	sub_314623 eax, eax	
	short loc_31306D	
	<pre>[ebp+arg_0], ebx</pre>	
	short loc_313066 eax, [ebp+var 70]	
	eax, [ebp+var_84]	
	short loc_313066	
Exploiting	eax, [ebp+var_84] esi	
push		
	eax	
Out-of-Order-Exect		
	eax, eax short loc 31306D	
	<pre>eax, [ebp+arg_0]</pre>	
Processor Side Channels to Enab	eax	
FIDLESSUI JIUE CHAIMEIS LU LITAL		
$O_{\text{rescale}} \setminus / \setminus / O_{\text{rescale}} = \nabla_{\text{rescale}} = \nabla_{\text{rescale}} + \nabla_{\text{rescale}} $	[ebp+arg_4]	
Cross VM Code Execution	edi sub 314623	
jz cmp	<pre>short loc_31306D [ebp+arg_0], esi</pre>	
	short loc_31308F	
loc_313066:		
	sub_31411B	
loc_31306D:		
Sophia D'Antoine	sub_3140F3 eax, eax	
Soonia D Ani.oine	short loc_31307D	
call	sub_3140F3	
	short loc_31308C	
REcon 2015		
10c_31307D:		
call and	sub_3140F3 eax, OFFFFh	
loc_31308C:		
100_313080;		

The Cloud

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
push	esi



client-side encryption, server-side computation

beginning with encrypted search

LEARN MORE

	loc_31307D:	loc_31307D:		
			sub_3140F3	
		and	eax, OFFFFh	
06/19/2015	Exploiting Out-of-Order-Execution			2/46
	loc_31308C:			
			[ebp+var 4], ea	

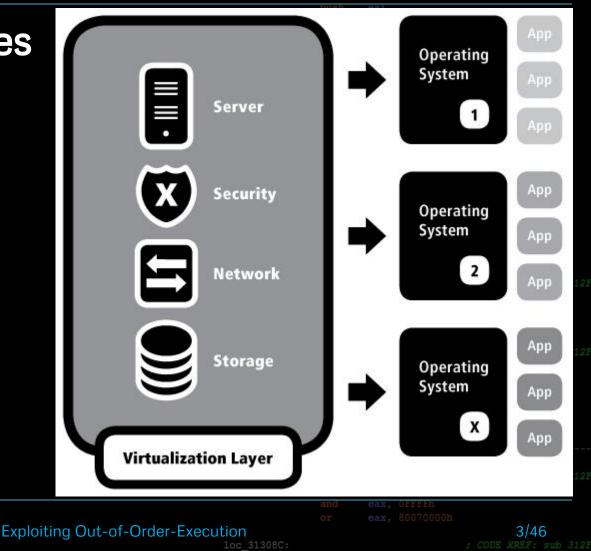
Cloud Computing (IaaS)

```
edi
sub_314623
eax, eax
short loc_31306D
[ebp+arg_0], ebx
short loc_313066
eax, [ebp+var_70
eax, [ebp+var_84
short loc_313066
eax, [ebp+var_84
esi
```

- Virtual instances
- Hypervisors

Dynamic allocation

=> Reduces cost



Everyone's Happy

06/19/2015

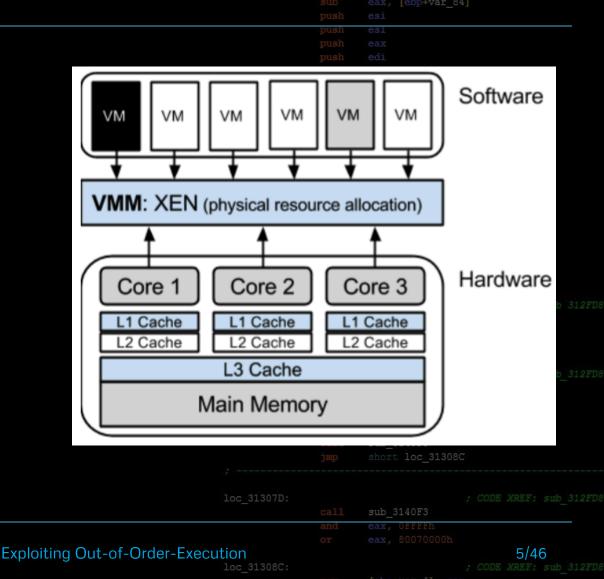
p	ush	edi		
		sub_314623		
		short loc_3130	6D	
		[ebp+arg_0], el		
		short loc_3130		
		eax, [ebp+var_]		
		eax, [ebp+var_6	84]	
		short loc_31300		
		eax, [ebp+var_6	84]	
p	ush	esi		
A Same game	18 Cal	ebp+arg_0], ea		
		ub_31486A		
		ax, eax		
	1 M. 1	hort loc_3130	6D	
	Contraction of	si		
	1.0	ax, [ebp+arg_(
64.		ax		
and the second	1. 18 1 1	si, 1D0h		
and a start	4	si		
		ebp+arg_4]		
1.19		di		
	1	ub_314623		
and the	Sec. 1	ax, eax		
	s (* 26	hort loc_3130		
1.0	7. 11	ebp+arg_0], es		
1. C. 11		hort loc_31308	BF	
a trans	1. 1.			
1.14		ne se		
		Dh		
	Sec. 2	ub_31411B		
and the	No all			
1.22				
1				
Sec. 3.		ub_3140F3		
1.8.8	1.1	ax, eax		
9		hort loc_3130		
 omaca 		Sub_3140F3		
		short loc_31308		
7D:				
		sub_3140F3		



Problems with the Cloud

Security issues with cloud computing

- Sensitive data stored remotely
- Vulnerable host
- Untrusted host
- Co-located with foreign VM's



Physical co-locatio to side chann vulnerabilities



		sub_314623
		short loc_31306D
		[ebp+arg_0], ebx
		short loc_313066
		eax, [ebp+var_70]
		eax, [ebp+var_84]
	jb	short loc_313065
	dur	eaos
	ush	Edus
	-ush	
	push	
-	mc 7	[ebp+arg_0], eax
	23 11	sub_31486A
	test	
	jz	short loc_31306D
		eax, [ebp+arg_0]
	mov	
	push	
		[ebp+arg_4]
		edi
		sub_314623
		eax, eax
		short loc_31306D
		[ebp+arg_0], esi
		short loc_31308F
		sub_31411B
		sub_3140F3
		short loc_31307D
		sub_3140F3
		short loc_31308C
		sub_3140F3
	and	eav OFFFF

