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

How much can we learn in a single execution?

Motivation?

Function prototypes | arity

arity
type of parameters

arity

type of parameters

Function coupling

arity

type of parameters

Function coupling

 $malloc \longrightarrow free$

arity

type of parameters

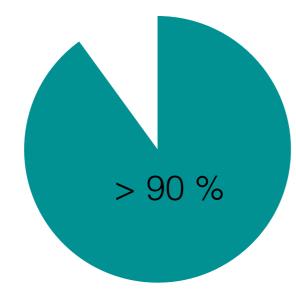
Function coupling

malloc \longrightarrow free fopen \longrightarrow fclose

Accuracy:

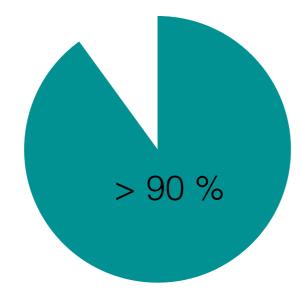
Accuracy:



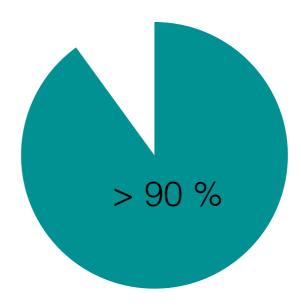


Accuracy:

Function arity

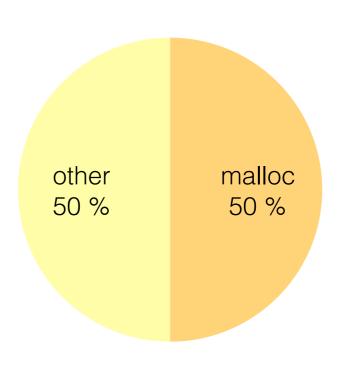


Parameter types



Coupling (on MuPDF):

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



Lightweight

Lightweight dynamic

Lightweight dynamic instrumentation

Lightweight dynamic instrumentation to recover

Lightweight dynamic instrumentation to recover function prototypes

Lightweight dynamic instrumentation to recover function prototypes and

Lightweight dynamic instrumentation to recover function prototypes and coupling

Lightweight dynamic instrumentation to recover function prototypes and coupling from

Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries

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

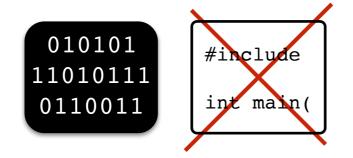
Lightweight dynamic instrumentation to recover function prototypes and coupling from binaries in one execution

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

binary program

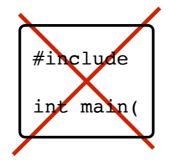
010101 11010111 0110011

- binary program
- no source code



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







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

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

Intuition:

Functions that are semantically linked

- malloc \longrightarrow free
- fopen \longrightarrow fclose

Definition. p-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 .

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

- 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

Registers read before written are parameters.

```
75 a9
401e55:
                                                401e00
                                         jne
               90
401e57:
                                        nop
401e58:
               с9
                                         leaveq
              с3
401e59:
                                        retq
               55
401e5a:
                                        push
                                                %rbp
              48 89 e5
                                                %rsp,%rbp
401e5b:
                                        mov
              48 83 ec 30
401e5e:
                                                $0x30,%rsp
                                         sub
               48 89 7d d8
                                                %rdi,-0x28(%rbp)
401e62:
                                        mov
               48 8b 45 d8
401e66:
                                                -0x28(%rbp), %rax
                                        mov
               48 8b 80 b0 27 00 00
                                                0x27b0(%rax),%rax
401e6a:
                                        mov
              eb 2e
                                                401eba
401e8a:
                                         jmp
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401e8c:
                                        mov
               8b 40 20
                                                0x20(%rax),%eax
401e90:
                                        mov
                                                %eax,%eax
401e93:
               85 c0
                                        test
               75 17
401e95:
                                                401eae
                                         ine
               48 8b 55 d8
401e97:
                                                -0x28(%rbp),%rex
                                        mov
                                                -0x20(%rbp),%rcx
               48 8d 4d e0
401e9b:
                                         lea
 ... :
                                         •••
              •••
              48 89 45 f8
                                                %rax, -0x8(%rbp)
401eb6:
                                        mov
               48 83 7d f8 00
                                                $0x0,-0x8(%rbp)
401eba:
                                         cmpq
401ebf:
               75 cb
                                         jne
                                                401e8c
401ec1:
               90
                                        nop
401ec2:
               с9
                                         leaveq
              c3
401ec3:
                                        retq
              55
401ec4:
                                                %rbp
                                        push
               48 89 e5
                                                %rsp,%rbp
401ec5:
                                        mov
                                         •••
              48 89 7d f8
                                                %rdi,-0x8(%rbp)
401fcb:
                                        mov
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401fcf:
                                        mov
                                                %rax,%rdi
               48 89 c7
401fd3:
                                        mov
               e8 7f fe ff ff
401fd6:
                                                401e5a
                                         callq
               48 8b 45 f8
                                                -0x8(%rbp),%rax
401fdb:
                                        mov
                                                0x27b0(%rax),%rax
               48 8b 80 b0 27 00 00
401fdf:
                                        mov
```

function

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

function

401e55: 401e57: 401e58: 401e59:	75 90 c9 c3	a9						jne nop leaveq retq	401e00
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: 401e8c: 401e90: 401e93: 401e95: 401e97: 401e9b: : 401eb6: 401eba: 401ec1:	48 8b 85 75 48 48 48 75 90	40 c0 17 8b 8d	45 20 55 4d 45 7d	d8 e0 f8	00			ing mov mov test jne mov lea in mov cmpq jne nop	401eba -0x8(%rbp),%rax 0x20(%rax),%eax %eax,%eax 401eae -0x28(%rbp),%rex -0x20(%rbp),%rcx %rax,-0x8(%rbp) \$0x0,-0x8(%rbp) 401e8c
401ec2: 401ec3:	c9							leaveq retq	
401ec4:	55							push	%rbp
401ec5:		89	e5					mov	%rsp,%rbp
: 401fcb:	 10	00	7d	ŧο					 %ndi (0,0(%nhn)
4011cb: 401fcf:			7a 45					mov mov	<pre>%rdi,-0x8(%rbp) -0x8(%rbp),%rax</pre>
401fd3:		89		TO				mov	%rax,%rdi
401fd6:			fe	ff	ff			callq	401e5a
401fdb:	48		45					mov	-0x8(%rbp),%rax
401fdf:			80		27	00	00	mov	0x27b0(%rax),%rax

