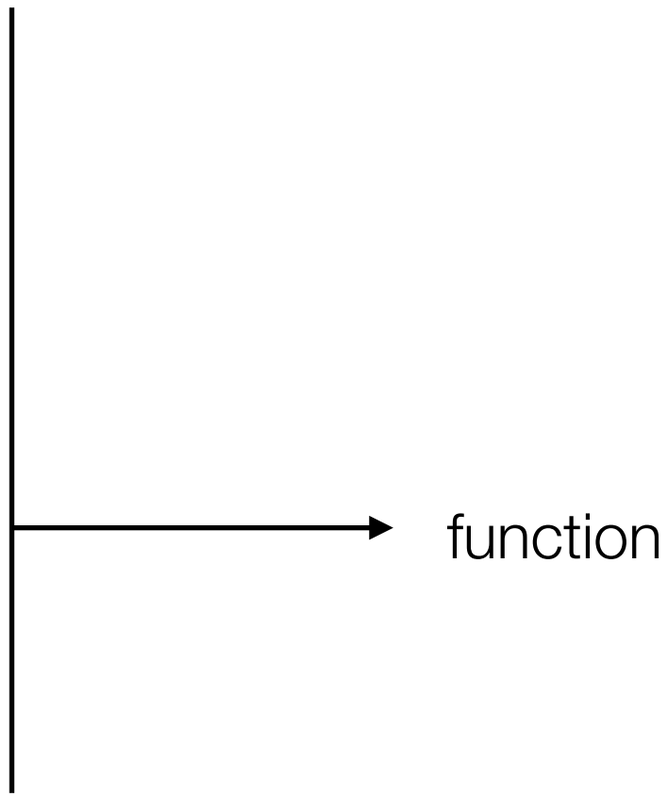
```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```



to recover

#1 — arity

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```



to recover

#1 — arity

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

→ function

- (at least) one parameter

to recover

#1 — arity

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

→ function

- (at least) one parameter

to recover

#1 — arity

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

→ function

- (at least) one parameter
- returning void

#2 — type

- Three types considered: INT, ADDR, FLOAT
- Distinction INT/ADDR vs. FLOAT given by register
- Main problem: distinction INT vs. ADDR

#2 — type

A location dereferenced (e.g. in `load`/`store`) is an address.

to recover

#2 — type

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
```

INT or ADDR ?

```
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

to recover

#2 — type

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
```

```
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

to recover

#2 — type

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
```

ADDR

```
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

to recover

#2 — type

