Contact: Samira.Briongos@neclab.eu, ida.bruhns@uni-luebeck.de

Aim, Wait, Shoot: How the CacheSniper Technique Improves Unprivileged Cache Attacks

Samira Briongos[1], Ida Bruhns[2], Pedro Malagón[3], Thomas Eisenbarth[2] and José M. Moya[3]

[1]NEC Laboratories Europe, [2]Universität zu Lübeck, [3]Universidad Politécnica de Madrid

September 10, 2021

UNIVERSITÄT ZU LÜBECK

POLITÉCNICA

Cache attacks and countermeasures

- Traditional cache attacks: Flush+Reload and Prime+Probe.

```
1: function victim_function
       .
       :
2:     :
3:
4:
5:     load table[secret]
6: end function
```

- Attacker:
  - Removes table[secret] from the cache.
  - Waits for the victim to execute.
  - Looks for table [secret] in the cache.
    - If it is in the cache: victim has used it.
    - If not: victim has not used it.
  - Infers the value of **secret.**

Cache attacks and countermeasures

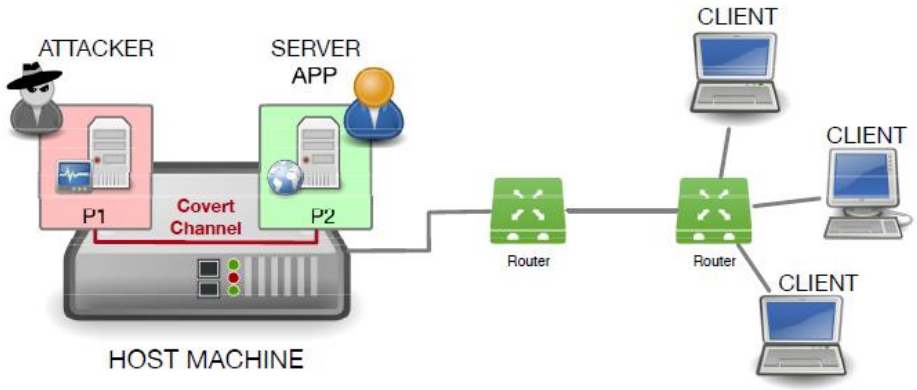
- One possible countermeasure: prefetching or always-load-strategy

```
1: function victim_function
       .
       .
2:     .
3:
4:     load table
5:     load table[secret]
6: end function
```

- Idea:
  - Ensure the entire table is always available in the cache.
  - Limit the attacker capabilities: she can not distinguish. whether the access is due to prefetching or not.
- Easy to implement.
- Applied in different cryptographic libraries.
  - It does not remove secret dependent access to the cache.

Window for attack

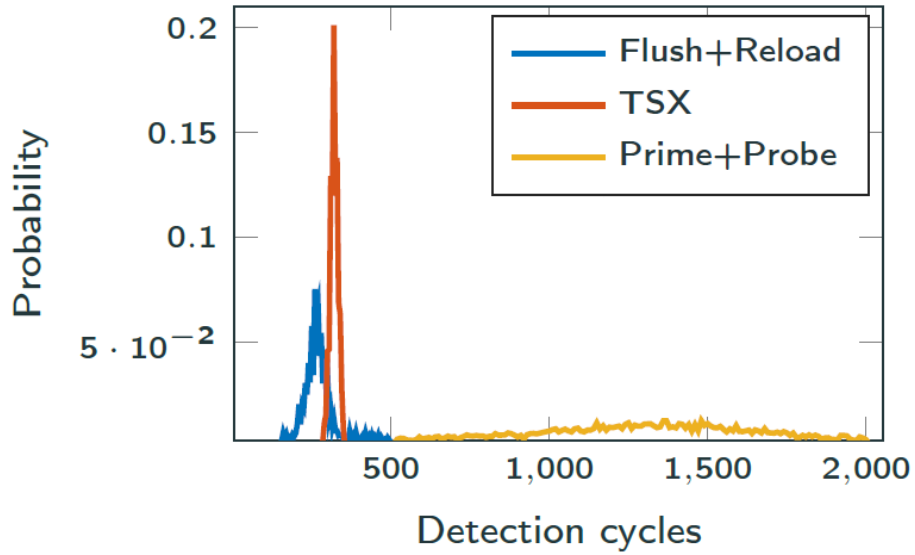
- Prefetch-protected implementations can still be attacked.



