The Spy In The Sandbox: Practical Cache Attacks In Javascript And Their Implication

Yossef Oren Vasileios P. Kemerlis Simha Sethumadhavan Angelos D. Keromytis

Apresentação por Renan Greca

INTRODUÇÃO

CACHE ATTACKS

- CPUs Intel
- Cache inclusivo
- Cache L3 compartilhado entre núcleos

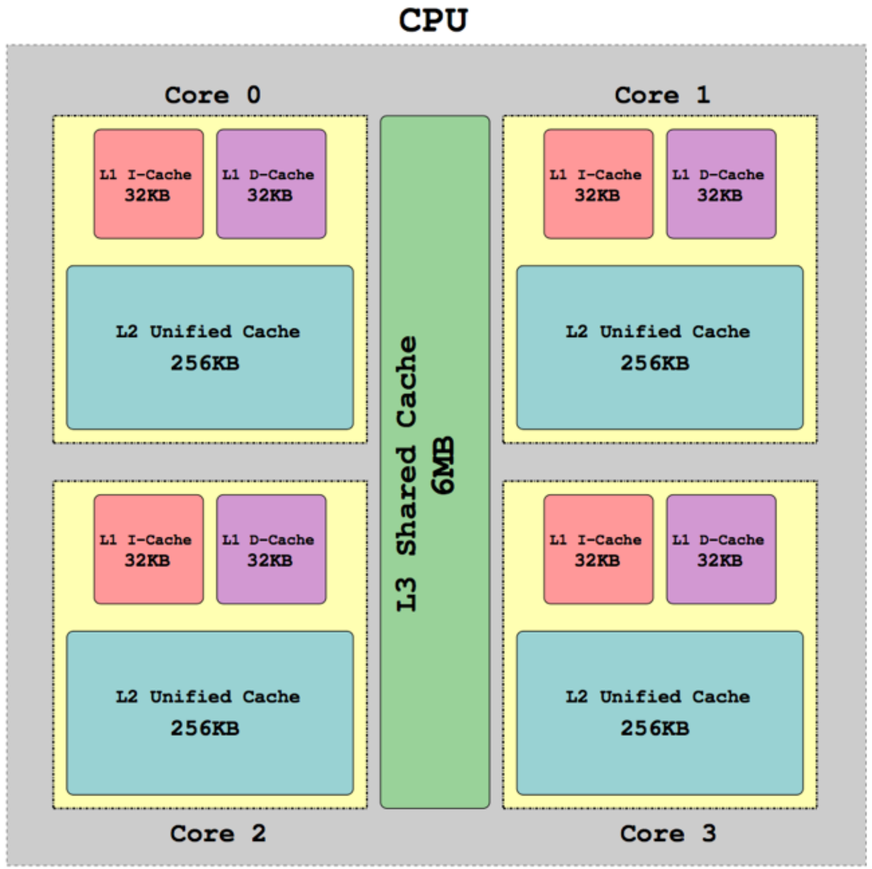

Figure 1: Cache memory hierarchy of Intel CPUs (based on Ivy Bridge Core i5-3470).

PRIME + PROBE

- Eviction sets
- "Preparo e consulta"
- Deixar cache em estado conhecido
- Consultar cache após operações do usuário
- Encontrar regiões interessantes de cache

Algorithm 1 Profiling a Cache Set.

Let S be the set of currently unmapped page-aligned addresses, and address x be an arbitrary page-aligned address in memory.

1. Repeat k times:
   (a) Iteratively access all members of S.
   (b) Measure t_1, the time it takes to access x.
   (c) Select a random page s from S and remove it.
   (d) Iteratively access all members of $S \backslash s$.
   (e) Measure t_2, the time it takes to access x.
   (f) If removing s caused the memory access to speed up considerably (i.e., $t_1 - t_2 > thres$), then this address is part of the same set as x. Place it back into S.
   (g) If removing s did not cause memory access to speed up considerably, then s is not part of the same set as x.
2. If $|S| = 12$, return S. Otherwise report failure.