```
401e55:       75 a9                   jne    401e00
401e57:       90                      nop
401e58:       c9                      leaveq
401e59:       c3                      retq
401e5a:       55                      push   %rbp
401e5b:       48 89 e5                mov    %rsp,%rbp
401e5e:       48 83 ec 30             sub    $0x30,%rsp
401e62:       48 89 7d d8             mov    %rdi,-0x28(%rbp)
401e66:       48 8b 45 d8             mov    -0x28(%rbp),%rax
401e6a:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
  …    :      …                       …      …
401e8a:       eb 2e                   jmp    401eba
401e8c:       48 8b 45 f8             mov    -0x8(%rbp),%rax
```

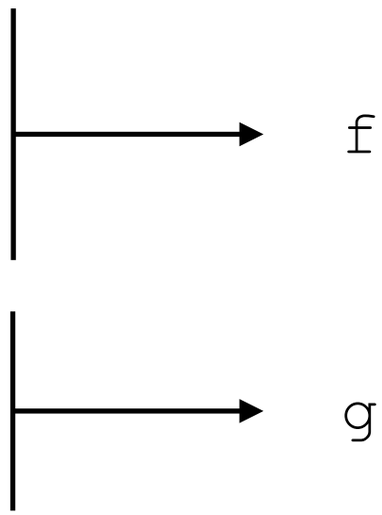
```
401e90:       8b 40 20                mov    0x20(%rax),%eax
401e93:       85 c0                   test   %eax,%eax
401e95:       75 17                   jne    401eae
401e97:       48 8b 55 d8             mov    -0x28(%rbp),%rex
401e9b:       48 8d 4d e0             lea    -0x20(%rbp),%rcx
  …    :      …                       …      …
401eb6:       48 89 45 f8             mov    %rax,-0x8(%rbp)
401eba:       48 83 7d f8 00          cmpq   $0x0,-0x8(%rbp)
401ebf:       75 cb                   jne    401e8c
401ec1:       90                      nop