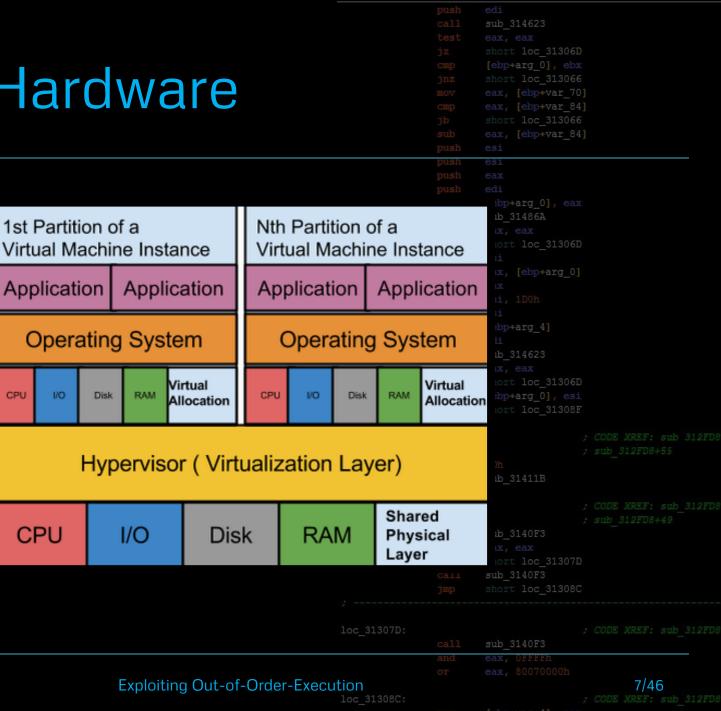
06/19/2015

Exploiting Out-of-Order-Execution

6/46

Cloud Hardware

CPU



06/19/2015

Universal Vulnerabilities

edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84] esi esi

1) Translation between physical and virtual hardware based on need

2) Allocation causes contention

ahi	sub_31486A
	short loc_31306D
	eax, [ebp+arg_0]
	[ebp+arg_4]
	sub_314623
	short loc_31306D
	[ebp+arg_0], esi
	short loc_31308F

3) Private VM activities not opaque to sub_31412 co-residents

		0400_011010			
		short loc_31307D			
		sub_3140F3			
		short loc_31308C			
		sub_3140F3			
	and	eax, OFFFFh			
			8/4	6	
8C:					
		[ebp+var 4], eax			

Overview

- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the processor
- 4. Importance of memory models
- 5. Design of an Out-of-Order-Execut cha
- Der 6.

06/19/2015

7. Cor

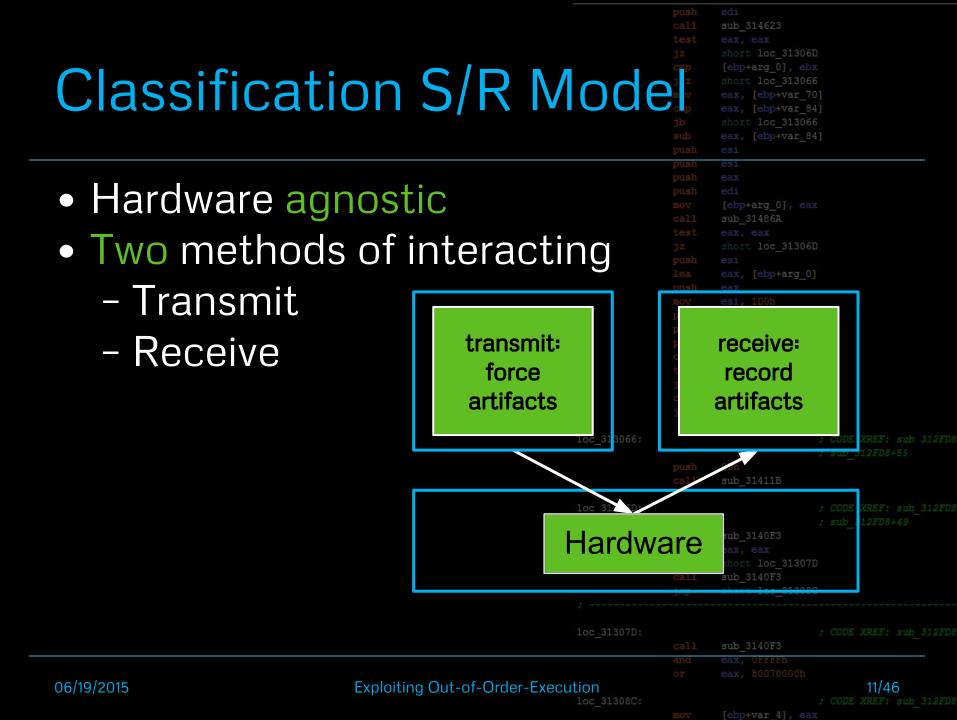
	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	eax, [ebp+var_70]
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
Ish	esi
	edx edi [ebp+arg_0], eax sub_31486A
	sub_31486A
	short loc_31306D
	<pre>eax, [ebp+arg_0] eax</pre>
	[ebp+arg_4] edi
	sub_314623
st	
T FF I	$\bigcirc 100 \text{ c}_{31306D}$
	elprang 01, esi

Innei		loc_313066:			
no				0Dh sub_31411B	
nclusion		loc_31306D:		sub_3140F3 eax, eax short loc_31307I	
				sub_3140F3 short loc_313080	
		loc_31307D:		sub_3140F3	
	Exploiting Out-of-Order-Exe	cution 10c 31308C;	and or	eax, OFFFFh eax, 80070000h	9/46 ; CODE XREF: sub 312FD8
		100_010000.		[ebp+var_4], eax	

Side Channel Attack

"In cryptography, a **sidechannel** attack is any attack based on information gained from the physical implementation of a cryptosystem"

sub 314623 **Cloud Computing** Hardware side channel [ebp+arg_4] Cross virtua machine Information gained through recordable changes in the system 10/46



Possible Exploits

• Receive (exfiltrate)

- 1. crypto key theft
- 2. process monitoring
- 3. environment keying
- 4. broadcast signal

• Transmit (infiltrate)

