# Looking Beyond **Microarchitectural-Only Side Channels**

Mengjia Yan

mengjia@csail.mit.edu http://people.csail.mit.edu/mengjia/

> SEED Keynote September 27, 2022





compute. collaborate. create





# Meltdown & Spectre on the Headlines in 2018

# **Meltdown and Spectre: 'worst ever' CPU bugs** affect virtually all computers

Everything from smartphones and PCs to cloud computing affected by major security flaw found in Intel and other processors – and fix could slow devices.

Quotes from https://www.theguardian.com/technology/2018/jan/04/meltdownspectre-worst-cpu-bugs-ever-found-affect-computers-intel-processors-security-flaw











CACHE MISSING FOR FUN AND PROFIT

Last-Level Cache Side-Channel Attacks are Practical

#### **DRAMA: Exploiting DRAM Addressing** for Cross-CPU Attacks

Peter Pessl, Daniel Gruss, Clémentine Maurice, Michael Schwarz, and Stefan Mangard, Graz University of Technology







CACHE MISSING FOR FUN AND PROFIT

Last-Level Cache Side-Channel Attacks are Practical

#### **DRAMA: Exploiting DRAM Addressing** for Cross-CPU Attacks

Port Contention for Fun and Profit

Alejandro (

#### Don't Mesh Around: Side-Channel Attacks and Mitigations on Mesh Interconnects

Miles Dai, *MIT*; Riccardo Paccagnella, *University of Illinois at Urbana-Champaign*; Miguel Gomez-Garcia, *MIT*; John McCalpin, *Texas Advanced Computing Center*; Mengjia Yan, MIT







CACHE MISSING FOR FUN AND PROFIT

Last-Level Cache Side-Channel Attacks are Practical

#### **DRAMA: Exploiting DRAM Addressing** for Cross-CPU Attacks

Port Contention for Fun and Profit

Alejandro (

#### **Don't Mesh Around: Side-Channel Attacks** and Mitigations on Mesh Interconnects

#### Hertzbleed: Turning Power Side-Channel Attacks **Into Remote Timing Attacks on x86**

Yingchen Wang\* UT Austin

Riccardo Paccagnella\* UIUC

Hovav Shacham UT Austin

Christopher W. Fletcher UIUC

Elizabeth Tang He UIUC

> David Kohlbrenner UW

> > 3







CACHE MISSING FOR FUN AND PROFIT

Last-Level Cache Side-Channel Attacks are Practical

#### **DRAMA: Exploiting DRAM Addressing** for Cross-CPU Attacks

Port Contention for Fun and Profit

Alejandro (

#### **Don't Mesh Around: Side-Channel Attacks** and Mitigations on Mesh Interconnects

#### Hertzbleed: Turning Power Side-Channel Attacks **Into Remote Timing Attacks on x86**

Yingchen Wang\* UT Austin

Riccardo Paccagnella\* UIUC

Hovav Shacham UT Austin

Christopher W. Fletcher UIUC

Elizabeth Tang He UIUC

> David Kohlbrenner UW

> > 3





# The Age of Pervasive Hardware Attacks



# The Age of Pervasive Hardware Attacks





Spectre, Meltdown Foreshadow, Arithmetic timing



CacheOut, RIDL, Fallout Port contention, cache banking, 4K Alias

RNG Unit Covert Chanel

Flush+Reload, Prime+Probe Directory Attacks

> RowHammer, DRAMA RAMBleed





# However...



# However...

Microarchitectural Side Channels

> Side Channel via Software Resources

Memory Corruption Vulnerabilities



# Limitations of Looking At Microarchitectural-only Side Channels





# Limitations of Looking At **Microarchitectural-only Side Channels**

- Part 1: Miss threats that arise from compound threat models • Part 2: Misunderstand root causes of existing side channel attacks









PACMAN: Attacking ARM Pointer Authentication with Speculative Execution; Joseph Ravichandran\*, Weon Taek Na\*, Jay Lang, Mengjia Yan; ISCA, 2022.



## Buffer[0]

## Buffer[1]

## **Function Pointer**



## Buffer[0] **Buffer Overflow** Buffer[1] overwrites the function pointer! **Function Pointer**



# **ARM Pointer Authentication**

# PAC **16 Bits**

# Pointer

# **48 Bits**



# **ARM Pointer Authentication**

# PAC **16 Bits** Verifies

PAC = crypto\_func(pointer, salt, key)

# Pointer

# **48** Bits

# **Two Operations**

# Sign

# Before saving a pointer to memory, compute the PAC

# Verify

## Before using a pointer, check the pointer's PAC



## Buffer[0]

## Buffer[1]

...

## **Function Pointer**







# Invalid PAC means we crash!



# **Extending ARM Pointer Authentication**

# **Extending ARM Pointer Authentication**

#### PAC it up: Towards Pointer Integrity using ARM Pointer Authentication

Hans Liljestrand, Aalto University, Huawei Technologies Oy; Thomas Nyman, Aalto University: Kui Wang, Huawei Technologies Oy, Tampere University of Technologies

Ca

## PTAuth: Temporal Memory Safety via Robust Points-to Authentication

Reza Mirzazade Farkhani, Mansour Ahmadi, and Long Lu, Northeastern

#### Protecting Indirect Branches against Fault Attacks using ARM Pointer Authentication

Pascal Nasahl Graz University of Technology pascal.nasahl@iaik.tugraz.at Robert Schilling Graz University of Technology robert.schilling@iaik.tugraz.at Stefan Mangard Graz University of Technology Lamarr Security Research stefan.mangard@iaik.tugraz.at

