SegScope: Probing Fine-grained Interrupts via Architectural Footprints

Xin Zhang[*12], Zhi Zhang[*3], Qingni Shen[†12], Wenhao Wang[4], Yansong Gao[5], Zhuoxi Yang[12], Jiliang Zhang[6]

[*] Both authors contributed equally to this work

[1]School of Software and Microelectronics, Peking University [2]PKU-OCTA Laboratory for Blockchain and Privacy Computing, Peking University
[3]The University of Western Australia [4]Institute of Information Engineering, CAS [5]Data61, CSIRO [6]College of Semiconductors, Hunan University

Motivation

As they are related to system activities, **interrupts** can be used to mount various side-channel attacks (e.g., monitoring keystrokes and inferring website visits). Given that all these attacks rely on system file interfaces or architectural timers to probe interrupts, various countermeasures have been proposed to either remove the unprivileged access to the file interfaces or detect/cripple architectural timers.

To answer the question: **Is there a microarchitectural technique across x86 CPUs probing interrupts without any timers?** we propose **SegScope**, a new technique that abuses segment protection on x86 to acquire fine-grained interrupt observations without relying on any timer.

Interrupt Probing

- To the best of our knowledge, **SegScope** is the first interrupt probing technique without any timer. Since segment protection is supported on x86 by default, **SegScope** affects mainstream x86-based CPUs.

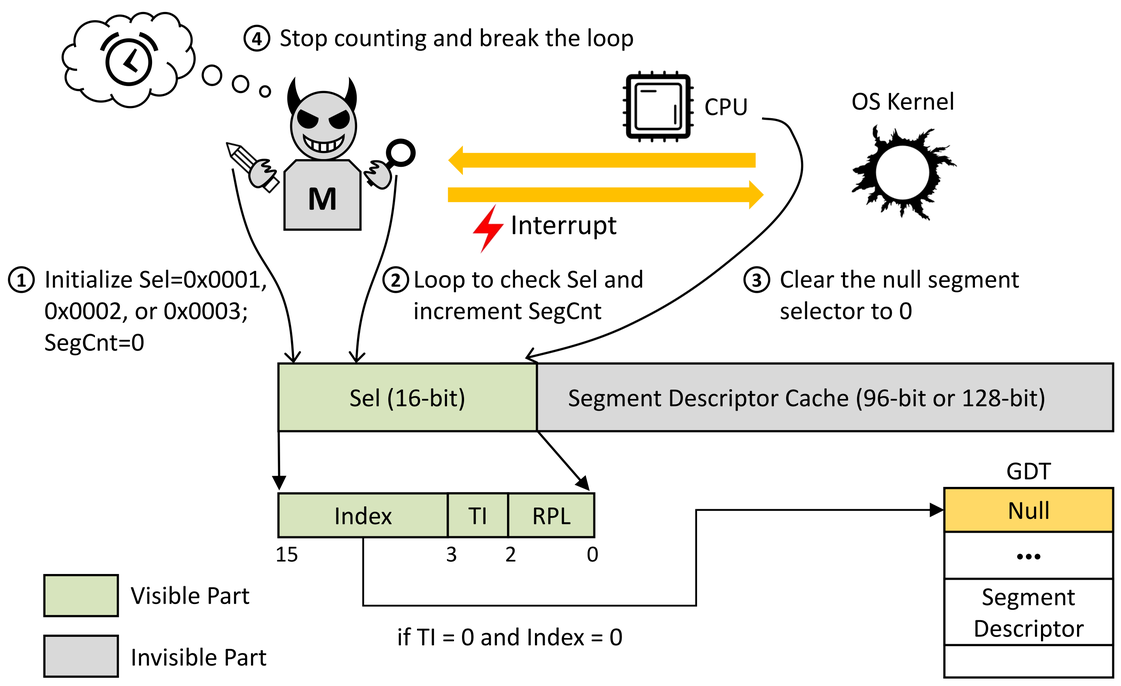

Figure 1. The overview of **SegScope**, which exploits the footprints left by segment protection to detect interrupts and leverages a counter (i.e., SegCnt) to represent the time interval between two consecutive interrupts.

Fingerprinting Websites

With fine-grained interrupt observations, SegScope is used as a side channel to infer website visits with a respective success rate of 92.4% on Chrome and 87.4% on Tor Browser in the default system setting.

Setting	Chrome 109		Tor Browser 12	
	Top-1 Acc	Top-5 Acc	Top-1 Acc	Top-5 Acc
Default	$92.4\% \pm 0.4$	$98.4\% \pm 0.2$	$87.4\% \pm 1.4$	$97.3\% \pm 0.4$
Different cores used	$91.0\% \pm 0.8$	$98.1\% \pm 0.4$	$83.3\% \pm 1.4$	$96.3\% \pm 0.2$
Frequency scaling disabled	$94.6\% \pm 0.5$	$98.9\% \pm 0.3$	$87.4\% \pm 0.9$	$96.5\% \pm 0.3$
Hyper-threading disabled	$94.5\% \pm 0.7$	$98.8\% \pm 0.3$	$89.5\% \pm 0.8$	$97.2\% \pm 0.3$

Table 1. Classification accuracy (avg±std) across 10-fold cross validation for website fingerprinting.

Enhancing Spectral Attack

A Spectral attacker cannot distinguish cache line writes and interrupts because both the two architectural events clear the used carry flag. We show that a Spectral attacker can use SegScope to filter out the interrupted measurements, thereby enhancing his attack.