- 1. DoS
- 2. co-residency

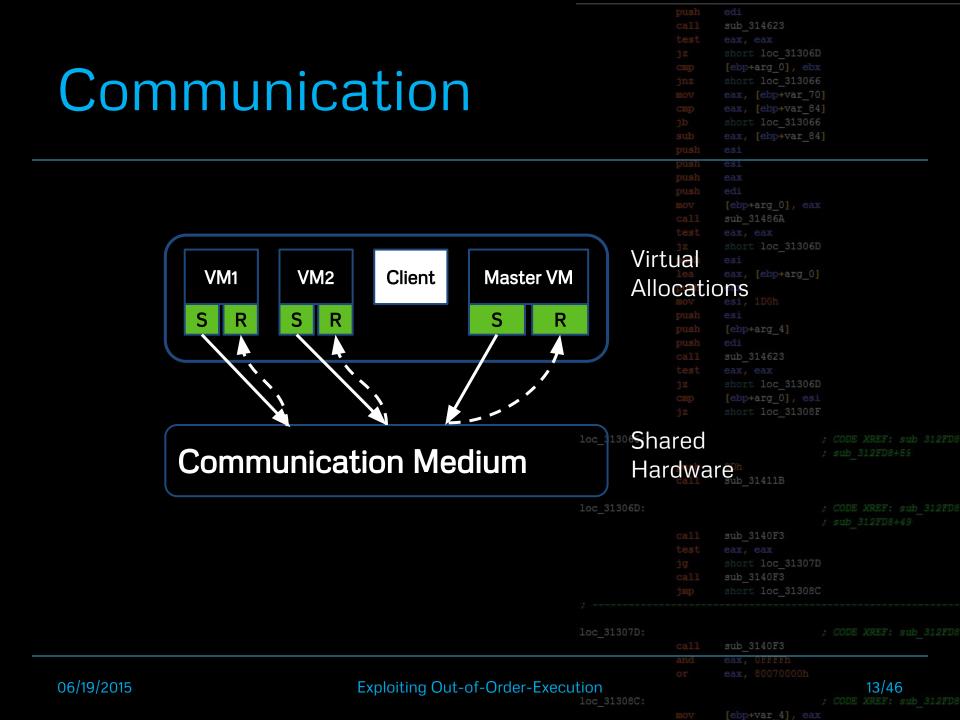
Transmit & Receive (network)

1. communication (C&C)

		sub_314623
		short loc_31306D
		[ebp+arg_0], ebx
		short loc_313066
		eax, [ebp+var_70]
		eax, [ebp+var_84]
		short loc_313066
		eax, [ebp+var_84]
	push	esi
	push	esi
		[ebp+arg_0], eax
		sub_31486A
		eax, eax
		short loc_31306D
		eax, [ebp+arg_0]
		[ebp+arg_4]
		edi
		sub_314623
		eax, eax
		short loc_31306D
		[ebp+arg_0], esi
		short loc_31308F
313066:		
		sub 31411B
31306D:		
		sub_3140F3
		short loc_31307D
		sub 3140F3
		short loc_31308C
_31307D:		
		sub_3140F3
	and	eax, OFFFFh
		12/16

Exploiting Out-of-Order-Execution

; CODE XREF: sub_312



Cache Side Channel Example (1, ebc 313060 boot loc 313060 [3]

Flush+Reload targets the L3 Cache

- Receiving Mechanism (Adversary)
 Flushes & queries
- Transmitting Mechanism (Victim)
 - Accesses same L3 line
- Leaked GnuPG Private Key

sophia.re/cache.pdf

		sub_314623		
		short loc_31306D		
		[ebp+arg_0], ebx		
		abort 100 312056		[3]
	- IV	e x [e pr -v 0]]	[3]
		elax Log v 4		
		short loc_313066		
		eax, [ebp+var_84		
	push	esi		
_	push	eax		
noh	push	Tior		
acł		ebpHarry_0], eax		
	call	sub_31486A		
		eax, eax		
	JZ	short loc_31306D		
ary	ousn	esi		
лц	Lea	eax, [ebp+arg_0]		
0	pusn			
		esi Jebatarg (1		
		[ebp+arg_4] edi		
		sub_314623		
m)		eax, eax		
111/		short loc 31306E		
		[ebp+arg_0], esi		
		short loc_31308E		
066:				
		sub_31411B		
06D:				
		sub_3140F3		
		short loc_31307D		
		sub_3140F3		
		short loc_313080		
07D:				
	call	sub_3140F3		
				11/16
				14/46

Pipeline vs	Cache Chan	test jz cmp jo jb sub push	<pre>short loc_313066 eax, [ebp+var_84] esi</pre>	
9	annel cache misses, etc. amplifies in a crowded	push push mov call test jz push lea push lea push call test jz cmp jz	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0] eax esi 1D0h UC_tap+arg_4] edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi short loc_31308F	
	loc_313066:			
	loc_31306D: ;			
06/19/2015	Loc_31307D: Exploiting Out-of-Order-Execution Loc_31308C:	call and or	sub_3140F3 eax, 0FFFFh eax, 80070000h	CODE XREF: sub_312FD8

Overview

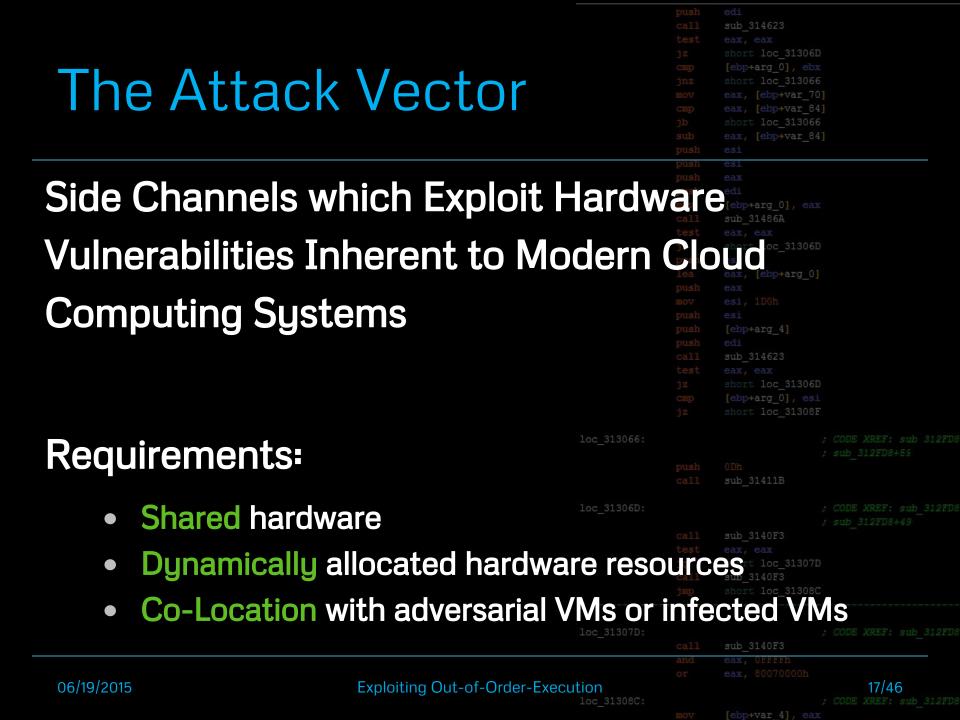
- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the pipeline
- 4. Importance of memory models
- 5. Design of an Out-of-Order-Execution channel
- 6. Demo