#### Hardware-based Always-On Heap Memory Safety

Yonghae Kim Georgia Institute of Technology yonghae@gatech.edu Jaekyu Lee Arm Research jaekyu.lee@arm.com Hyesoon Kim Georgia Institute of Technology hyesoon@cc.gatech.edu



# **Extending ARM Pointer Authentication**

#### PAC it up: Towards Pointer Integrity using **ARM Pointer Authentication**

Hans Liljestrand, Aalto University, Huawei Technologies Oy; Thomas Nyman, Aalto University Kui Wana Huawei Technologies Ov Tampere University of Technol

Ca

#### **PTAuth: Temporal Memory Safety via Robust Points-to Authentication**

Reza Mirzazade Farkhani, Mansour Ahmadi, and Long Lu, Northeastern

#### Protecting Indirect Branches against Fault Attacks using ARM Pointer Authentication

Pascal Nasahl Graz University of Technology pascal.nasahl@iaik.tugraz.at

**Robert Schilling** Graz University of Technology robert.schilling@iaik.tugraz.at

Stefan Mangard Graz University of Technology Lamarr Security Research stefan.mangard@iaik.tugraz.at

### Hardware-based Always-On Heap Memory Safety

Yonghae Kim Georgia Institute of Technology yonghae@gatech.edu

Jaekyu Lee Arm Research jaekyu.lee@arm.com

Hyesoon Kim Georgia Institute of Technology hyesoon@cc.gatech.edu

The security properties of these mechanisms have been examined **solely** under the memory safety threat model.





Memory corruption vulnerability

16



Memory corruption vulnerability





Memory corruption vulnerability

## last line of defense for software attacks





Control-flow hijacking attack



Memory corruption vulnerability



## last line of defense for software attacks

PAC





PACMAN **Bypass defenses using** microarchitectural attacks

# Key Insight

## **Break PAC with Microarchitectural Attacks**

- 1. Guess a PAC speculatively to prevent crashes
- 2. Leak verification results via side channel

## ely to prevent crashes /ia side channel

if (...) { //Branch Inst A Inst B





### In-order execution:

time

if (...) { //Branch Inst A Inst B





## In-order execution:

time

time

Branch	Inst A	Inst B

Speculative execution:

Branch Inst B Inst A

18

if (...) { //Branch Inst A Inst B





## In-order execution:

time

time

Branch	Inst A	Inst B
--------	--------	--------

Speculative execution:

Branch

if (...) { //Branch Inst A Inst B





## In-order execution:

time

Branch	Inst A	Inst B
--------	--------	--------

## Speculative execution:

time



# **PACMAN Gadgets**

if (condition):

load(verified\_ptr)





Data Gadget




if (condition):

verified\_ptr = AUT(guess\_ptr) // AUT

load(verified\_ptr) // LD

time



#### Branch mispredict



Branch mispredict if (condition):

verified\_ptr = AUT(guess\_ptr) // AUT

load(verified\_ptr) // LD

time



#### Branch mispredict



Branch mispredict











#### Branch mispredict



Branch mispredict









#### Branch mispredict



Branch mispredict



#### 20

#### TARGET

The world's first desktop CPU that supports Pointer Authentication.

Image: Apple ("Apple Unleashes M1")

#### TARGET



The world's first desktop CPU that supports Pointer Authentication.

Image: Apple ("Apple Unleashes M1")

# **Challenges of Real World Hardware**

- No documentation of microarchitectural details.
- No high resolution timer.
- macOS is a difficult system to integrate attacks on.

**Essentially, we had to reinvent the wheel.** 

### **Conjectured TLB Hierarchy**

iTLB



#### PAC Oracle Accuracy







#### PAC Oracle Accuracy

 $\overline{}$ 







With a highly reliable PAC oracle, the attacker can brute-force the PAC value.

#### PacmanOS

A Rust-based bare metal environment for performing experiments.





#### PACMAN @DEFCON









# **Memory Corruption** Attacks







#### Limitations of Looking At Microarchitectural-only Side Channels

- Part 1: Miss threats that arise from compound threat models
- Part 2: Misunderstand root causes of existing side channel attacks



m compound threat models es of existing side channel attacks











Shared Resources

CPU Branch Predictor Cache DRAM

Software Resources

. . .









Signal



Shared Resources

CPU Branch Predictor Cache DRAM

Software Resources

. . .







Shared Resources

CPU Branch Predictor Cache DRAM

Software Resources

. . .







Shared Resources

CPU Branch Predictor Cache DRAM

Software Resources

. . .