function

- (at least) one parameter

401e55: 401e57: 401e58: 401e59:	75 90 c9 c3	a9	_					jne nop leaveq retq	401e00
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	<pre>%rdi,-0x28(%rbp)</pre>
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	8ъ	45	f8				mov	-0x8(%rbp),%rax
401e90:	8b	40	20					mov	0x20(%rax),%eax
401e93:		c0						test	%eax,%eax
401e95:		17						jne	401eae
401e97:			55					mov	-0x28(%rbp),%rex
401e9b:	48	8d	4d	e0				lea	-0x20(%rbp),%rcx
:	•••							•••	
401eb6:			45					mov	%rax,-0x8(%rbp)
401eba:			7d	f8	00			cmpq	\$0x0,-0x8(%rbp)
401ebf:		cb						jne	401e8c
401ec1:	90							nop	
401ec2:	c9							leaveq	
401ec3:	c3							retq	0. 1
401ec4:	55	00	_					push	%rbp
401ec5:	48	89	e 5					mov	%rsp,%rbp
:		00	7 1					•••	···
401fcb:			7d					mov	%rdi,-0x8(%rbp)
401fcf:			45	İ8				mov	-0x8(%rbp),%rax
401fd3:	48	89						mov	%rax,%rdi
401fd6:	e8		fe		ΪΪ			callq	401e5a
401fdb:	48	8b	45	f8	27	00	00	mov	-0x8(%rbp),%rax
401fdf:	48	מא	80	שם	2/	90	00	mov	0x27b0(%rax),%rax

function

- (at least) one parameter

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

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

```
75 a9
401e55:
                                                401e00
                                         ine
               90
401e57:
                                        nop
401e58:
               с9
                                         leaveq
401e59:
               c3
                                         reta
               55
401e5a:
                                         push
                                                %rbp
              48 89 e5
                                                %rsp,%rbp
401e5b:
                                        mov
              48 83 ec 30
401e5e:
                                                $0x30,%rsp
                                         sub
                                                                                              INT or ADDR?
              48 89 7d d8
                                                %rdi,-0x28(%rbp)
401e62:
                                        mov
401e66:
               48 8b 45 d8
                                                -0x28(%rbp), %rax
                                        mov
              48 8b 80 b0 27 00 00
401e6a:
                                                0x27b0(%rax),%rax
                                        mov
              eb 2e
                                                401eba
401e8a:
                                         jmp
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401e8c:
                                        mov
                                                0x20(%rax),%eax
               8b 40 20
401e90:
                                        mov
401e93:
               85 c0
                                                %eax,%eax
                                        test
               75 17
401e95:
                                                401eae
                                         ine
              48 8b 55 d8
401e97:
                                                -0x28(%rbp),%rex
                                        mov
                                                -0\times20(\%rbp),\%rcx
               48 8d 4d e0
401e9b:
                                         lea
 ... :
                                         •••
              •••
              48 89 45 f8
                                                %rax, -0x8(%rbp)
401eb6:
                                        mov
               48 83 7d f8 00
                                                $0x0,-0x8(%rbp)
401eba:
                                         cmpq
401ebf:
               75 cb
                                                401e8c
                                         jne
401ec1:
               90
                                        nop
401ec2:
               с9
                                         leaveq
401ec3:
               c3
                                        retq
              55
401ec4:
                                                %rbp
                                        push
              48 89 e5
                                                %rsp,%rbp
401ec5:
                                        mov
                                         •••
              48 89 7d f8
                                                %rdi,-0x8(%rbp)
401fcb:
                                        mov
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401fcf:
                                        mov
                                                %rax,%rdi
               48 89 c7
401fd3:
                                        mov
               e8 7f fe ff ff
401fd6:
                                                401e5a
                                        callq
               48 8b 45 f8
                                                -0x8(%rbp),%rax
401fdb:
                                        mov
                                                0x27b0(%rax),%rax
               48 8b 80 b0 27 00 00
401fdf:
                                        mov
```

#2 — type

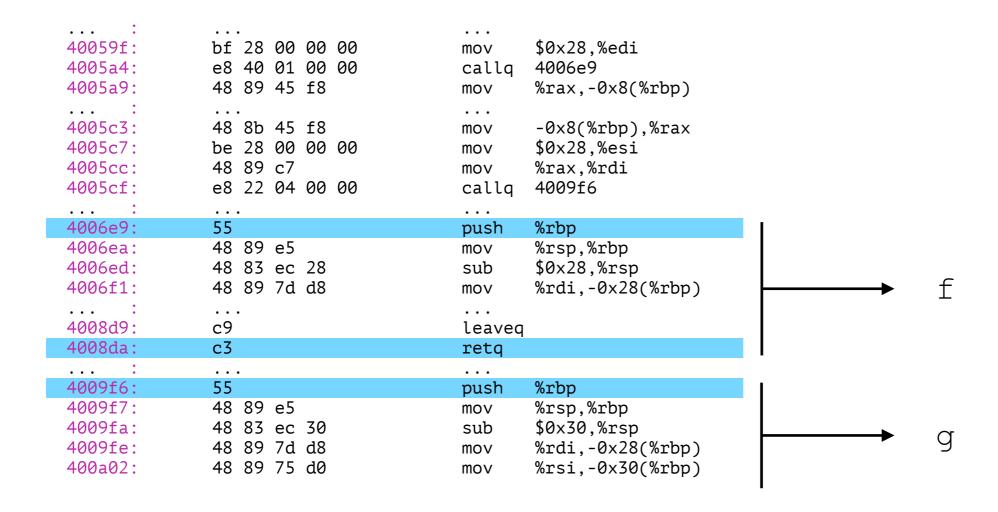
401e55: 401e57: 401e58: 401e59:	75 90 c9 c3	a9						jne nop leaveq retq	401e00
401e5a:	55							push	%rbp
401e5b:	48	89	e5					mov	%rsp,%rbp
401e5e:	48	83	ec	30				sub	\$0x30,%rsp
401e62:	48	89	7d	d8				mov	<pre>%rdi,-0x28(%rbp)</pre>
401e66:	48	8ъ	45	d8				mov	-0x28(%rbp),%rax
401e6a:	48	8b	80	b0	27	00	00	mov	0x27b0(%rax),%rax
: 401e8a: 401e8c: 401e90: 401e93: 401e95: 401e97: 401e9b: : 401eb6: 401eba: 401ebf: 401ec1:	 eb 48 8b 85 75 48 48 48 75 90	2e 8b 40 c0 17 8b 8d 89 83	45 20 55 4d 45	f8 d8 e0				index index	401eba -0x8(%rbp),%rax 0x20(%rax),%eax %eax,%eax 401eae -0x28(%rbp),%rex -0x20(%rbp),%rcx %rax,-0x8(%rbp) \$0x0,-0x8(%rbp) 401e8c
401ec2:	c9							leaveq	
401ec3:	c3 55							retq	% nhn
401ec4: 401ec5:		89	۵5					push mov	%rbp %rsp,%rbp
		0)	63						<i>ж</i> 15р, <i>ж</i> 10р
 401fcb:	 48	89	7d	f৪				mov	 %rdi,-0x8(%rbp)
401fcf:			45					mov	-0x8(%rbp),%rax
401fd3:		89		_ ~				mov	%rax,%rdi
401fd6:				ff	ff			callq	401e5a
401fdb:				f8				mov '	-0x8(%rbp),%rax
401fdf:	48	8ъ	80	b0	27	00	00	mov	0x27b0(%rax),%rax