06/19/2015

7. Conclusion

	edi sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	eax, [ebp+var_70]
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
ish	es1
	eax edi [ebp+arg_0], eax
	sub 31486A
	eax, eax short loc_31306D
	eax, [ebp+arg_0]
	[ebp+arg_4] edi sub_314623
	sub_314623
est	

		oc_313066:			
				sub_31411B	
		c_31306D:			
DN				sub_3140F3	
				short loc_31307D	
				sub_3140F3	
				short loc_313080	
		c_31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
	Exploiting Out-of-Order-Execution	on			16/46
		c_31308C:			
				[ebp+var 4], eax	



Pipeline Side Channel

	sub_314623
	short loc_31306D
	<pre>[ebp+arg_0], ebx</pre>
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	<pre>eax, [ebp+var_84]</pre>
	short loc_313066
	<pre>eax, [ebp+var_84]</pre>
ush	esi

ub 314623

sub 3140F3

We chose to target the processor as the hardware medium. [ebp+arg 0]

- => CPU's pipeline => System artifacts queried dynamically
 - Instruction order

06/19/20

Results from instruction sets

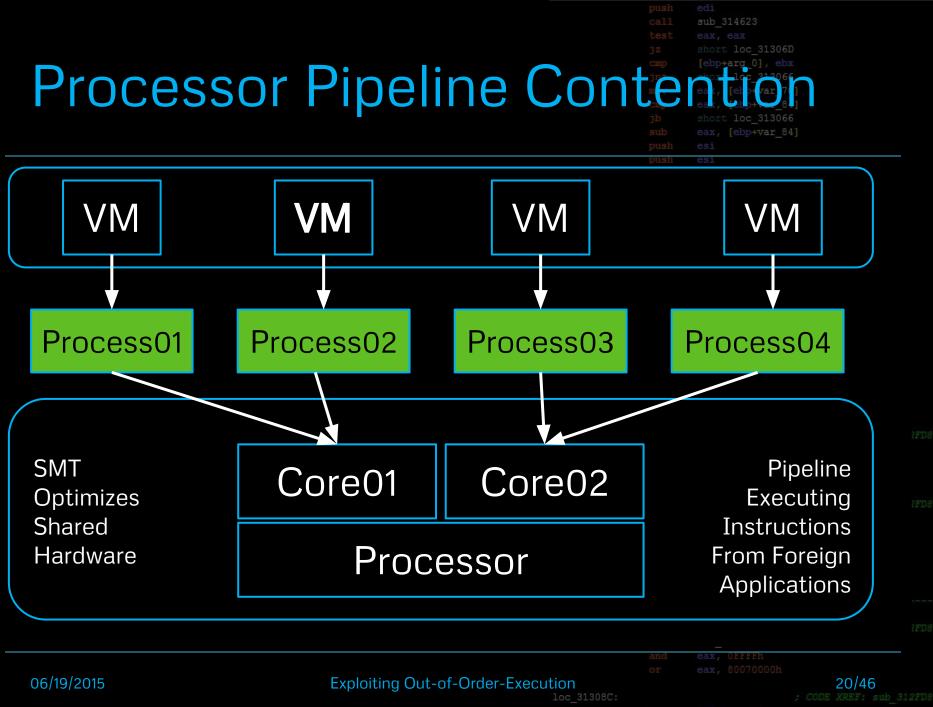
			short loc_31308	
	loc 31307D:			
			sub_3140F3	
		and	eax, OFFFFh	
15	Exploiting Out-of-Order-Execution			18/46
	loc_31308C:			
			[ebp+var 4], ea	

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	eax, [ebp+var_70]
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	eax, [ebp+arg_0]
push	[ebp+arg_4]

Out-of-Order-Execution 314623

			<pre>eax, eax short loc_31306 [ebp+arg_0], esi short loc_31308F</pre>	
	loc_313066:			
			sub_31411B	
	loc 31306D:			
			sub_3140F3	
			short loc_31307D	
			sub_3140F3	
			short loc_313080	
	loc_31307D:			
			sub_3140F3	
		and	eax, OFFFFh	
Exploiting Out-of-Order-Execu				19/46
	loc_31308C:			
			Tehn+var 41, eax	

06/19/2015



\square	EI)		\square
		$V \square$	\square
			-

sub_314623
short loc_31306D
[ebp+arg_0], ebx
short loc_313066
eax, [ebp+var_70]
<pre>eax, [ebp+var_84]</pre>
short loc_313066
eax, [ebp+var_84]
[ebp+arg_0], eax
sub_31486A
short loc_31306D
eax, [ebp+arg_0]
[ebp+arg_4]
sub_314623
short loc_31306D
<pre>[ebp+arg_0], esi</pre>
short loc_31308F

	21/46
and	eax, OFFFTh
	sub_3140F3
	short loc_31308C
	sub_3140F3
	short loc_31307D
	sub_3140F3
	sub_31411B

06/19/2015

Record Out of Order	Exe jz Exe jz cm jz c cm jz c c c c c c c c c c c c c c c c c c	<pre>11 sub_314623 st eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 c.k, [ebp+var_84] short loc_313066 c.k, [ebp+var_84]</pre>	[6]
	pus pus mov	sh esi sh eax sh edi v [ebp+arg_0], eax	

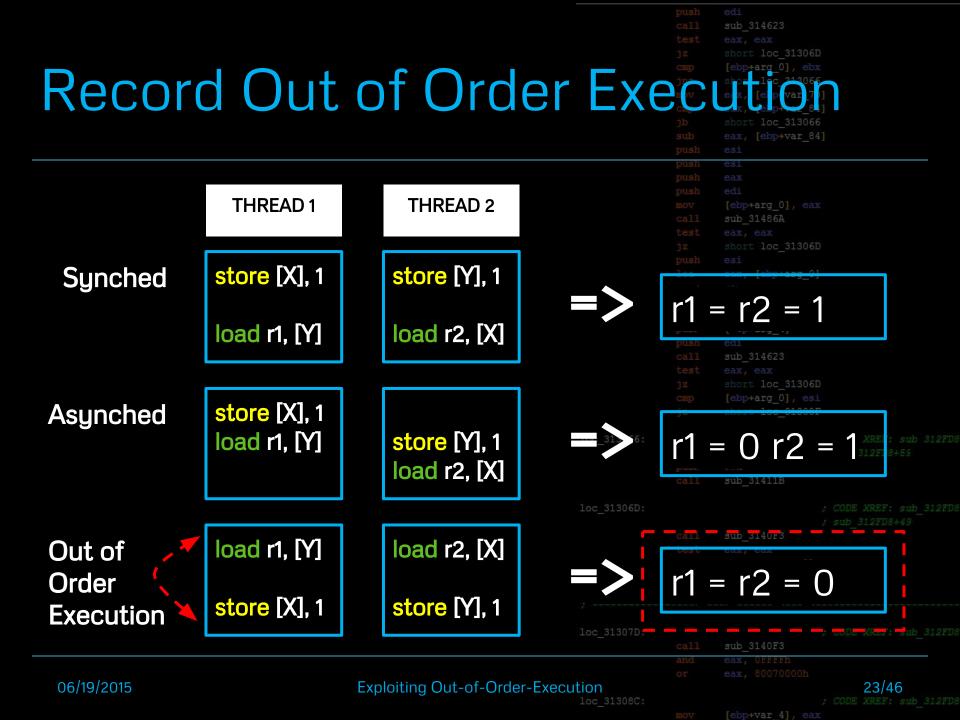
8.2.3.4 Loads May Be Reordered with Earlier Stores to Different Locations

The Intel-64 memory-ordering model allows a load to be reordered with an earlier store to a different location. However, loads are not reordered with stores to the same location.