Machine Learning is used to **DECODE** the signal











Machine Learning is used to **DECODE** the signal



#### ATTACKER'S CODE

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
    SWEEP_CACHE();
  }
 Trace[start] = counter;
}</pre>
```

















#### **TRACE COLLECTION**



TRAIN



#### **TRACE COLLECTION**



TRAIN









### **A Surprising Experiment**

#### **ATTACKER'S CODE**

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
    SWEEP_CACHE();
  }
 Trace[start] = counter;
}</pre>
```

### **A Surprising Experiment**

#### **ATTACKER'S CODE**

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
    REMOVE MEMORY ACCESSES
  }
  Trace[start] = counter;
}</pre>
```

### **A Surprising Experiment**

#### **ATTACKER'S CODE**

```
loop {
  start = time()
  counter = 0;
  while(time() - start < 5ms) {
    counter++;
    REMOVE MEMORY ACCESSES
  }
  Trace[start] = counter;
}</pre>
```



Chrome on Linux Chrome on Windows Safari on macOS

# What Is The Primary Side Channel?











# What Is The Primary Side Channel?







ML-assisted side-channel attacks work as a black box. It is challenging to find the root cause.
## What Is The Primary Side Channel?







ML-assisted side-channel attacks work as a black box. It is challenging to find the root cause(s).

































## System Interrupts

- Used to deal with asynchronous events
  - e.g. Graphics interrupts render content on a display

victim process



## System Interrupts

- Used to deal with asynchronous events
  - e.g. Graphics interrupts render content on a display

victim process



## System nterrupts

- Used to deal with asynchronous events
  - e.g. Graphics interrupts render content on a display

victim process





## System nterrupts

- Used to deal with asynchronous events
  - e.g. Graphics interrupts render content on a display
- Some can be "pinned" to specific cores, some can't

victim process











0 • •	🔯 jackcool	k — jack@j	ack-DX4860:
[jack@jac	k-DX4860:~\$	cat /prod	/interrupts
	CPU0	CPU1	CPU2
0:	8	0	0
8:	0	0	1
9:	0	4	0
16:	31	0	0
18:	0	8	0
23:	1943	934	0
24:	0	0	0
25:	0	0	0
26:	0	0	0
27:	0	376	0
28:	8201	0	11531
29:	0	0	17
30:	0	193	0
NMI:	0	0	0
LOC:	22059	18076	19010
SPU:	0	0	0
PMI:	0	0	0
IWI:	5794	4910	4950
RTR:	0	0	0
RES:	1400	1339	1359
CAL:	6122	6547	6563
TLB:	295	377	285
TRM:	0	0	0
THR:	0	0	0
DFR:	0	0	0
MCE:	0	0	0
MCP:	1	2	2
ERR:	0		
MIS:	0		
PIN:	0	0	0
NPI:	0	0	0
PIW:	0	0	0
jack@jac	k-DX4860:~\$		

~ — ssh < ssh jack@csg-exp2.csail.mit.edu — 94×35

CPU3

0	IO-APIC 2-edge	timer
0	IO-APIC 8-edge	rtc0
0	IO-APIC 9-fasteoi	acpi
0	IO-APIC 16-fasteoi	ehci_hcd:usb1
0	IO-APIC 18-fasteoi	i801_smbus
0	IO-APIC 23-fasteoi	ehci_hcd:usb2
0	PCI-MSI 458752-edge	PCIe PME
0	PCI-MSI 468992-edge	PCIe PME
0	PCI-MSI 524288-edge	xhci_hcd
10880	PCI-MSI 1048576-edge	enp2s0
0	PCI-MSI 512000-edge	ahci[0000:00:1f.2]
0	PCI-MSI 360448-edge	mei_me
364	PCI-MSI 32768-edge	i915
0	Non-maskable interrup	ots
27837	Local timer interrup	ts
0	Spurious interrupts	
0	Performance monitorin	ng interrupts
7493	IRQ work interrupts	
0	APIC ICR read retries	5
1262	Rescheduling interrup	ots
3100	Function call interru	upts
290	TLB shootdowns	
0	Thermal event interru	upts
0	Threshold APIC intern	rupts
0	Deferred Error APIC	interrupts
0	Machine check excepti	lons
2	Machine check polls	
0	Posted-interrupt not	ification event
0	Nested posted-interru	upt event
	Bench and States and the second	

0 Posted-interrupt wakeup event

			i i	ackco	ok —	jack@	jac	k-D	X48	60	):	
--	--	--	-----	-------	------	-------	-----	-----	-----	----	----	--

[jack@	jack-DX4860:~\$ CPU0	cat /proc/	'interrupts CPU2	CPUS			
0:	8	0	0	0	IO-APIC	2-edae	timer
8:	0	0	1	0	IO-APIC	8-edge	rtc0
9:	0	4	0	0	IO-APIC	9-fasteoi	acpi
16:	31	0	0	0	IO-APIC	16-fasteoi	ehci_hcd:usb1