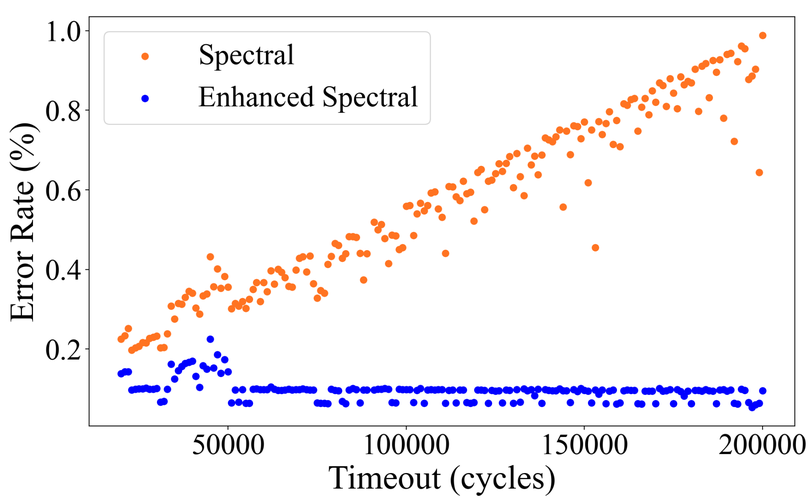

Figure 2. **SegScope** can remove the impacts of interrupts and significantly reduce the error rate of Spectral.

Fine-grained Timer

- We further rely on **SegScope** to probe timer interrupts and thus craft a fine-grained timer. The crafted timer has been evaluated on multiple machines with Intel- or AMD-based CPUs in either local or Amazon cloud environments.

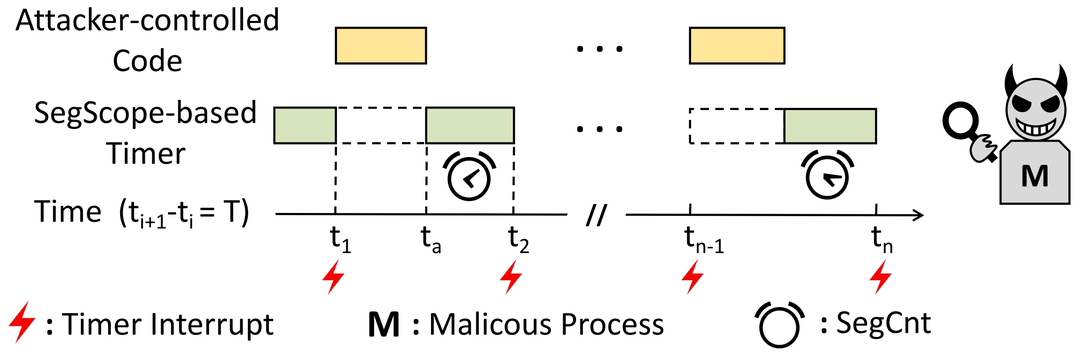

Figure 3. An illustration of using **SegScope**-based timer to measure the execution time of attacker-controlled code.

Extracting Cryptographic Keys

We verify that the workload of cryptographic library (i.e., CIRCL) can effect our SegCnt. A correct key-bit guess ($(m_i \neq m_{i-1})$) causes the processor to execute at a higher frequency than an incorrect key-bit guess does, resulting in higher SegCnt.

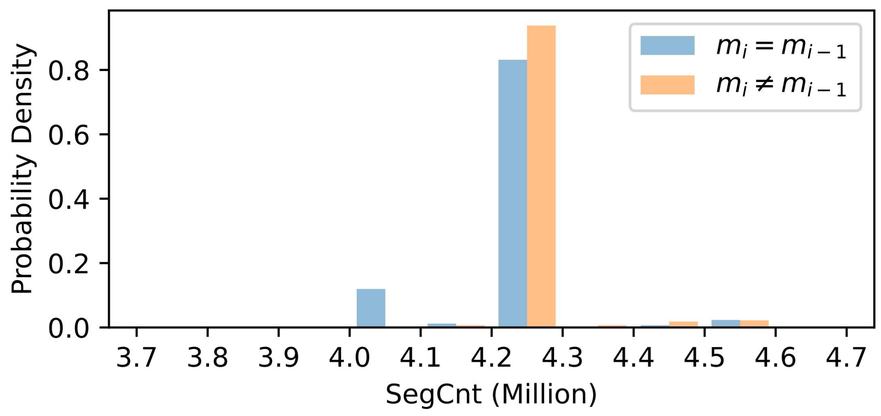

Figure 4. Distribution of readout when the challenge ciphertext introduces an anomalous 0 ($m_i \neq m_{i-1}$) or not ($m_i = m_{i-1}$).

Stealing DNN Model Architectures

SegScope can be used to steal DNN model architectures with an accuracy of over 80%.

Layer	Conv	BN	ReLu	MP	AP	Linear	Overall
SA (%)	98.2	77.8	58.6	85.2	50.4	52.8	97.7
LDA (%)	87.7	86.0	85.6	85.6	86.5	86.9	87.2

Table 2. Classification accuracy of network steps (BN = Batch Normalize, MP = Max Pooling, AP = Average Pooling).

Breaking KASLR

The idea of breaking KASLR using **SegScope** is to build a **SegScope**-based stealthy timer to measure the execution time of attacker's operations, obviating usage of architectural timer. Experimental results show that We can break KASLR within about 10 second.

Machine	Param. C	Time (s)	Top-1 Acc	Top-5 Acc
Xiaomi Air 13.3	1	2.14	63.7%	98.4%
	5	10.28	100%	100%
Lenovo Yangtian 4900v	1	2.05	96.1%	100%
	5	10.24	100%	100%
Amazon t2.large	1	2.05	83.0%	99.7%
	5	10.21	100%	100%
Amazon c5.large	1	2.06	87.2%	99.2%
	5	10.31	100%	100%

Table 3. Evaluation of using **SegScope** to break KASLR on various environments.