The fact that a load may be reordered with an earlier store to a different location is illustrated by the following example:

Example 8-3. Loads May be Reordered with Older Stores

Processor 0			Process	or 1		
mov [_x], 1	mov [_y]	, 1				
mov r1, [_y]	mov r2, [_x]				24.077
Initially $x = y = 0$						312FD
r1 = 0 and $r2 = 0$ is allowed						
		loc_31306D:			; CODE XREF:	sub 312FI
				sub_3140F3 eax, eax short loc_31307 sub_3140F3 short loc_31308		
) loc_31307D:		sub_3140F3		
06/19/2015	Exploiting Out-of-Order-Exploiting	ecution	and or	eax, OFFFFh eax, 80070000h	22/46 ; CODE XREF:	



Record Out of Order Executive and the short loc_313060

Exploiting Out-of-Order-Ex

int X,Y,count_OoOE;

....initialize semaphores Sema1 & Sema2...
pthread_t thread1, thread2;
pthread_create(&threadN, NULL, threadNFunc, NULL);

for (int iterations = 1; ; iterations++)

X,Y = 0;

sem_post(beginSema1 & beginSema2);
sem_wait(endSema1 & endSema2);

```
if (r1 == 0 && r2 == 0)
```

```
count_OoOE ++;
```

El	
	short loc 313066
	eax, [ebp+var_84]
push	esi
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	eax, [ebp+arg_0]
	[ebp+arg_4]
	sub_314623
	short loc_31306D
	[ebp+arg_0], esi
	short loc_31308F

	Vera	ages n	natter
loc_31306D:		sub_3140F3 eax, eax short loc_31307 sub_3140F3 short loc_31308	
loc_31307D:		sub_3140F3	
	and	eax, OFFFFh	
ecution loc_31308C:		eax, 80070000h	24/46 ; CODE XREF: sub_312FD8

TRANSMITTER

06/19/2015

sub_314623
short loc_31306D
[ebp+arg_0], ebx
short loc_313066
<pre>eax, [ebp+var_70]</pre>
<pre>eax, [ebp+var_84]</pre>
short loc_313066
<pre>eax, [ebp+var_84]</pre>
[ebp+arg_0], eax
sub_31486A
short loc_31306D
eax, [ebp+arg_0]
[ebp+arg_4]
sub_314623
short loc_31306D
<pre>[ebp+arg_0], esi</pre>
short loc_31308F

	loc_313066;			
	loc_31306D:			
	loc_31307D:		sub_3140F3	
Exploiting Out-of-Order-Execu	tion loc_31308C:	and or mov	eax, Offffh eax, 80070000h	25/46 ; CODE XREF: sub_312FD8

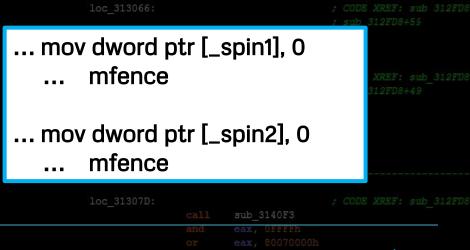
Force Out of Order Executive short loc_313060

Memory Fences

Mfence:

- x86 instruction full memory barrier prevents memory reordering of any kind
- order of 100 cycles per operation

	$\begin{bmatrix} ax & ebp \\ y & r_1 \end{bmatrix} \begin{bmatrix} 0 \\ y & r_1 \end{bmatrix}$
ib	short loc 313066
	eax, [ebp+var_84]
oush	esi
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	<pre>eax, [ebp+arg_0]</pre>
	[ebp+arg_4]
	sub_314623
	eax, eax
	short loc_31306D
	<pre>[ebp+arg_0], esi</pre>
	short loc_31308F



26/46

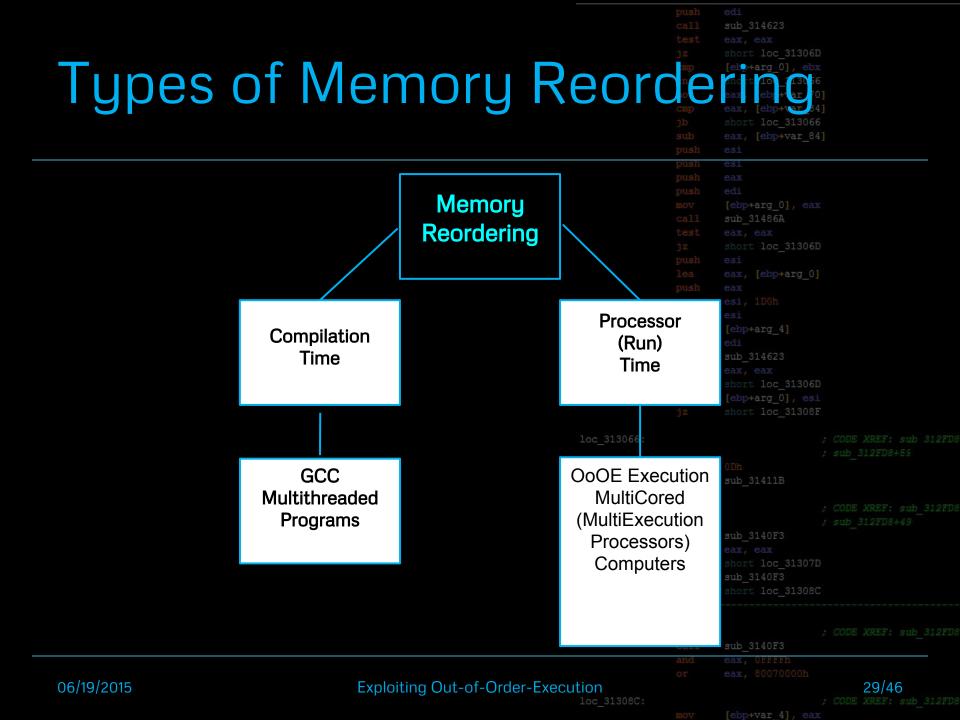
Force Out	c of Order	Exe	push call test jz cmp mo jb sub push	eax, [ebp+var_84] esi
THE PIPELINE				esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0]
NOP Store	[X], 1 mfence	Load r1, [X]	NOP
		loc_31306D:		<pre>sub_31411B</pre>
06/19/2014	Exploiting Out-of-Order-Exploiting Out-Order-Exploiting	loc_31307D: xecution loc_31308C:	call and or mov	; CODE XREF: sub_312FL sub_3140F3 eax, OFFFFh eax, 80070000h 27/46 ; CODE XREF: sub_312FL [ebp+var_4], eax