18:	0	8	0	0	IO-APIC	18-fasteoi	i801_smbus
23:	1943	934	0	0	IO-APIC	23-fasteoi	ehci_hcd:usb2
24:	0	0	0	0	PCI-MSI	458752-edge	PCIe PME
25:	0	0	0	0	PCI-MSI	468992-edge	PCIe PME
26:	0	0	0	0	PCI-MSI	524288-edge	xhci_hcd
27:	0	376	0	10880	PCI-MSI	1048576-edge	enp2s0
28:	8201	0	11531	0	PCI-MSI	512000-edge	ahci[0000:00:1f.2]
29:	0	0	17	0	PCI-MSI	360448-edge	mei_me
30:	0	193	0	364	PCI-MSI	32768-edge	i915
NMT:	Ø	U	U	Ø	Non-mase	kapie interrup	τς
LOC:	22059	18076	19010	27837	Local ti	imer interrupt	S
SPU:	0	0	0	0	Spurious	s interrupts	
PMI:	0						rupts
IWI:	5794		_				
RTR:	0	ПЛ	have	lo ir	<b>stor</b>	runte	
RES:	1400		Jvan		Ιίζι	ιμισ	
CAL:	6122						
TLB:	295	0	0	0			
I KM:	0	0	0	0	Thermal	event interru	pts
IHK:	0	0	0	0	Inresno	La APIC interr	upts
DFR:	0	0	0	0	Deterred	a Error APIC 1	nterrupts
MCE:	0	0	0	0	Machine	check excepti	ons
	1	Z	Z	2	маспіпе	cneck poils	
	0						
DTN:	0	0	0	0	Postod d	interrunt noti	fication event
	0	0	0	0	Nested r	nceriupt noti	
DTW.	9	9	0	0	Posted-i	interrunt wake	up event
iack0	iack-DY4860.~¢		0	0	rosteu-1	Incerrupt wake	
Jacke	Jack DA4000. 4						

## - — ssh ∢ ssh jack@csg-exp2.csail.mit.edu — 94×35

27837	Loca	1 .	time	r i	inte	erru	pts	
~	<b>A</b>							

			i i	ackco	ok —	jack@	jac	k-D	X48	60	):	
--	--	--	-----	-------	------	-------	-----	-----	-----	----	----	--

[ <b>jack</b> @j	jack-DX4860:~\$	cat /proc/	/interrupts				]
	CPU0	CPU1	CPU2	CPU3			
0:	8	0	0	0	IO-APIC 2-edg	e timer	
8:	0	0	1	0	IO-APIC 8-edg	e rtc0	
9:	0	4	0	0	IO-APIC 9-fas	teoi acpi	
16:	31	0	0	0	IO-APIC 16-tas	teoi ehci_hcd:usb1	
18:	0	8	0	0	IO-APIC 18-tas	teoi 1801_smbus	
23:	1943	934	0	0	IO-APIC 23-tas	teo1 ehc1_hcd:usb2	
24:	0	0	0	0	PCI-MSI 458752-	edge PCIe PME	
25:	0	0	0	0	PCI-MSI 468992-	edge PCIe PME	
26:	0	0	0	0	PCI-MSI 524288-	edge xhci_hcd	
27:	0	376	0	10880	PCI-MSI 10485/6	-edge enp2s0	0.45.01
28:	8201	0	11531	0	PCI-MSI 512000-	edge ancil0000:0	0:17.2]
29:	0	0	1/	0	PCI-MSI 360448-	edge mei_me	
30:	0	193	0	364	PCI-MSI 32/68-e	dge 1915	
NM1:	00050	0	10010	07007	Non-maskable in	terrupts	
LUC:	22059	18076	19010	2/83/	Local timer int	errupts	
SPU:	0	0	0	0	Spurious interr	upts	
	570/					rupts	
IWI:	5794						
	1400		nvah	<b>D</b> ir	ntorrun		
RES:	1400 4122				ILLIUP		
TIR.	205						
	275	0	Ø	0	Thormal event i	ntorrunts	
	0	0	0	0 0		interrunte	
DED.	0	0	0	0 0	Deferred Error	ADIC interrunts	
MCE.	0	0	9	0 0	Machine check e	vcentions	
MCP ·	1	2	2	2	Machine check c	0110	
FRR.	à	2	2	2	Machine check p	0113	
MTS.	a						
PTN:	0	0	0	Q	Posted_interrun	t notification event	
NPT:	9	0	9	0	Nested posted-i	nterrunt event	
PTW:	0	0	0	0	Posted-interrup	t wakeup event	
jack@j	jack-DX4860:~\$					a nanoap orone	

## - – ssh ∢ ssh jack@csg-exp2.csail.mit.edu — 94×35

			i i	ackco	ok —	jack@	jac	k-D	X48	60	):	
--	--	--	-----	-------	------	-------	-----	-----	-----	----	----	--

-	CPU0	CPU1					
-			LPUZ	CPUR			
0:	8	0	0	0	IO-APIC 2-edge	timer	
8:	0	0	1	0	IO-APIC 8-edge	rtc0	
9:	0	4	0	0	IO-APIC 9-fasteoi	acpi	
16:	31	0	0	0	IO-APIC 16-fasteoi	ehci_hcd:usb1	
18:	0	8	0	0	IO-APIC 18-fasteoi	1801_smbus	
23:	1943	934	0	0	IO-APIC 23-fasteoi	ehci_hcd:usb2	🗧 🖛 23: Keyb
24:	0	0	0	0	PCI-MSI 458752-edge	PCIe PME	
25:	0	0	0	0	PCI-MSI 468992-edge	PCIe PME	
26:	0	0	0	0	PCI-MSI 524288-edge	xhci_hcd	
2/:	0	376	0	10880	PCI-MSI 10485/6-edge	enp2s0	46.01
28:	8201	0	11531	0	PCI-MSI 512000-edge	anci[0000:00:	11.2]
29:	0	100	1/	0	PCI-MSI 360448-edge	mei_me	
30:	0	193	0	364	PCI-MSI 32768-edge	1915	
	22050	10076	10010	0 70070	Non-maskable interrup	)TS	
	22009	T0010	14010	2/03/	Spurious interrupt	.5	
омт•	0	0	0	0	Spurious incerrupts	runte	
TWT •	5794					Lupus	
RTR.	0						
RES:	1400		ovah		nterriints		
CAL:	6122						
TLB:	295						
TRM:	0	0	0	0	Thermal event interru	ipts	
THR:	0	0	0	0	Threshold APIC inter	upts	
DFR:	0	0	0	0	Deferred Error APIC	nterrupts	
MCE:	0	0	0	0	Machine check exception	ons	