401ec2:       c9                      leaveq
401ec3:       c3                      retq
401ec4:       55                      push   %rbp
401ec5:       48 89 e5                mov    %rsp,%rbp
  …    :      …                       …      …
401fcb:       48 89 7d f8             mov    %rdi,-0x8(%rbp)
401fcf:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fd3:       48 89 c7                mov    %rax,%rdi
401fd6:       e8 7f fe ff ff          callq  401e5a
401fdb:       48 8b 45 f8             mov    -0x8(%rbp),%rax
401fdf:       48 8b 80 b0 27 00 00    mov    0x27b0(%rax),%rax
```

#3 — coupling

A shared value between a ret and a param. includes a coupling.

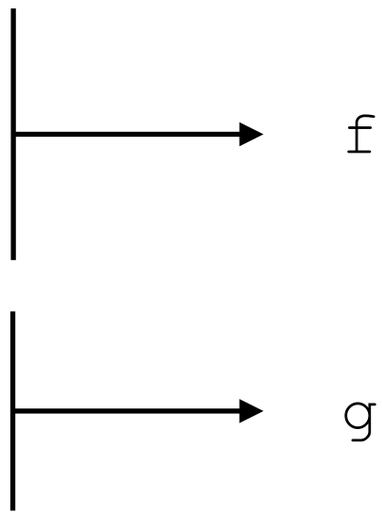
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



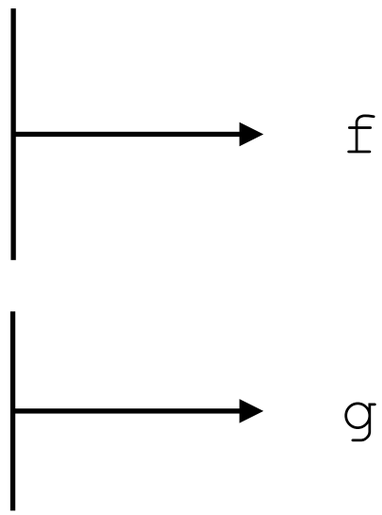
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

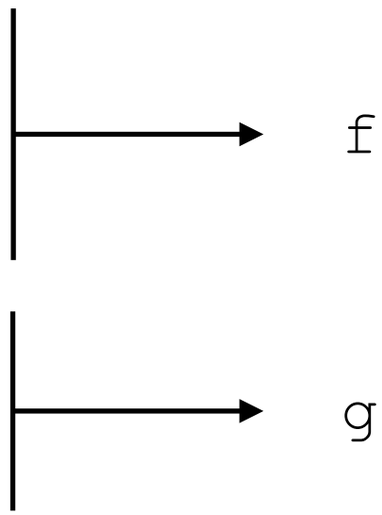
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```

f

g

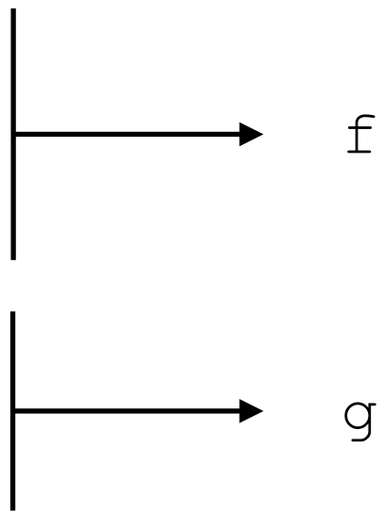
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



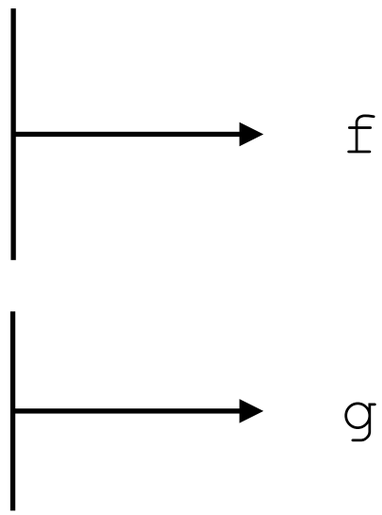
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



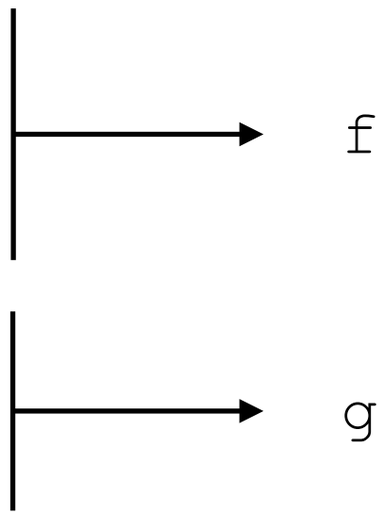
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



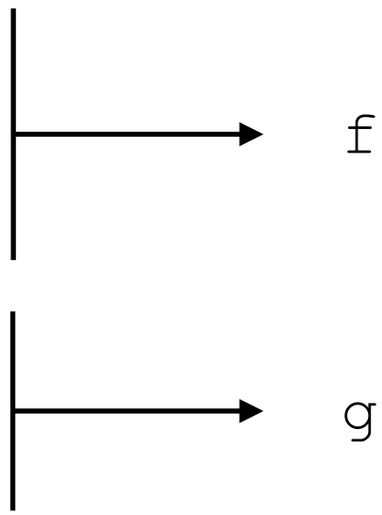