DETECÇÃO DA CACHE

- Verificar quais regiões de cache foram usadas através da latência

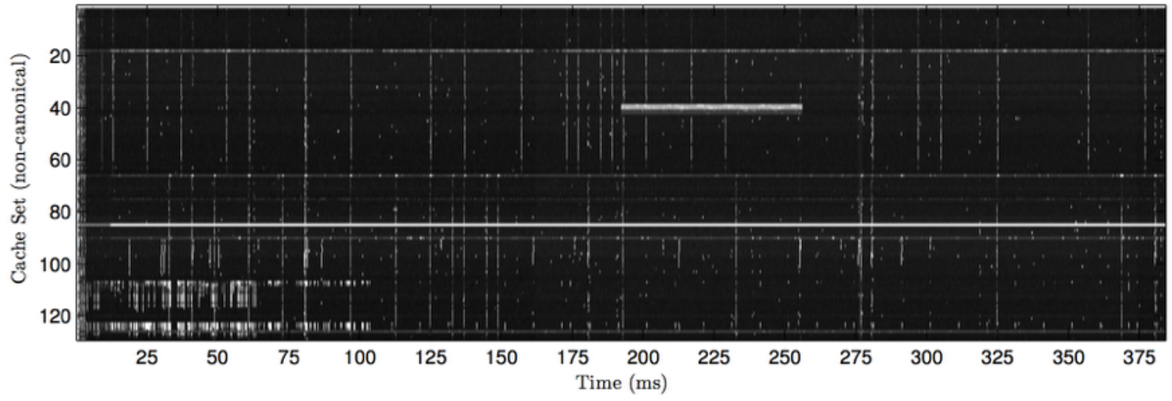

Figure 4: Sample memorygram collected over an idle period of 400ms. The X axis corresponds to time, while the Y axis corresponds to different cache sets. The sample shown has a temporal resolution of 250µs and monitors a total of 128 cache sets. The intensity of each pixel illustrates the access latency of the particular cache set, with black representing low latency and white representing a higher latency.

DETEÇÃO DO USUÁRIO

- Atividades de rede
- Atividades do mouse

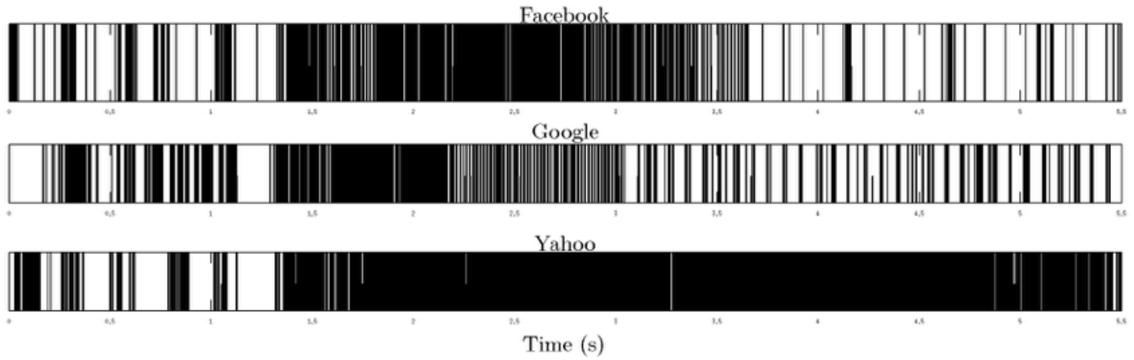

Figure 6: Memorygrams for three popular websites (Facebook, Google, Yahoo).

Classifier Output→, Ground Truth↓	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Amazon (1)	.8	-	-	-	-	-	-	.2
Baidu (2)	.2	.8	-	-	-	-	-	-
Facebook (3)	-	-	.5	-	-	.5	-	-
Google (4)	-	-	-	1	-	-	-	-
Twitter (5)	-	-	-	-	1	-	-	-
Wikipedia (6)	-	-	.2	-	-	.8	-	-
Yahoo (7)	-	-	-	-	-	-	1	-
Youtube (8)	-	-	-	-	.4	-	-	.6

Table 2: Confusion matrix for FFT-based classifier (Safari Private Browsing).

VULNERABILIDADE COMUM

- CPUs Intel são mais de 80% dos computadores vendidos desde 2011
- Navegadores vulneráveis também representam mais de 80% dos usuários

Brand	Hi-Res. Time Support	Typed Arrays Support	Worldwide Prevalence
Internet Explorer	v10	v11	11.77%
Safari	v8	v6	1.86%
Chrome	v20	v7	50.53%
Firefox	v15	v4	17.67%
Opera	v15	v12.1	1.2%
Total	–	–	83.03%

Table 4: Affected desktop browsers: minimal version and prevalence [26].

PROBLEMAS

- Detecção correta de rede apenas 58%
- Pequeno número de *websites*
- Código JavaScript é sempre aberto
- Operações em *background* podem afetar os resultados

Our results indicate that it is possible to reliably detect mouse and network activity. The measurement rate of our network classifier did not allow us to count individual packets, but rather monitor periods of network (in)activity. Our detector was able to correctly detect 58% of these active periods, with a false positive rate of 2.86%. The mouse de-

SOLUÇÕES À VULNERABILIDADE

- Poderia ser evitado com modificações ao JavaScript ou à arquitetura
- Possivelmente detecção pelo *runtime* do JavaScript