MCP:	1	2	2	2	Machine check polls		
ERR:	0						
MIS:	0						
PIN:	0	0	0	0	Posted-interrupt not	fication event	
NPI:	0	0	0	0	Nested posted-interru	ipt event	
D T L.L.	a	0	0	9	Posted-interrupt wake	eup event	
PIM:	0			<u> </u>			

## – — ssh < ssh jack@csg-exp2.csail.mit.edu — 94×35



			i i	ackco	ok —	jack@	jac	k-D	X48	60	):	
--	--	--	-----	-------	------	-------	-----	-----	-----	----	----	--

[jack@j	ack-DX4860:~\$	cat /proc/	'interrupts				]
	СРПО	CPI I1	CPU2	CPUS			
0:	8	0	0	0	IO-APIC 2-edge	timer	
8:	0	0	1	0	IO-APIC 8-edge	rtc0	
9:	0	4	0	0	IO-APIC 9-fasteoi	acpi	
16:	31	0	0	0	IO-APIC 16-fasteoi	ehci_hcd:usb1	
18:	0	8	0	0	IO-APIC 18-fasteoi	1801_smbus	
23:	1943	934	0	0	IO-APIC 23-fasteoi	ehci_hcd:usb2	
24:	0	0	0	0	PCI-MSI 458752-edge	PCIe PME	
25:	0	0	0	0	PCI-MSI 468992-edge	PCIe PME	
26:	0	0	0	0	PCI-MSI 524288-edge	xhci_hcd	
27:	0	376	0	10880	PCI-MSI 1048576-edge	enp2s0	
28:	8201	0	11531	0	PCI-MSI 512000-edge	ahcı[0000:00:	
29:	0	0	17	0	PCI-MSI 360448-edge	mei_me	
30:	0	193	0	364	PCI-MSI 32768-edge	1915	
NM1:	00050	0	0	0	Non-maskable interrup	τς	
LUC:	22059	18076	19010	2/83/	Local timer interrupt	S	
SPU:	0	0	0	0	Spurious interrupts		
	5704					rupts	
IWI:	5794						
	1400	ПЛ	hyah	lo ir	ntorrunte		
RES:	1400 6122				iciupis		
TIR.	205						
	275	0	0	0	Thormal event interru	nte	
	0	0	0	0	Threshold ADIC interr	unte	
	0	0	0	0	Deferred Error ADIC i	nterrunte	
MCE.	0	9	9	a	Machine check excenti	ons	
MCP:	1	2	2	2	Machine check nolls	0115	
FRR.	a a	2	2	2	Machine check poirs		
MTS.	0						
PIN:	0	9	0	Ø	Posted_interrunt noti	fication event	
NPT:	0	9	9	0	Nested nosted_interru	nt event	
PTW:	0	9	0	0	Posted-interrunt wake	up event	
jack@j	ack-DX4860:~\$				rootou incorrupt nako	ap orone	

## – — ssh < ssh jack@csg-exp2.csail.mit.edu — 94×35



			i i	ackco	ok —	jack@	jac	k-D	X48	60	):	
--	--	--	-----	-------	------	-------	-----	-----	-----	----	----	--

[jack@j	ack-DX4860:~\$	cat /proc/	'interrupts				1	
	СРИА	CPU1	CPU2	CPU3				
0:	8	0	0	0	IO-APIC 2-edge	timer		
8:	0	0	1	0	IO-APIC 8-edge	rtc0		
9:	0	4	0	0	IO-APIC 9-fasteoi	acpi		
16:	31	0	0	0	IU-APIC 16-fasteoi	enci_ncd:usb1		
18:	0	8	0	0	IO-APIC 18-fasteoi	1801_smbus		
23:	1943	934	0	0	IU-APIC 23-Tasteo1	enci_ncd:usb2		
24:	0	0	0	0	PCI-MSI 458/52-edge	PCIE PME		
25:	0	0	0	0	PCI-MSI 468992-edge	PCIE PME		
20:	0	0	0	10000	PCI-MSI 524288-edge	xnci_nca		
2/:	0 0001	370	11521	10000	PCI-MSI 10485/0-edge	enpzsø shoi[0000.00.	1 - 01	
20:	8201	0	17	0	PCI-MSI 512000-edge			
29:	0	102	о Т/	264	PCI-MSI 300440-euge	io15		· Cranhia
SU.	0	173	0	<u>504</u> и	Non-maskanle interrun	1713		. Graphic
	22059	18076	19010	27837	local timer interrunt	с. с		
SPU:	0	10070	0	2,00,	Sourious interrupts			
PMI:	0	Ű			opulloud intollupto	rupts		
IWI:	5794							
RTR:	0	R //						
RES:	1400		ovab	ie ir	nterrubts			
CAL:	6122							
TLB:	295							
TRM:	0	0	0	0	Thermal event interru	pts		
THR:	0	0	0	0	Threshold APIC interr	upts		
DFR:	0	0	0	0	Deferred Error APIC i	nterrupts		
MCE:	0	0	0	0	Machine check excepti	ons		
MCP:	1	2	2	2	Machine check polls			
ERR:	0							
MIS:	0							
PIN:	0	0	0	0	Posted-interrupt noti	fication event		
NPI:	0	0	0	0	Nested posted-interru	pt event		
PIW:	0	0	0	0	Posted-interrupt wake	up event		
jack@j	ack-DX4860:~\$							

## – — ssh < ssh jack@csg-exp2.csail.mit.edu — 94×35

### 40



			i i	jackc	ook –	- jack@	@jac	:k-C	<b>X</b> 4	86	0:	-
--	--	--	-----	-------	-------	---------	------	------	------------	----	----	---

[jack@	jack-DX4860:~\$ CPU0	cat /proc/	'interrupts CPU2	CPUS			
0:	8	0	0	0	IO-APIC	2-edae	timer
8:	0	0	1	0	IO-APIC	8-edge	rtc0
9:	0	4	0	0	IO-APIC	9-fasteoi	acpi
16:	31	0	0	0	IO-APIC	16-fasteoi	ehci_hcd:usb1
18:	0	8	0	0	IO-APIC	18-fasteoi	i801_smbus
23:	1943	934	0	0	IO-APIC	23-fasteoi	ehci_hcd:usb2
24:	0	0	0	0	PCI-MSI	458752-edge	PCIe PME
25:	0	0	0	0	PCI-MSI	468992-edge	PCIe PME