#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

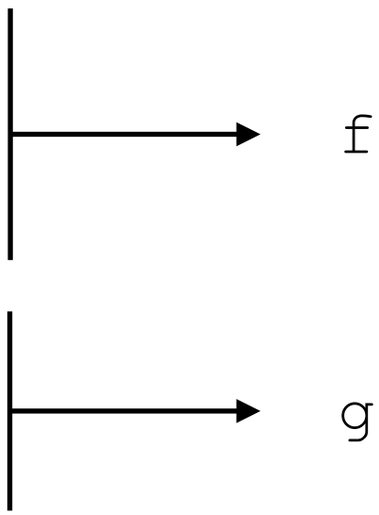
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

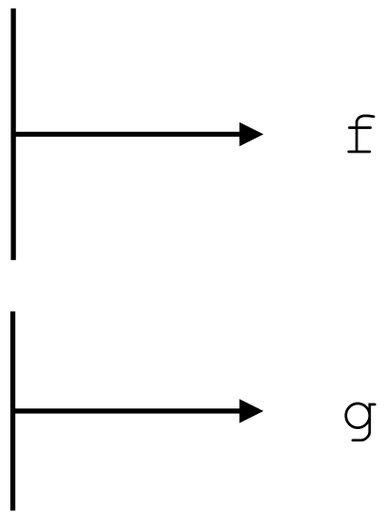
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

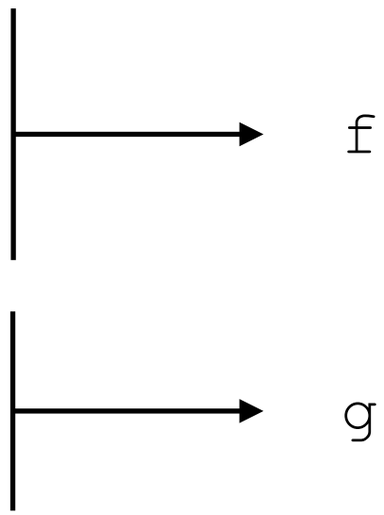
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

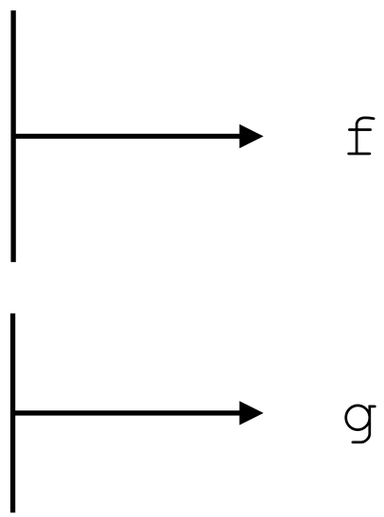
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```

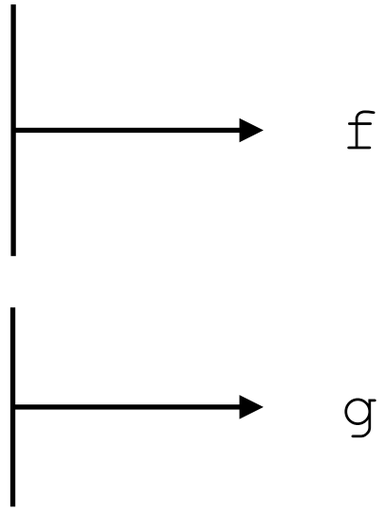


to recover

#3 — coupling

1) `f` returns `0x7fe047419010`

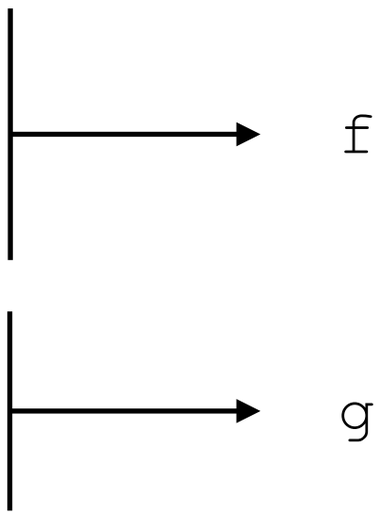
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

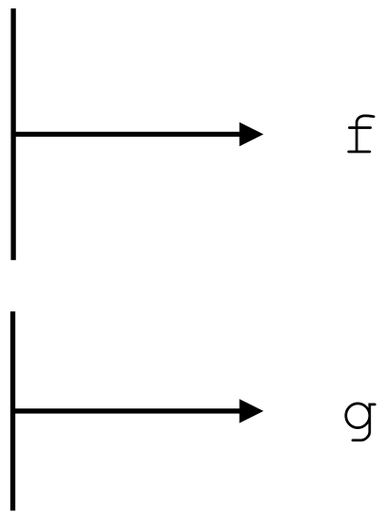
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

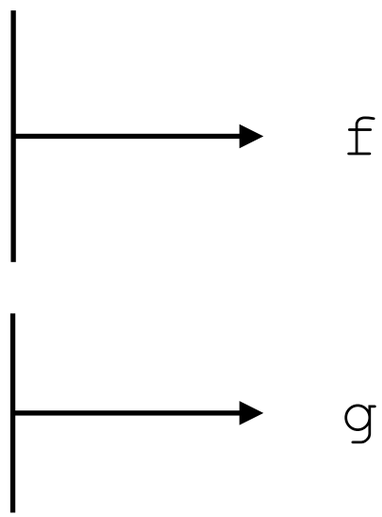
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

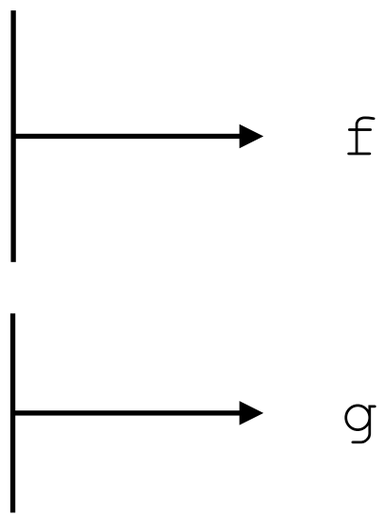
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

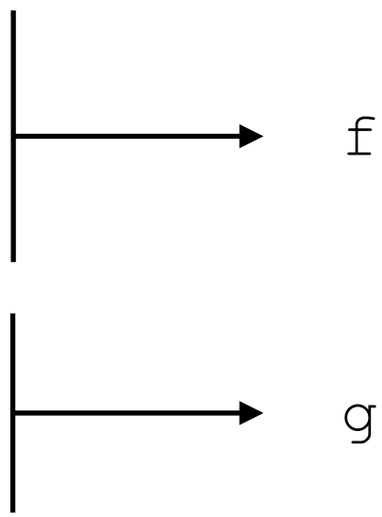
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



#3 — coupling

1) `f` returns `0x7fe047419010`

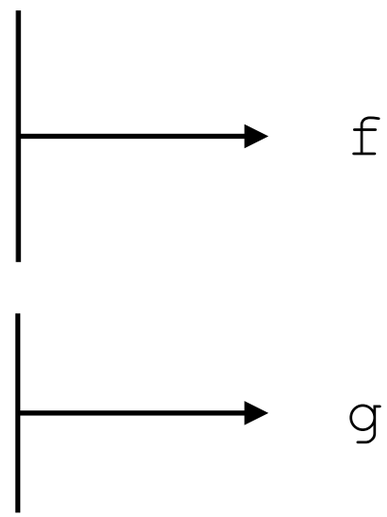
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



to recover

#3 — coupling

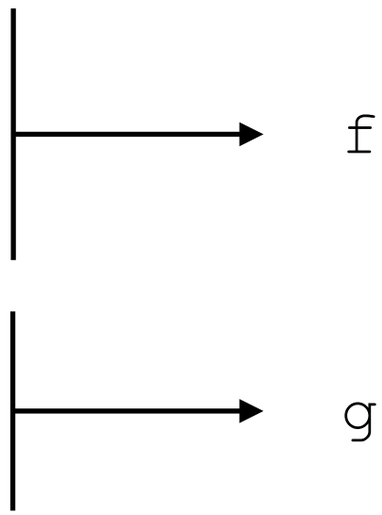
```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