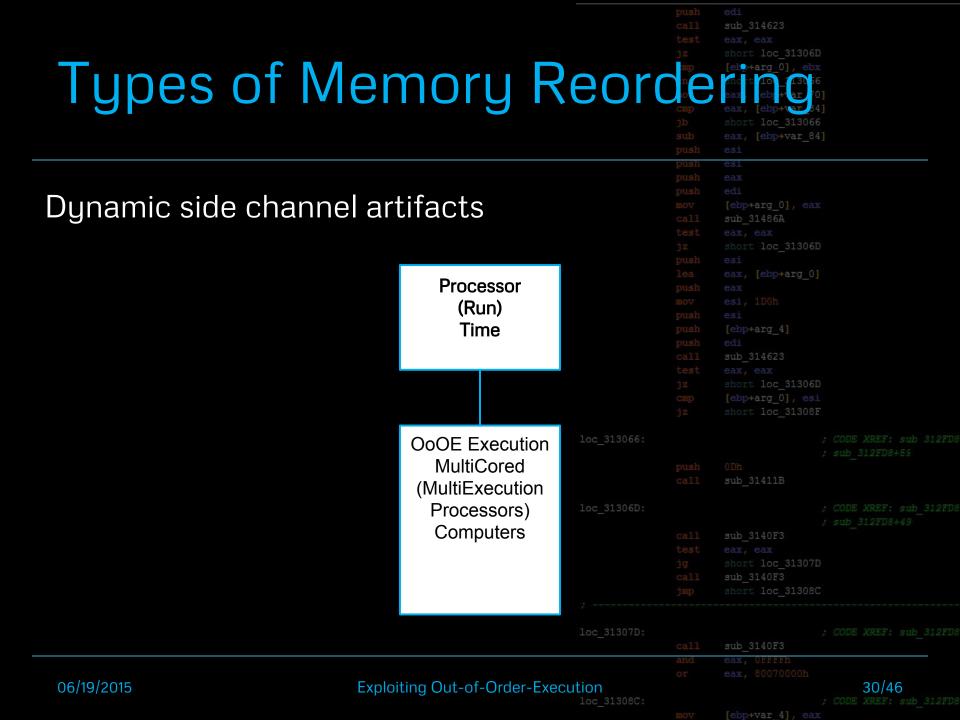
Overview

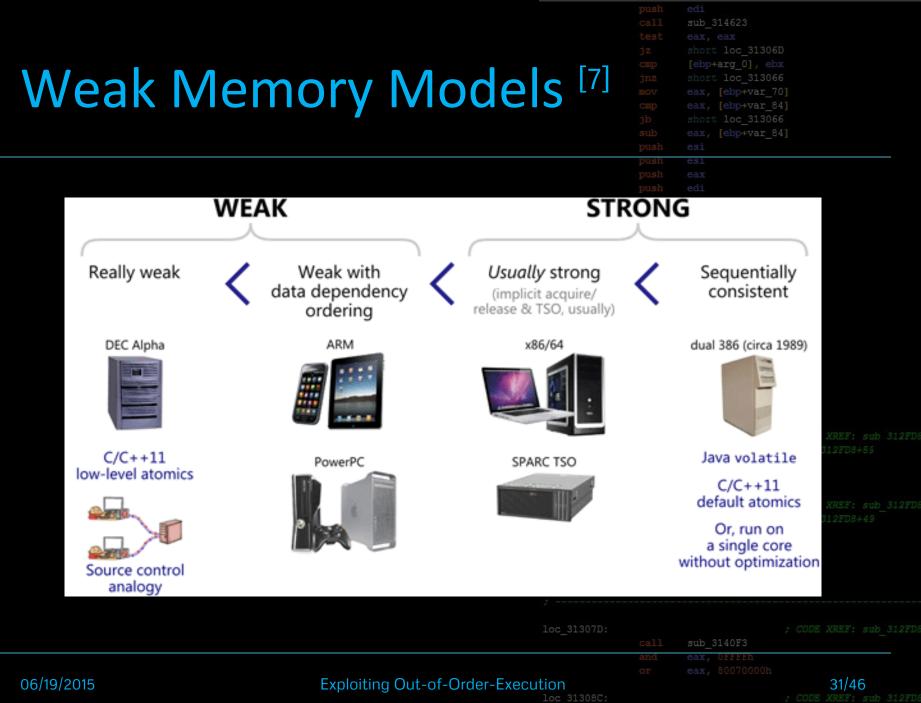
- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the processor
- 4. Importance of memory models
- 5. Design of an Out-of-Order-Execution
- 6. Demo
- 7. Conclusion

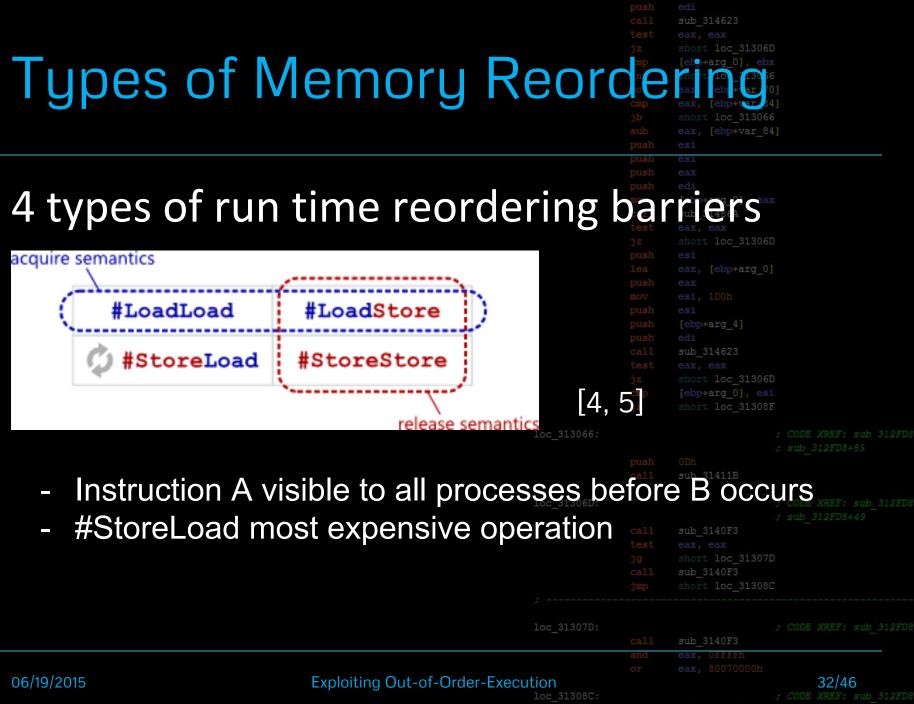
	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	<pre>eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax</pre>
	eax, [ebp+var_70]
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
oush	es1
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	<pre>eax, [ebp+arg_0]</pre>
	[ebp+arg_4]
	sub_314623
test	eax, eax
Z	loc 31306D