26:	0	0	0	0	PCI-MSI	524288-edge	xhci_hcd
27:	0	376	0	10880	PCI-MSI	1048576-edge	enp2s0
28:	8201	0	11531	0	PCI-MSI	512000-edge	ahci[0000:00:1f.2]
29:	0	0	17	0	PCI-MSI	360448-edge	mei_me
30:	0	193	0	364	PCI-MSI	32768-edge	i915
NMT:	Ø	U	U	Ø	Non-mase	kapie interrup	τς
LOC:	22059	18076	19010	27837	Local ti	imer interrupt	S
SPU:	0	0	0	0	Spurious	s interrupts	
PMI:	0						rupts
IWI:	5794		_				
RTR:	0	ПЛ	have	lo ir	<b>stor</b>	runte	
RES:	1400		Jvan		Ιίζι	ιμισ	
CAL:	6122						
TLB:	295	0	0	0			
I KM:	0	0	0	0	Thermal	event interru	pts
THK:	0	0	0	0	Inresno	La APIC interr	upts
DFR:	0	0	0	0	Deterred	a Error APIC 1	nterrupts
MCE:	0	0	0	0	Machine	check excepti	ons
	1	Z	Z	2	маспіпе	cneck poils	
ERR:	0						
MIS:	0	0	0	0	Destad a	intorrunt noti	figation overt
	0	0	0	0	Nosted-1	ncerrupt noti	
DTW.	0	0	0	0	Postod-	interrunt wake	
iack0	iack-DY/860.~¢		U	U	FUSTED-1		up event
Jacke	Jack-074000						

## - — ssh ∢ ssh jack@csg-exp2.csail.mit.edu — 94×35

				-				
27837	Loca	1 1	timer	i	.nte	rru	pts	
~	<b>A</b>							

• • •	🔯 jackcoo	ok — jack@ja	ck-DX4860	: ~ — ssh ·	ssh jack@csg-exp2.csail.mit.edu — 94×3	35
jack@jack	-DX4860:~\$	s cat /proc.	/interrupts			]
	CPU0	CPU1	CPU2	CPU3		
0:	8	0	0	0	IO-APIC 2-edge timer	
8:	0	0	1	0	IO-APIC 8-edge rtc0	
9:	0	4	0	0	IO-APIC 9-fasteoi acpi	
16:	31	0	0	0	IO-APIC 16-fasteoi ehci_hcd:usb1	
18:					S	
23:	1				usb2	
24:		Non-	mov	ahlc	nterrinte 📲	
25:				UNIC		
20:					cd	
27:	9201	0	11501	0	DCT_MST_512000_odgaoboi[0000.00	1f 01
20:	0201	0	17	0	PCI-MSI SIZ000-edge anci[0000:00	.11.2]
27.	0	103	0 T/	364	PCI-MSI 300446-euge mei_mei PCI-MSI 32768-edge i015	
MT ·	0	143	0	304	Non-maskable interrunts	
00:	22059	18076	19010	27837	local timer interrunts	
SPU:	9	0	0	2,007	Spurious interrunts	
PMT:	ő	0	Ő	0	Performance monitoring interrunts	
IWI:	5794	4910	4950	7493	IRQ work interrupts	
RTR:	0	0	0	0	APIC ICR read retries	
RES:	1400	1339	1359	1262	Rescheduling interrupts	
CAL:	6122	6547	6563	3100	Function call interrupts	
TLB:	295	377	285	290	TLB shootdowns	
TRM:	0	0	0	0	Thermal event interrupts	
THR:	0	0	0	0	Threshold APIC interrupts	
DFR:	0	0	0	0	Deferred Error APIC interrupts	
MCE:	0	0	0	0	Machine check exceptions	
MCP:	1	2	2	2	Machine check polls	
ERR:	0					
MIS:	0					
PIN:	0	0	0	0	Posted-interrupt notification event	
NPI:	0	0	0	0	Nested posted-interrupt event	
	a	0	0	0	Posted-interrupt wakeup event	
PIW:	0					

0 • •	🌔 🚾 jackcod	ok — jack@ja	ck-DX4860:	~ — ssh -	ssh jack@csg-exp2.csail.mit.edu — 94×35
[jack@ja	ck-DX4860:~	\$ cat /proc	/interrupts		]
	CPU0	CPU1	CPU2	CPU3	
0:	8	0	0	0	IO-APIC 2-edge timer
8:	0	0	1	0	IO-APIC 8-edge rtc0
9:	0	4	0	0	IO-APIC 9-fasteoi acpi
16:	31	0	0	0	IO-APIC 16-fasteoi ehci_hcd:usb1
18:					S
23:	1	_			usb2
24:		Non_	mov	ahlo	hintorrunte ME
25:				anc	FIILGIIUPLO ME
26:					cd
27:					0
28:	8201	0	11531	0	PCI-MSI 512000-edge ahci[0000:00:1f.2]
29:	0	0	17	0	PCI-MSI 360448-edge mei_me
30:	0	193	0	364	PCI-MSI 32768-edge i915
NMI:	0	0	0	0	Non-maskable interrupts
LOC:	22059	18076	19010	27837	Local timer interrupts 🗧 🖛 Hmer Interrupts
SPU:	0	0	0	0	Spurious interrupts
PMI:	0	0	0	0	Performance monitoring interrupts
IWI:	5794	4910	4950	7493	IRQ work interrupts
RTR:	0	0	0	0	APIC ICR read retries
RES:	1400	1339	1359	1262	Rescheduling interrupts
CAL:	6122	6547	6563	3100	Function call interrupts
TLB:	295	377	285	290	TLB shootdowns
TRM:	0	0	0	0	Thermal event interrupts
THR:	0	0	0	0	Threshold APIC interrupts
DFR:	0	0	0	0	Deferred Error APIC interrupts
MCE:	0	0	0	0	Machine check exceptions
MCP:	1	2	2	2	Machine check polls
ERR:	0				
MIS:	0				
PIN:	0	0	0	0	Posted-interrupt notification event
NPI:	0	0	0	0	Nested posted-interrupt event
PIW:	0	0	0	0	Posted-interrupt wakeup event
јаскеја	CK-DX4860:~	\$			

• • •	🛅 jackco	ok — jack@ja	ck-DX4860:	~ — ssh -	ssh jack@	csg-exp2.csa	ail.mit.edu — 94×35
jack@jacl	(-DX4860:~	\$ cat /proc/	/interrupts				]