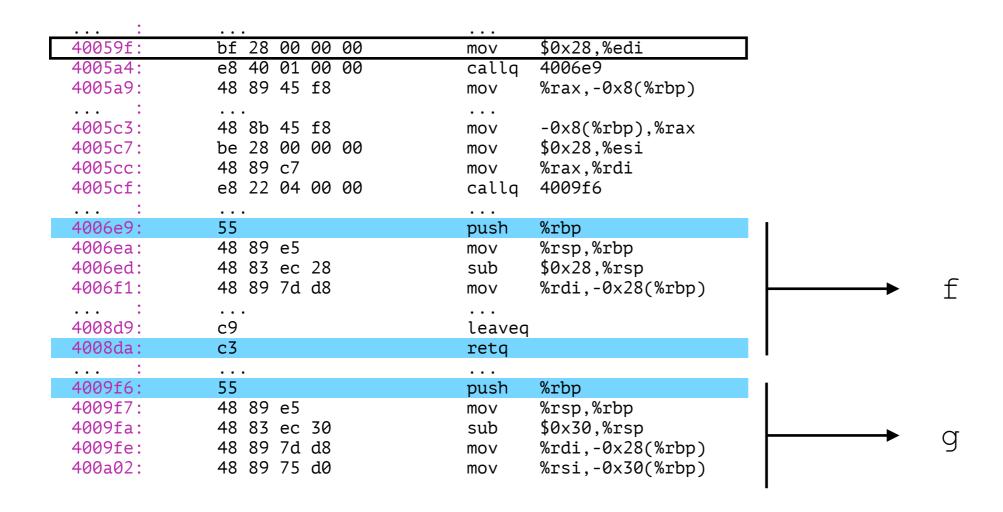
```
75 a9
401e55:
                                                401e00
                                         jne
               90
401e57:
                                         nop
401e58:
               с9
                                         leaveq
401e59:
               c3
                                         reta
               55
401e5a:
                                         push
                                                %rbp
              48 89 e5
                                                %rsp,%rbp
401e5b:
                                        mov
              48 83 ec 30
401e5e:
                                                $0x30,%rsp
                                         sub
                                                %rdi,-0x28(%rbp)
401e62:
               48 89 7d d8
                                         mov
              48 8b 45 d8
401e66:
                                                -0x28(%rbp), %rax
                                         mov
                                                                                              ADDR
              48 8b 80 b0 27 00 00
401e6a:
                                                0x27b0(%rax), %rax -
                                         mov
 ...
              eb 2e
                                                401eba
401e8a:
                                         jmp
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401e8c:
                                         mov
               8b 40 20
                                                0x20(%rax),%eax
401e90:
                                         mov
                                                %eax,%eax
401e93:
               85 c0
                                         test
401e95:
               75 17
                                                401eae
                                         ine
401e97:
               48 8b 55 d8
                                                -0x28(%rbp),%rex
                                         mov
                                                -0\times20(\%rbp),\%rcx
               48 8d 4d e0
401e9b:
                                         lea
 ... :
                                         •••
              •••
              48 89 45 f8
                                                %rax, -0x8(%rbp)
401eb6:
                                         mov
              48 83 7d f8 00
                                                $0x0,-0x8(%rbp)
401eba:
                                         cmpq
401ebf:
               75 cb
                                         jne
                                                401e8c
401ec1:
               90
                                         nop
401ec2:
               с9
                                         leaveq
              c3
401ec3:
                                         retq
              55
401ec4:
                                                %rbp
                                         push
              48 89 e5
                                                %rsp,%rbp
401ec5:
                                         mov
                                         •••
              48 89 7d f8
                                                %rdi,-0x8(%rbp)
401fcb:
                                         mov
              48 8b 45 f8
                                                -0x8(%rbp),%rax
401fcf:
                                         mov
                                                %rax,%rdi
               48 89 c7
401fd3:
                                         mov
               e8 7f fe ff ff
401fd6:
                                                401e5a
                                         callq
               48 8b 45 f8
                                                -0x8(%rbp),%rax
401fdb:
                                        mov
              48 8b 80 b0 27 00 00
                                                0x27b0(%rax),%rax
401fdf:
                                         mov
```