1) `f` returns `0x7fe047419010`
2) `g` takes `0x7fe047419010`

#3 — coupling

```
...    :        ...                     ...
40059f:         bf 28 00 00 00          mov    $0x28,%edi
4005a4:         e8 40 01 00 00          callq  4006e9
4005a9:         48 89 45 f8             mov    %rax,-0x8(%rbp)
...    :        ...                     ...
4005c3:         48 8b 45 f8             mov    -0x8(%rbp),%rax
4005c7:         be 28 00 00 00          mov    $0x28,%esi
4005cc:         48 89 c7                mov    %rax,%rdi
4005cf:         e8 22 04 00 00          callq  4009f6
...    :        ...                     ...
4006e9:         55                      push   %rbp
4006ea:         48 89 e5                mov    %rsp,%rbp
4006ed:         48 83 ec 28             sub    $0x28,%rsp
4006f1:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
...    :        ...                     ...
4008d9:         c9                      leaveq
4008da:         c3                      retq
...    :        ...                     ...
4009f6:         55                      push   %rbp
4009f7:         48 89 e5                mov    %rsp,%rbp
4009fa:         48 83 ec 30             sub    $0x30,%rsp
4009fe:         48 89 7d d8             mov    %rdi,-0x28(%rbp)
400a02:         48 89 75 d0             mov    %rsi,-0x30(%rbp)
```



1) `f` returns `0x7fe047419010`
2) `g` takes `0x7fe047419010`

⇒ `f` and `g` are coupled

Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

- Using PIN
- Three pintools: `arity.cpp`, `type.cpp`, `couple.cpp`
- Instrumentation example:

```
if (INS_RegRContain(ins, REG_RDI) && !INS_RegWContain(ins, REG_RDI)) {
            INS_InsertCall(ins, IPOINT_BEFORE, (AFUNPTR) reg_access,
                        IARG_UINT32, REG_RDI,
                        IARG_ADDRINT, INS_Address(ins),
                        IARG_END);
```

Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

one execution

- `pin -t arity.so — mupdf ./pprew_slides.pdf`

- `pin -t type.so — emacs ./type.so`

- `pin -t couple.so — midori`

Key Idea

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

Overhead of instrumentation

	grep	tar	a2ps
#function	46	101	127
T1 (in s.)	0,80	0,99	0,80
T2 (in s.)	1,70	2,64	31,6
T3 (in s.)	1,06	1,79	13,2

Time execution depending on the instrumentation

- T1: no instrumentation
- T2: arity inference
- T3: type inference

Final words

Accuracy of arity inference

	midori	grep	mupdf	emacs
#function	4094	51	526	591
accuracy (%)	95,8	95,6	98,7	92,4

Accuracy of type inference

	midori	grep	mupdf	emacs
#function	4094	51	526	591
accuracy (%)	96,2	100	92,5	90,4

Additional materials

- Paper: `Lightweight heuristics to retrieve parameter associations from binaries`

- Scat implementation: https://github.com/Frky/scat

- Documentation to come

Do you have

questions ?