nnel		loc_313066:			
10				0Dh sub_31411B	
clusion		loc_31306D:		sub 3140F3	
CIUSIOII				eax, eax short loc_31307E sub_3140F3 short loc_31308C	
		loc_31307D:		sub_3140F3	
	Exploiting Out-of-Order-Exec		and or	eax, OFFFFh eax, 80070000h	28/46
		loc_31308C:		[ebp+var 4], eax	









Force Out	of Order E	xe	call test jz cmp mo mo jb sub push	sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx bort bbp vr_0] ebp vr_0 short loc_313066 eax, [ebp+var_84] esi	
Memory Barrie'Lock-free pression		nn S	push push mov call test jz ov/h	es1 eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi	
 multiprocess #StoreLoad 		nts r	mov push push call jz	eak esi, 1D0h esi [ebp+arg_4] edi 2 = 0 = 31306D	
		2_3130 66:		0Dh sub_31411B	
		c_31306D:		<pre>sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C</pre>	
	lo	c_31307D:	call and	sub_3140F3 eax, 0FFFFh	; CODE XREF: sub_312FD8
06/19/2015	Exploiting Out-of-Order-Executio)n c_31308C:			33/46 ; CODE XREF: sub_312FD8

Channel Transmitter (Victim

- Force Out-of-Order-Execution patterns
- Affect the order of stores and loads
- Time frame dependant
- x86: mfence

06/19/2015



Overview

- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the processor
- 4. Importance of memory models
- 5. Design of an Out of Order Execution channe
- 6. Demo
- 7. Conclusion

Exploiting Out-of-Order-Execution			35/46
Tural a it is a four law Fundament			
	and	eax, OFFFFh	
		sub_3140F3	
loc_31307D:			
		short loc_3130	
		sub_3140F3	
		short loc_3130	
		sub 3140F3	
loc 31306D:			
		sub_31411B	
loc_313066:			
out of Order Exec	Uab I (jz	short loc_3130	
lut at (Irdar Evac	vi iti i	nn ch	annol

10C_31308C:

[ebp+var 4], eax

sub 314623

Lab Model

Scheduler Xen hypervisor

- Popular commercial IaaS platforms
- Xeon Processors

06/19/2

Shared multi-core/ multi-processor hardware

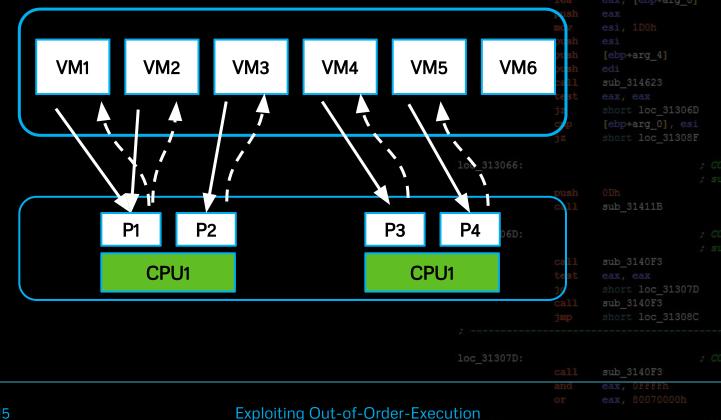
- 8 logical CPU's/ 4 cores
- 6 virtual machines (VM's)
- Parallel Processing/ Simultaneous Multi-Threading On (SMT)

sub 314623

; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C ; loc_31307D: ; CODE XREF: sub_312FD	015	Exploiting Out-of-Order-Execution	and or mov	eax, OFFFFh eax, 80070000h [ebp+var 4], eau	36/46 ; CODE XREF: sub_312FD
; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3		loc_31307D:	call	sub 3140F3	
		loc_31306D:		sub_3140F3 eax, eax short loc_313071 sub_3140F3	

Virtual Machines

• 6 Windows 7 VM's



loc 31308C:

, coos AN

37/46

06/19/2015

Virtual Machine S

VM1

R

١

S

			sub_314623	
			short loc_31306D	
			[ebp+arg_0], ebx	
S/R			short loc_313066	
			eax, [ebp+var_70]	
			eax, [ebp+var_84]	
			short loc_313066	
			eax, [ebp+var_84]	
		push	esi	
			esi	
			<pre>[ebp+arg_0], eax sub 31486A</pre>	
			eax, eax	
			short loc_31306D	
1/1/0				
VM2			eax, [ebp+arg 0]	
	-			
S R				
			[ebp+arg_4]	
	_			
			sub_314623	
			short loc_31306D	
			<pre>[ebp+arg_0], esi</pre>	
<u>v</u>			short loc_31308F	
	13066:			
P1				
			ODh	
			sub_31411B	
CPU1	1306D:			
0.01				
			sub_3140F3	
			eax, eax	
			short loc_31307D	
			sub_3140F3	
			short loc_31308C	
	loc_31307D:			
			sub_3140F3	
		and	eax, OFFFFh	
f-Order-Execut				38/46
	loc_31308C:			

06/19/2015

Exploiting Out-of-

Overview

- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the processor
- 4. Importance of memory models
- 5. Design of an Out-of-Order-Exec channel
- 6.

06/19/2015

7.