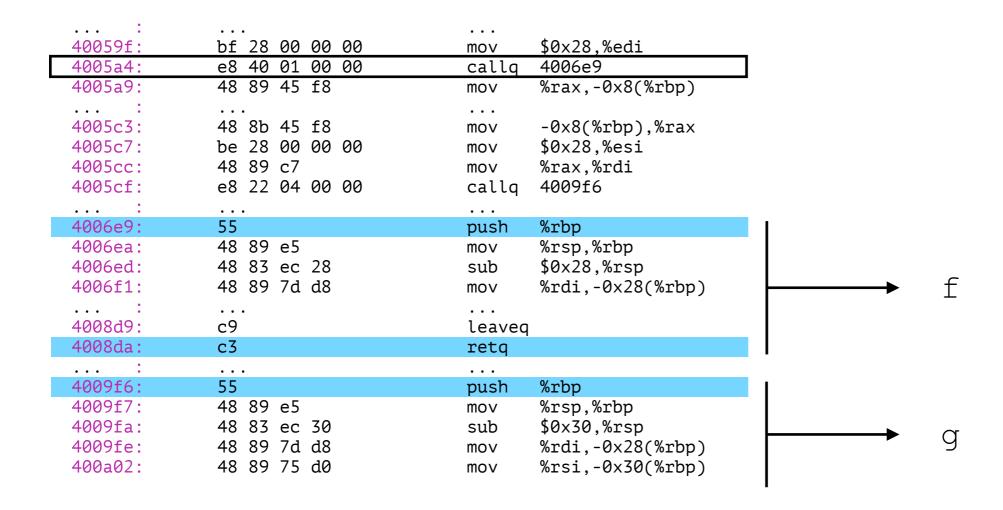
#2 — type

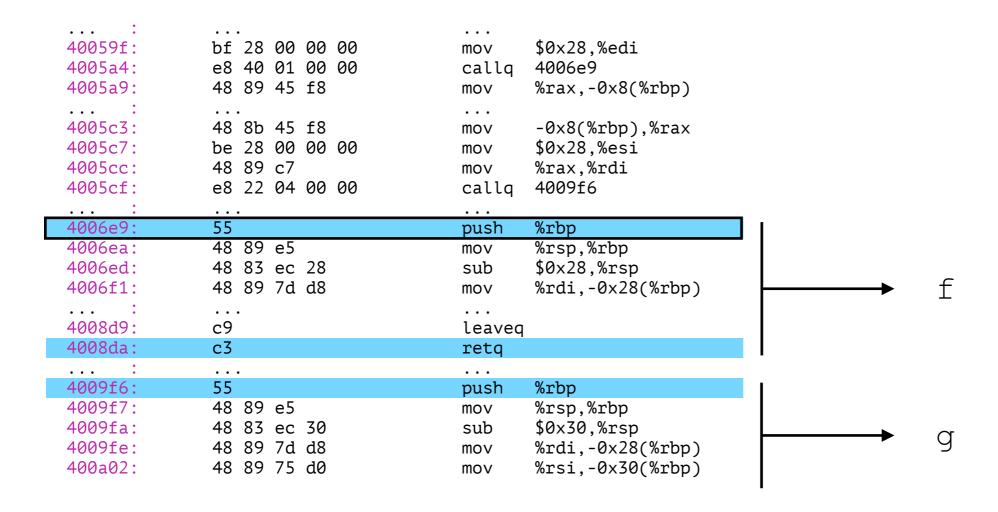
401e55: 401e57: 401e58: 401e59:	75 a9 90 c9 c3			jne nop leaveq retq	401e00
401e5a:	55			push	%rbp
401e5b:	48 89	e5		mov	%rsp,%rbp
401e5e:		ec 30		sub	\$0x30,%rsp
401e62:		7d d8		mov	<pre>%rdi,-0x28(%rbp)</pre>
401e66:		45 d8		mov	-0x28(%rbp),%rax
401e6a:	48 8b	80 ъ0 27	00 00	mov	<pre>0x27b0(%rax),%rax</pre>
: 401e8a: 401e8c: 401e90: 401e93: 401e95: 401e97: 401e9b: : 401eb6: 401eba: 401ebf:	 eb 2e 48 8b 8b 40 85 c0 75 17 48 8b 48 8d 48 89 48 83 75 cb	45 f8 20 55 d8 4d e0 45 f8 7d f8 00		jmp mov mov test jne mov lea mov cmpq jne	401eba -0x8(%rbp),%rax 0x20(%rax),%eax %eax,%eax 401eae -0x28(%rbp),%rex -0x20(%rbp),%rcx %rax,-0x8(%rbp) \$0x0,-0x8(%rbp) 401e8c
401ec1: 401ec2:	90 c9			nop	
401ec3:	c3			leaveq retq	
401ec4:	55			push	%rbp
401ec5:	48 89	e5		mov	%rsp,%rbp
401fcb: 401fcf: 401fd3: 401fd6: 401fdb: 401fdf:	48 8b 48 89 e8 7f 48 8b	7d f8 45 f8 c7 fe ff ff 45 f8 80 b0 27	00 00	mov mov mov callq mov mov	<pre>" "rdi,-0x8(%rbp) -0x8(%rbp),%rax %rax,%rdi 401e5a -0x8(%rbp),%rax 0x27b0(%rax),%rax</pre>

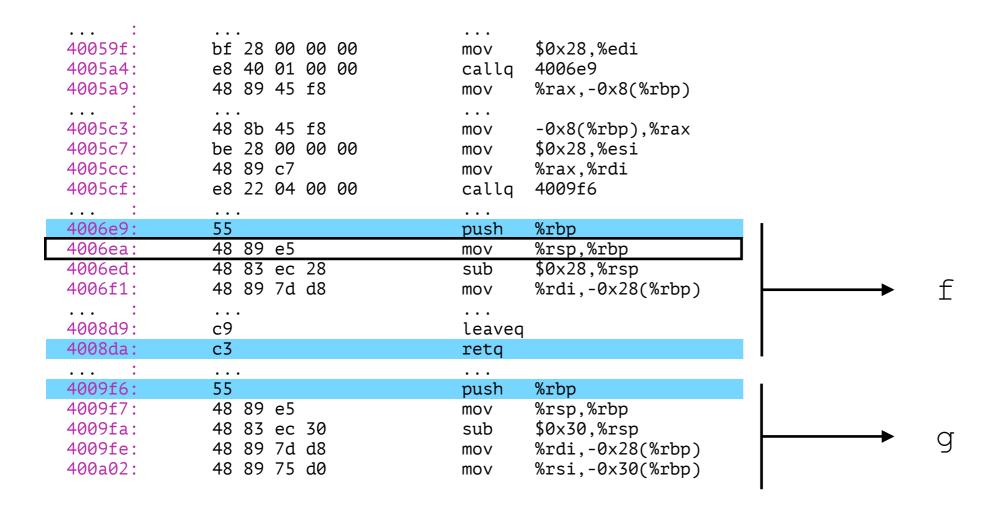
A shared value between a ret and a param. includes a coupling.