	CPU0	CPU1	CPU2	CPU3			
0:	8	0	0	0	IO-APIC	2-edge	timer
8:	0	0	1	0	IO-APIC	8-edge	rtc0
9:	0	4	0	0	IO-APIC	9-fasteoi	acpi
16:	31	0	0	0	IO-APIC	<u>16-fasteoi</u>	<u>ehci_hcd:</u> usb1
18:							s
23:	1				_		usb2
24:		Non_	mov	shle	s inte	Srriin	ME
25:						<b>σιι υ</b> μ	ME
26:						-	cd
27:			44504			10000	0
28:	8201	0	11531	0	PCI-MSI 5	12000-edge	ahci[0000:00:11.2]
29:	0	0	1/	0	PCI-MSI 3	60448-edge	mei_me
30:	0	193	0	364	PCI-MSI 3	2768-edge	1915
	22050	19076	10010	0		ble interru	
	22059	T90/0	14010	2/03/		intorrunto	
5PU:	0	0	0	0	Spurious	interrupts	
PM1: TWT•	6 570/	6010	0 4050	ں 7/02		intorrunto	🗕 IDO work interru
IWI. DTD.	0	4710	4950	/473 Q	ADIC TCD	read retrie	
	1400	1330	1350	1262	Peschedul	ing interru	
	6122	6547	6563	3100	Function	call interr	ints
	295	377	285	290		downs	
TRM:	0	0	0	270	Thermal e	vent interr	upts
THR:	0	0	0	0	Threshold	APIC inter	rupts
DFR:	0	0	0	0	Deferred	Error APIC	interrupts
MCE:	0	0	0	0	Machine c	heck except	ions
MCP:	1	2	2	2	Machine c	heck polls	
ERR:	0						
MIS:	0						
PIN:	0	0	0	0	Posted-in	terrupt not:	ification event
NPI:	0	0	0	0	Nested po	sted-interr	upt event
PIW:	0	0	0	0	Posted-in	terrupt wak	eup event
јаскејасн	C-DX4860:~	\$					

• • •	🔯 jackcoo	ok — jack@ja	ck-DX4860	: ~ — ssh ·	ssh jack@csg-exp2.csail.mit.edu — 94×3	35
jack@jack	-DX4860:~\$	s cat /proc.	/interrupts			]
	CPU0	CPU1	CPU2	CPU3		
0:	8	0	0	0	IO-APIC 2-edge timer	
8:	0	0	1	0	IO-APIC 8-edge rtc0	
9:	0	4	0	0	IO-APIC 9-fasteoi acpi	
16:	31	0	0	0	IO-APIC 16-fasteoi ehci_hcd:usb1	
18:					S	
23:	1				usb2	
24:		Non-	mov	ahlc	nterrinte 📲	
25:				UNIC		
20:					cd	
27:	9201	0	11501	0	DCT_MST_512000_odgaoboi[0000.00	1f 01
20:	0201	0	17	0	PCI-MSI SIZ000-edge anci mo	.11.2]
27.	0	103	0 T/	364	PCI-MSI 300446-euge mei_mei PCI-MSI 32768-edge i015	
MT ·	0	143	0	304	Non-maskable interrunts	
00:	22059	18076	19010	27837	local timer interrunts	
SPU:	9	0	0	2,007	Spurious interrunts	
PMT:	ő	0	Ő	0	Performance monitoring interrunts	
IWI:	5794	4910	4950	7493	IRQ work interrupts	
RTR:	0	0	0	0	APIC ICR read retries	
RES:	1400	1339	1359	1262	Rescheduling interrupts	
CAL:	6122	6547	6563	3100	Function call interrupts	
TLB:	295	377	285	290	TLB shootdowns	
TRM:	0	0	0	0	Thermal event interrupts	
THR:	0	0	0	0	Threshold APIC interrupts	
DFR:	0	0	0	0	Deferred Error APIC interrupts	
MCE:	0	0	0	0	Machine check exceptions	
MCP:	1	2	2	2	Machine check polls	
ERR:	0					
MIS:	0					
PIN:	0	0	0	0	Posted-interrupt notification event	
NPI:	0	0	0	0	Nested posted-interrupt event	
	a	0	0	0	Posted-interrupt wakeup event	
PIW:	0					

# **Non-Movable Interrupts**

- Can't be isolated from any cores
- Have not been studied in detail for side channels

Are necessary for the operating system to function










# System Instrumentation

In the kernel space: use eBPF

- Allows instrumentation of the Linux kernel at runtime
- We developed a tool to monitor interrupt characteristics by recording time at beginning and end of interrupt handlers

In the user space: attacker code in Rust

- Records time leaves and re-enters the user space



















Jack Cook, Jules Drean, Jonathan Behrens, Mengjia Yan

Published at the 49th International Symposium on Computer Architecture (ISCA)

Use this tool to collect your own traces! For best results, close all other programs and keep your mouse still while collecting traces.

	○ 1 second (	5 seconds	$\odot$ 15 seconds		
Website					
None	<ul> <li>nytimes.com</li> </ul>	amazon.com	○ weather.com	n 🔿 Custom	
	Collect trace	Down	load traces		
	<b>b</b>				

### Key Takeaways

- Machine learning can be used to identify activity on your computer from traces recorded in JavaScript that measure CPU instruction throughput over time
- We found this type of attack exploits signals from system interrupts, which operating systems use to interact with hardware devices