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc 313066
	eax, [ebp+var_70]
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var 84]
push	esi
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	eax, [ebp+arg_0]
	[ebp+arg_4]
	sub_314623
jz -	eax, eax Onag_0], esi
	[e]p-arg_0], esi
	short loc_31308F

Панны					
emo				sub_31411B	
		loc_31306D:			
onclusion				sub_3140F3 eax, eax short loc_31307I sub_3140F3 short loc_313080	
		; loc_31307D:			
			call	sub_3140F3	
				eax, OFFFFh	
	Exploiting Out-of-Order-Exe	ecution			39/46 ; CODE XREF: sub_312FD8
				[ebp+var_4], eax	

Demo Links

sophia.re/sender.py

sophia.re/receiver.py

		sub_314623	
		short loc_31306	
		<pre>[ebp+arg_0], eb</pre>	
		short loc_31306	
		eax, [ebp+var_7	
		eax, [ebp+var_8	4]
		short loc_31306	
		eax, [ebp+var_8	4]
	push	esi	
		[ebp+arg_0], ea	
		sub_31486A	
		short loc_31306	
		<pre>eax, [ebp+arg_0</pre>	
		[ebp+arg_4]	
		sub_314623	
		short loc_31306	
		<pre>[ebp+arg_0], es</pre>	
		short loc_31308	
_313066:			
		sub_31411B	
_31306D:			
		sub_3140F3	
		short loc_31307	
		sub_3140F3	
		short loc_31308	C
31307D:			
		sub 3140F3	
	and	eax, OFFFFh	
n la			40/46
313080			· CODE VEEK: sub 312FD8

06/19/2015

Overview

- 1. Introduction
- 2. Cloud exploitation techniques
- 3. Targeting the processor
- 4. Importance of memory models
- 5. Design of an Out-of-Order-Execution channel

loc 31308C:

[ebp+var 4], eax

Potential Channel Mitigati **Protected Resource Ownership** Isolating VM's ()Turn off hyperthreading Blacklisting resources for concurrent threads ()Downside: cloud benefits sub 3140F3 06/19/2015 Exploiting Out-of-Order-Execution 42/46

sub 314623

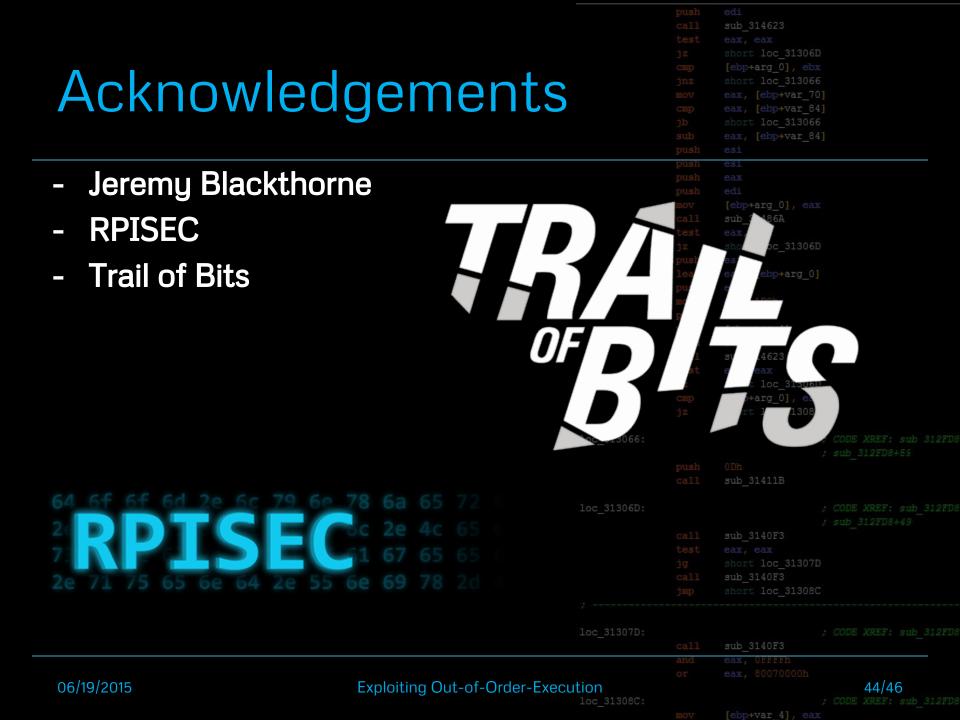
In Conclusion...

Contribution:

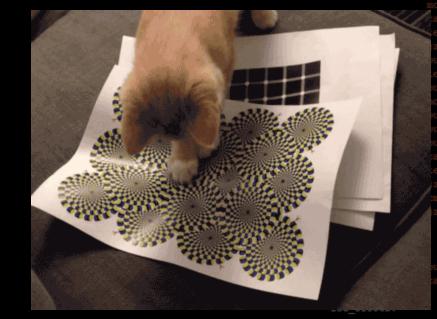
We demonstrate a novel Out of Order Execution side channel.

- **Dynamic** querying/ forcing method
- Application to cloud computing_313066
- Mitigation techniques

		sub_314623	
		short loc_31306D	
		[ebp+arg_0], ebx	
		short loc_313066	
		<pre>eax, [ebp+var_70]</pre>	
		eax, [ebp+var_84]	
		short loc_313066	
		eax, [ebp+var_84]	
	push	esi	
		[ebp+arg_0], eax	
		sub_31486A	
		short loc_31306D	
er		<pre>eax, [ebp+arg_0]</pre>	
		[ebp+arg_4]	
		sub 314623	
		short loc_31306D	
DC		[ebp+arg_0], esi	
		short loc_31308F	
066:			
		sub 31411B	
06D:			
		sub 3140F3	
		short loc_31307D	
		sub 3140F3	
		short loc_31308C	
		sub_3140F3	
	and	eax, OFFFFh	
			43/46



Any Questions?



; CODE XREF: sub 312FD8 ; sub_312FD8+59

0Dh sub_314111

> ; CODE XREF: sub_312FL : sub_312FD8+49

IRC: quend (#rpisec, #pwning) email: <u>sophia@trailofbits.com</u> thesis link: sophia.re/thesis.pdf

		45/4	
eax, OFFFFh			
sub_3140F3			
short loc_313080			
sub_3140F3			
short loc_313071			
30D_314013			

06/19/2015

References

		sub_314623
		short loc_31306D
		[ebp+arg_0], ebx
		short loc_313066
		<pre>eax, [ebp+var_70]</pre>
		eax, [ebp+var_84]
		short loc_313066
		eax, [ebp+var_84]
	push	esi
<u>-bu-2</u>	<u>2022-m</u>	organ-stanleyax
		ecasts-and-market-
<u>in p a</u>	0000	
		short loc_31306D
iaron	n ndf	eax, [ebp+arg_0] eax
Juioi	r-sh	
5/		
		[ebp+arg_4]
		sub_314623
		short loc 31306D

[6] http://www.intel.com/Assets/en_US/PDF/manual/253668.pdf [7]

http://preshing.com/20120913/acquire-and-release-semantics/

http://preshing.com/20120930/weak-vs-strong-memory-models/ [8]

http://en.wikipedia.org/wiki/Memory_barrier#An_illustrative_example [9]

http://preshing.com/20120710/memory-barriers-are-like-source-control-operations/

http://www.thewhir.com/web-hosting-news/aws-to-reach-24-billion-in-revenue [2] http://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-co

https://www.usenix.org/sustem/files/conference/usenixsecuritu14/sec14-paper-L

http://bartoszmilewski.com/2008/11/05/who-ordered-memoru-fences-on-an-x86

sub_3140F3
short loc_31307D
sub_3140F3

1

[3]

[4]

[5]

estimates-2015/