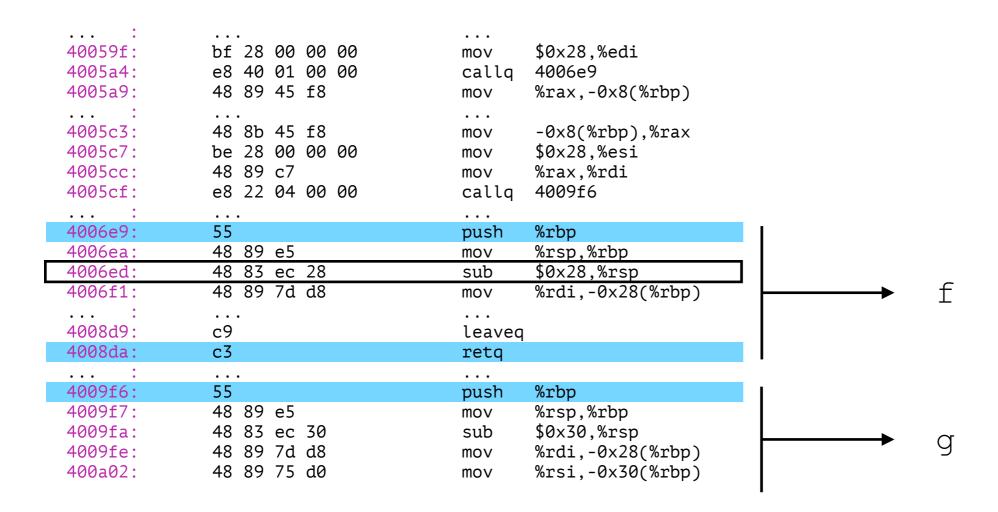


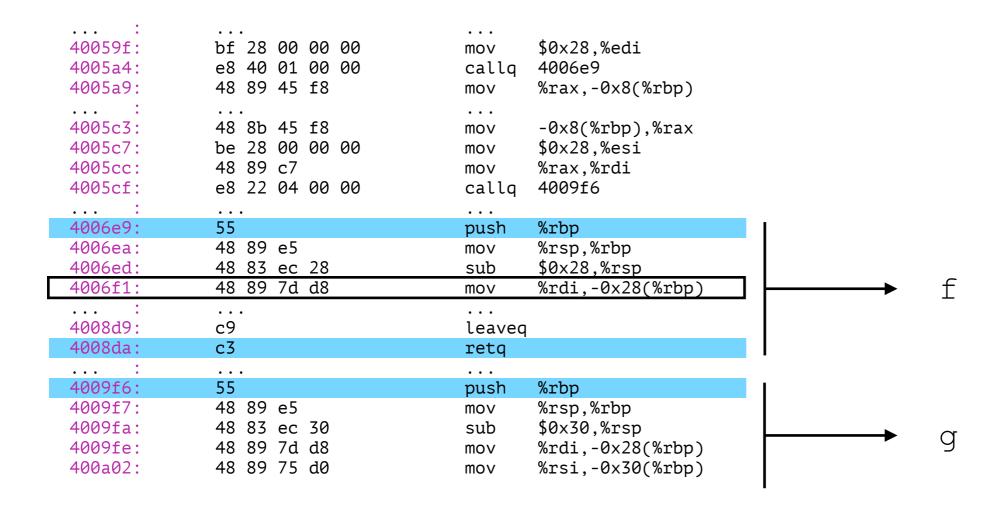


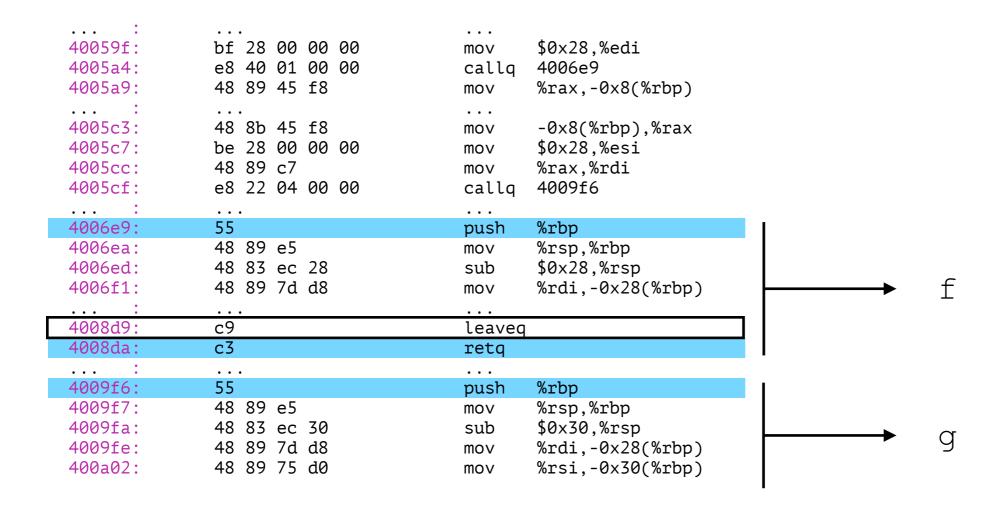


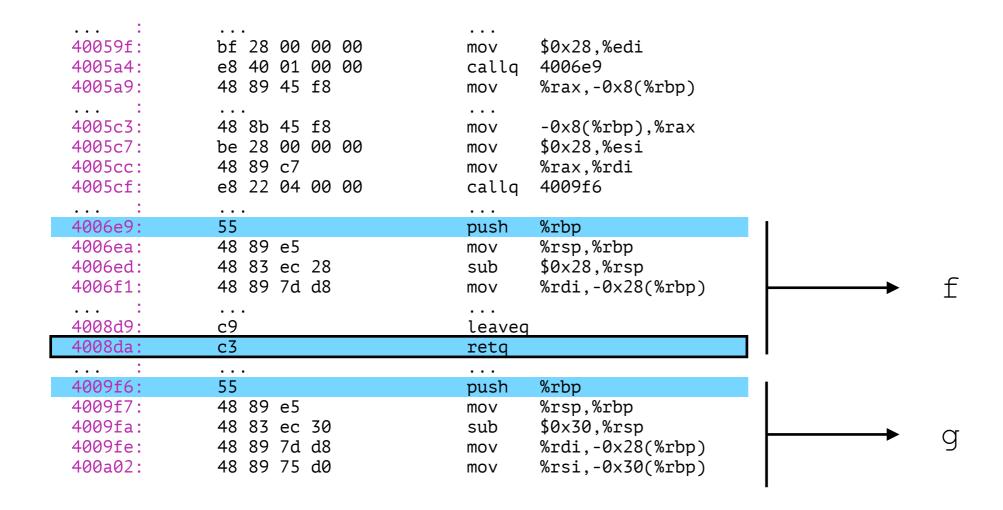


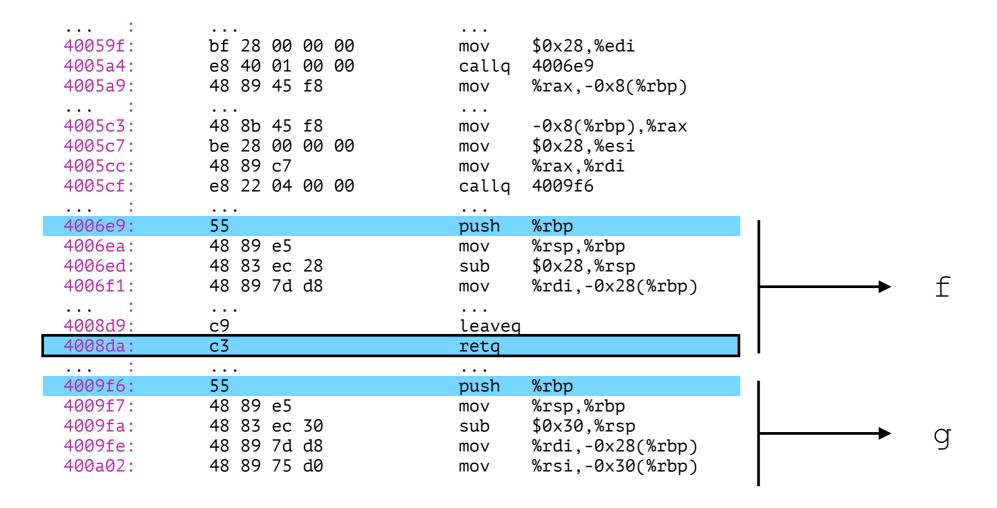


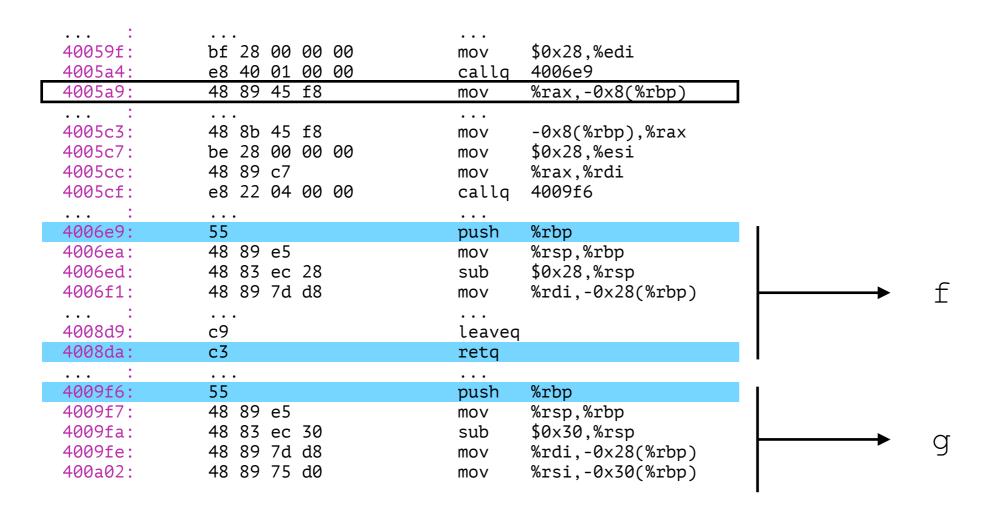


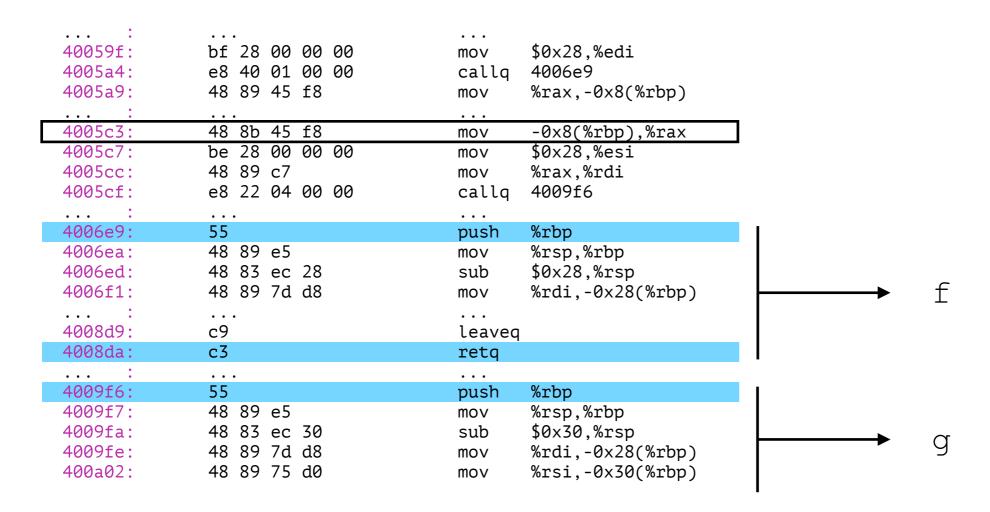