  - When a core processes interrupts, it pauses the execution of an attacker, creating a signal that can be exploited
- Our loop-counting attack can correctly identify one of 100 websites being opened 96.6% of the time in Chrome on Linux



### Interactive Demo

Trace I	ength
---------	-------





Jack Cook, Jules Drean, Jonathan Behrens, Mengjia Yan

Published at the 49th International Symposium on Computer Architecture (ISCA)

Use this tool to collect your own traces! For best results, close all other programs and keep your mouse still while collecting traces.

	○ 1 second (	5 seconds	$\odot$ 15 seconds		
Website					
None	<ul> <li>nytimes.com</li> </ul>	amazon.com	○ weather.com	n 🔿 Custom	
	Collect trace	Down	load traces		
	<b>b</b>				

### Key Takeaways

- Machine learning can be used to identify activity on your computer from traces recorded in JavaScript that measure CPU instruction throughput over time
- We found this type of attack exploits signals from system interrupts, which operating systems use to interact with hardware devices
  - When a core processes interrupts, it pauses the execution of an attacker, creating a signal that can be exploited
- Our loop-counting attack can correctly identify one of 100 websites being opened 96.6% of the time in Chrome on Linux



### Interactive Demo

Trace I	ength
---------	-------





Ρ	a

<ul> <li>1 second</li> </ul>	d
------------------------------	---

○ None	$\bigcirc$ nv	times.com	$\bigcirc$
		Collect tra	ace

### Key Takeaways

- time
- operating systems use to interact with hardware devices
  - creating a signal that can be exploited
- opened 96.6% of the time in Chrome on Linux

• Machine learning can be used to identify activity on your computer from traces recorded in JavaScript that measure CPU instruction throughput over

**Download traces** 

• We found this type of attack exploits signals from system interrupts, which

• When a core processes interrupts, it pauses the execution of an attacker,

Our loop-counting attack can correctly identify one of 100 websites being





## cache side channel

## interrupt side channel





### cache side channel

## interrupt side channel

## **Takeaway 2: There's** always a bigger fish!

## **Need comprehensive** security analysis in **complex SW-HW systems**





Run in separate VMs with interrupts pinned properly

- Run in separate VMs with interrupts pinned properly
  - 88.2% → 91.6%



- Run in separate VMs with interrupts pinned properly
  - 88.2% → 91.6%



How to decipher signals from the ML model output?



- Run in separate VMs with interrupts pinned properly
  - 88.2% → 91.6%



How to decipher signals from the ML model output?

• A "bigger bigger" fish?





Takeaway 1: New threats arising from compound threat models Takeaway 2: Need comprehensive security analysis for complex SW-HW systems



Takeaway 1: New threats arising from compound threat models Takeaway 2: Need comprehensive security analysis for complex SW-HW systems



Takeaway 1: New threats arising from compound threat models

# Takeaway 2: Need comprehensive security analysis for complex SW-HW systems

## Learning Computer Architecture Security For Fun — 5 Lab Assignments

## Learning Computer Architecture Security For Fun — 5 Lab Assignments



## Learning Computer Architecture Security For Fun — 5 Lab Assignments











## Learning Computer Architecture Security For Fun – 5 Lab Assignments







http://csg.csail.mit.edu/6.888Yan/for instructors/





## The Team



Peter Deutsch



Yuheng Yang



Weon Taek Na



Jules Drean









## Joseph Ravichandran



Jack Cook



Miguel Gomez-Garcia





Microarchitectural Side Channels



Side Channel via Software Resources

Many Others ...

Memory Corruption Vulnerabilities