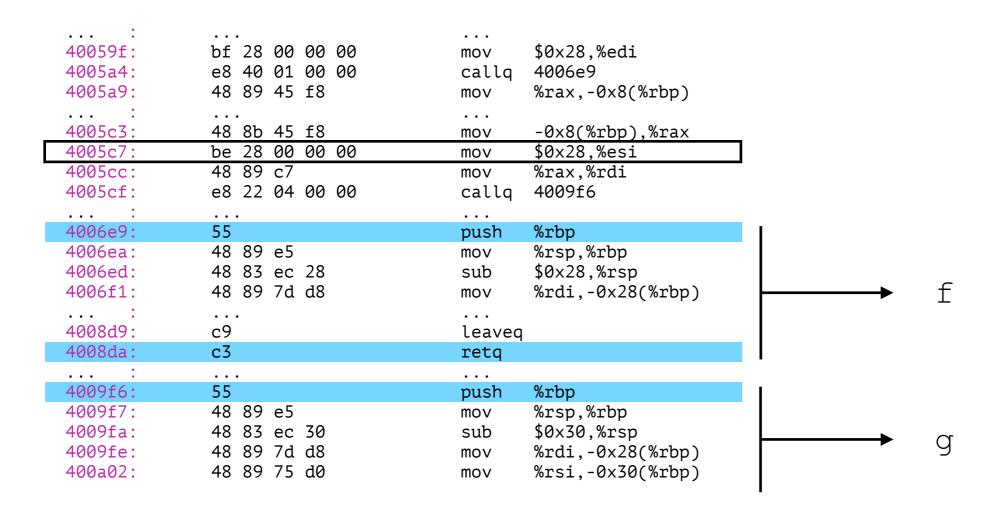


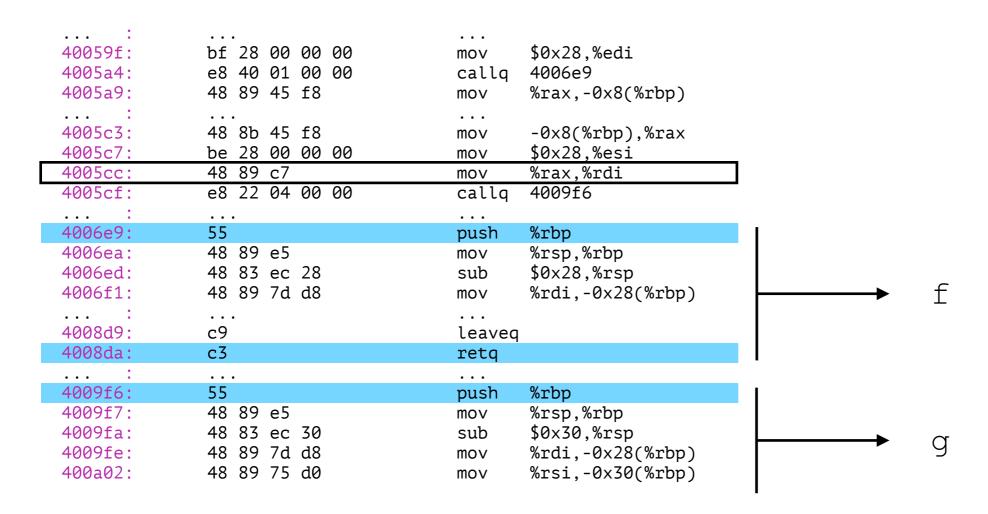


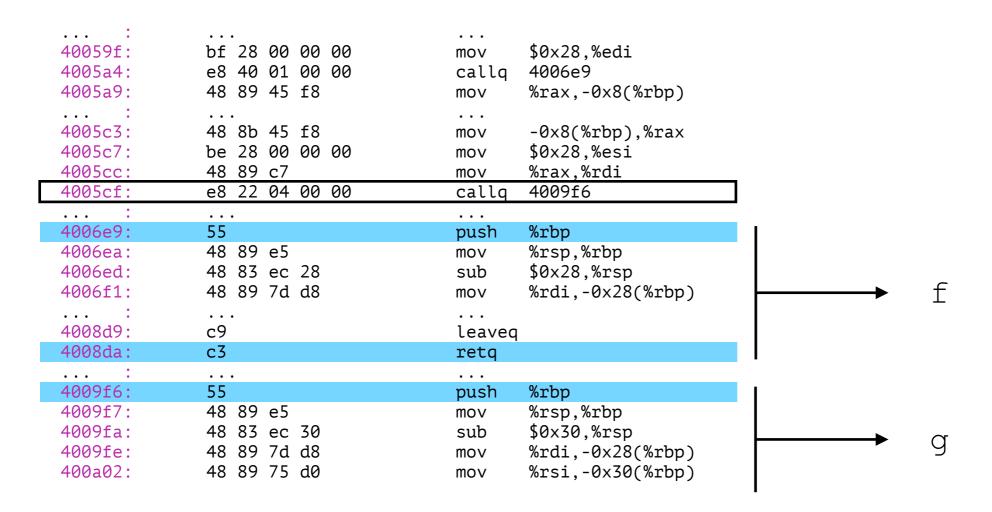


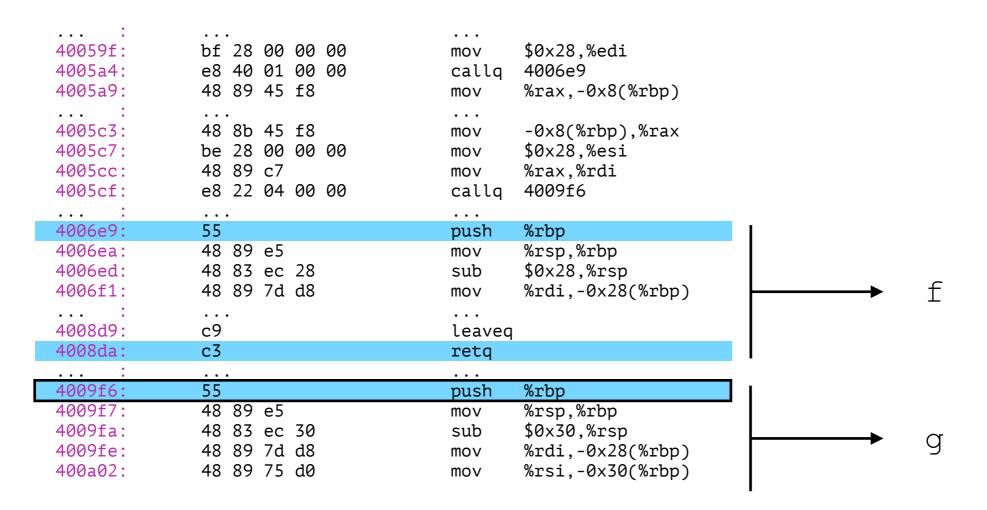


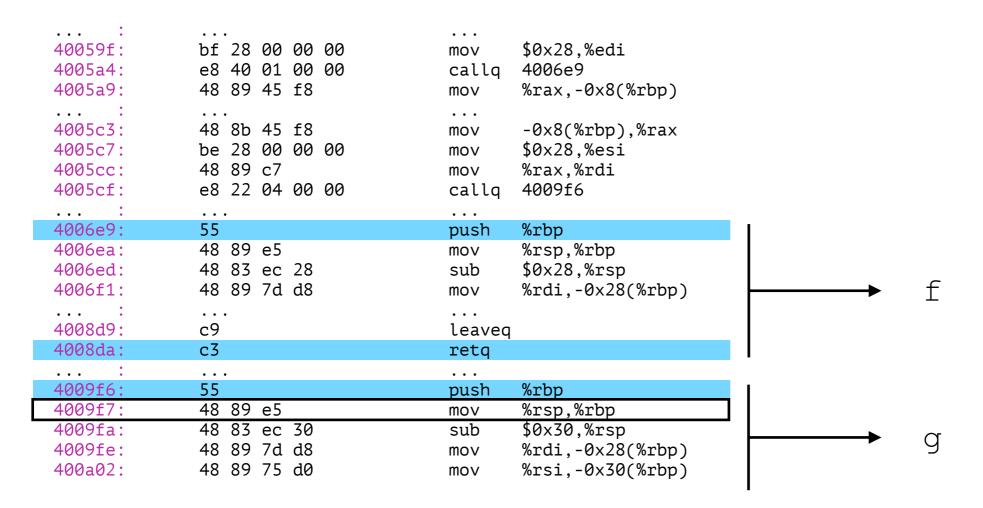


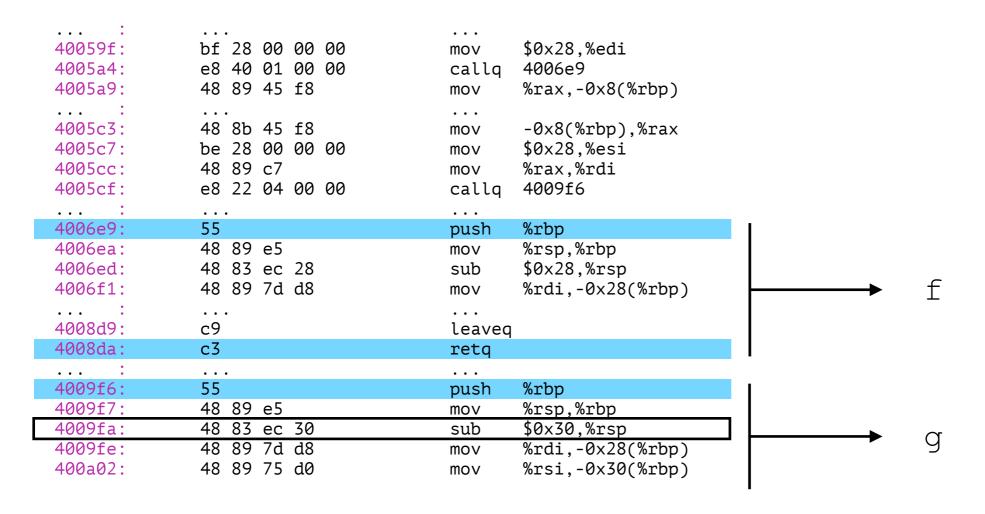


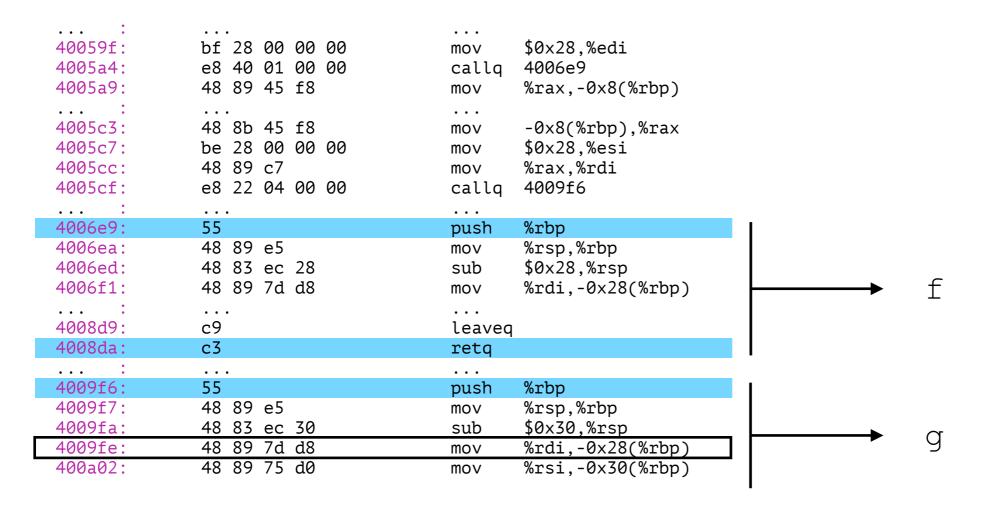






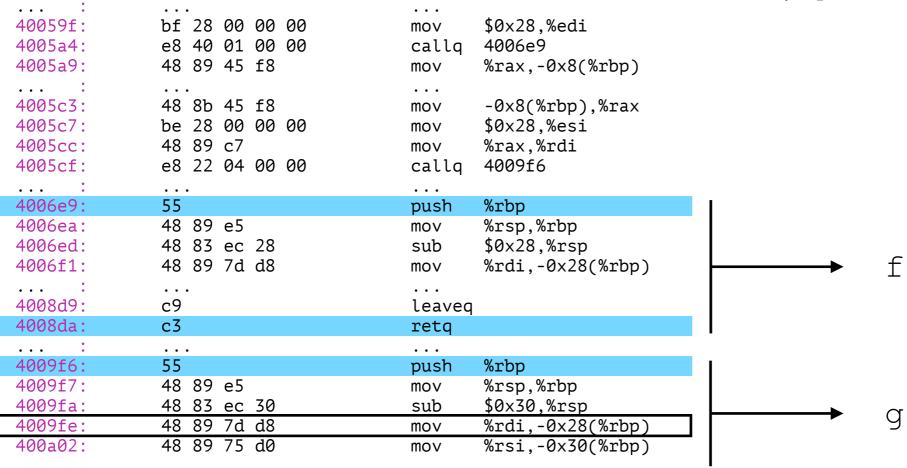




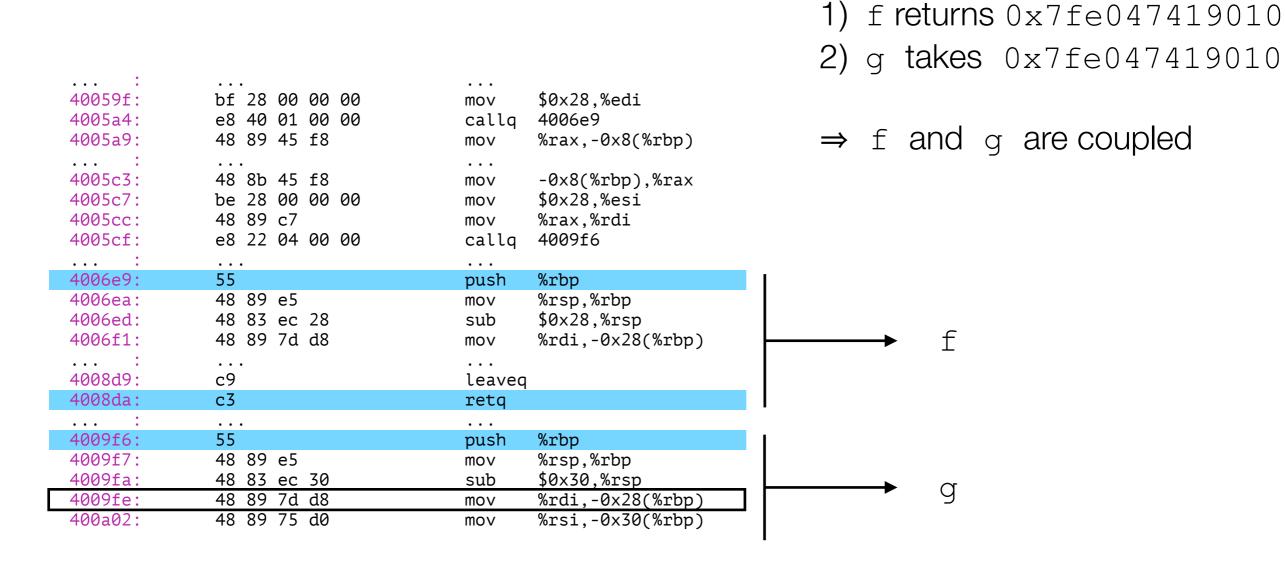


#3 — coupling

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



#3 — coupling



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:

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

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

• pin -t type.so - emacs ./type.so

• pin -t couple.so - midori

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

- Paper: Lightweight heuristics to retrieve parameter associations from binaries
- Scat implementation: https://github.com/Frky/scat
- Documentation to come

Do you have

questions?