- Attacker's challenges:
  1. Detect the victim's execution of the target algorithm.
  2. Determine the state of the target after detection.
  3. Calculate the remaining time until data is prefetched.
  4. Evict the target data from the LLC at the desired instant.

Detect and determine state

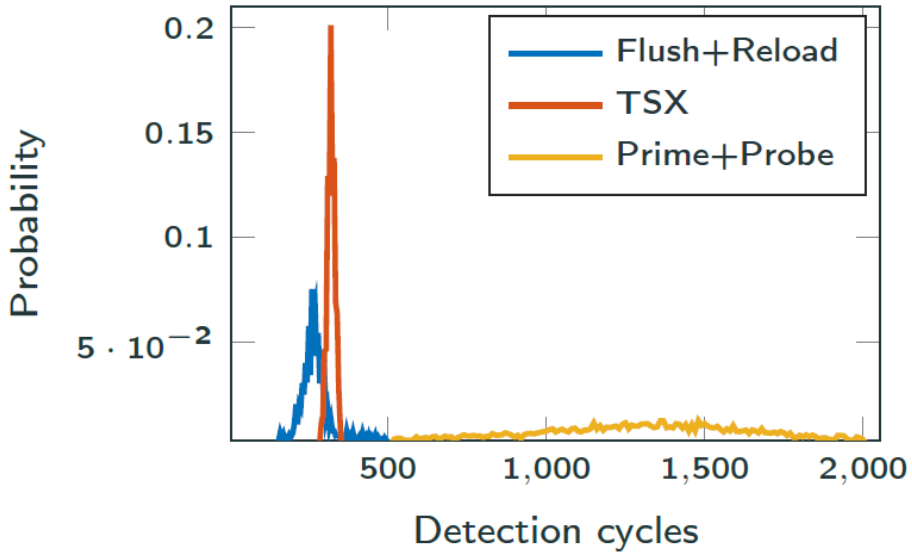
- Tested Flush+Reload, Prime+Probe and TSX-Abort for detection.

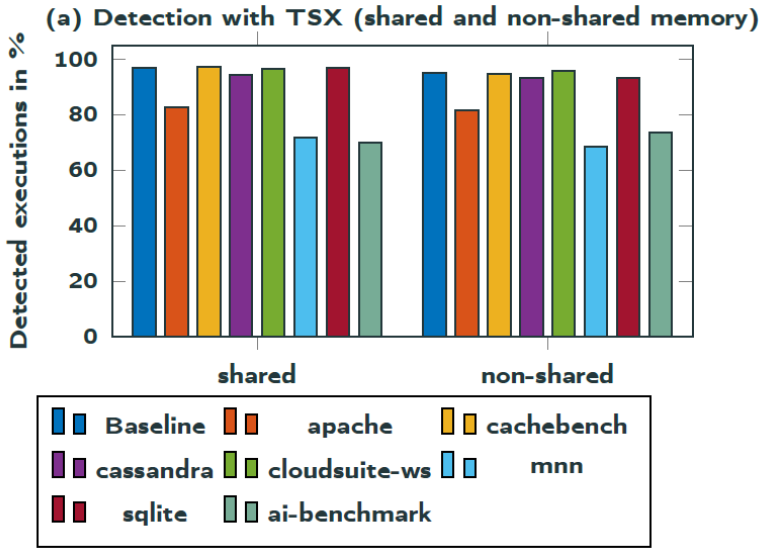


- TSX is the best approach for synchronization.
  - Data mapping to the same location of the target is loaded into a transaction.
  - An abort signal is received as soon as the victim starts executing the target algorithm.
  - Time between victim's call and abort is almost constant → Victim state is known after abort.

Detect and determine state

- Tested Flush+Reload, Prime+Probe and TSX-Abort for detection.





- TSX is the best approach for synchronization.
  - Data mapping to the same location of the target is loaded into a transaction.
  - An abort signal is received as soon as the victim starts executing the target algorithm.
  - Time between victim's call and abort is almost constant → Victim state is known after abort.

Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.

Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.
  - Known LLC replacement policy
  - Controlled cache state (Transactional Memory)

  - SINGLE ACCESS EVICTION

Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.
  - Known LLC replacement policy
  - Controlled cache state (Transactional Memory)

- SINGLE ACCESS EVICTION

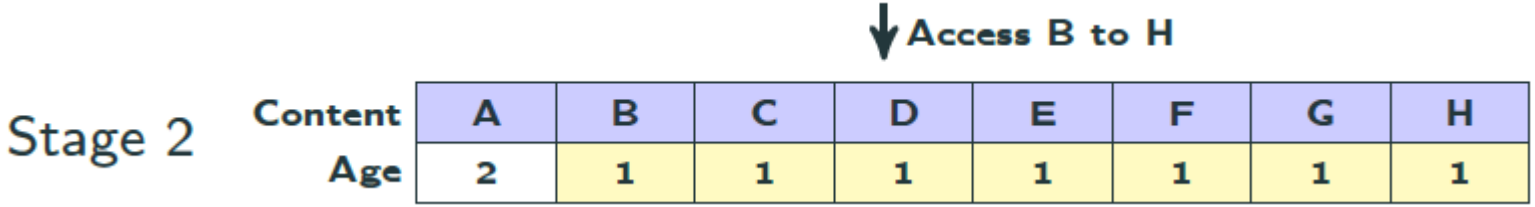
Stage 1

Content	A	B	C	D	E	F	G	H
Age	2	2	2	2	2	2	2	2

Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.
  - Known LLC replacement policy
  - Controlled cache state (Transactional Memory)

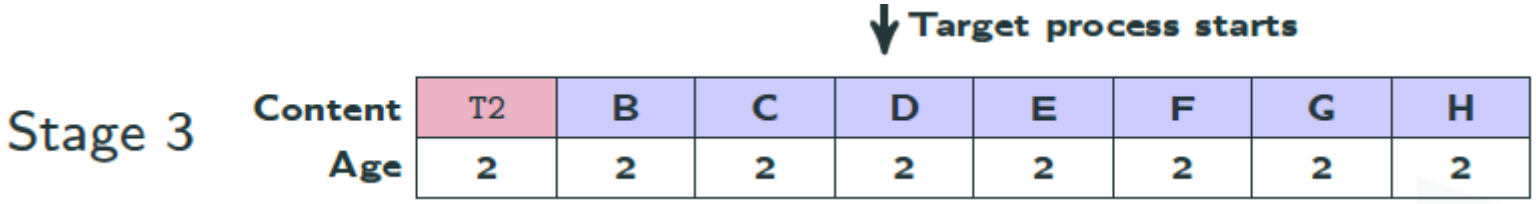
→ SINGLE ACCESS EVICTION



Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.
  - Known LLC replacement policy
  - Controlled cache state (Transactional Memory)

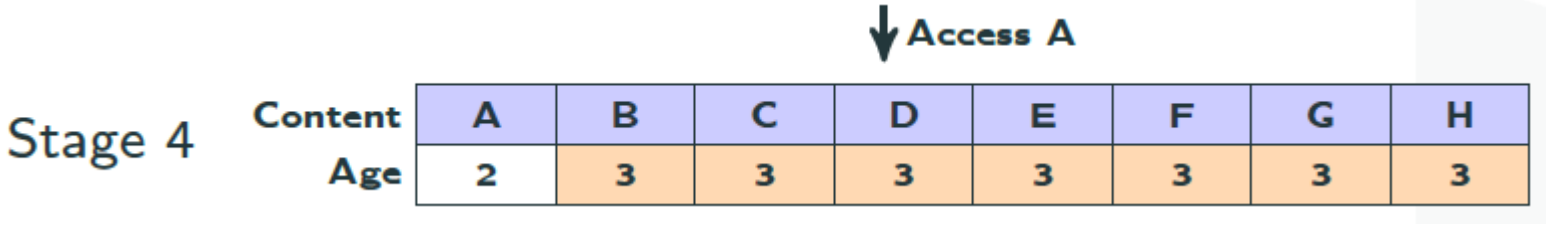
→ SINGLE ACCESS EVICTION



Wait for prefetch and evict

- Attacker can profile the target code beforehand and find the time elapsed between point of detection and prefetch.
- Evict the target data at the right instant.
  - Known LLC replacement policy
  - Controlled cache state (Transactional Memory)

→ SINGLE ACCESS EVICTION



Targets

- AES software implementations of OpenSSL.
  - S-Box implementation.
  - 4 cache lines in total.
  - Prefetch the table before each round.

Parameter	T-Table	S-Box
Detection target T1	AES_encrypt	S-Box (Prefetch in first round)
Eviction target T2	Tei[0]	S-Box (After Prefetch in last round)
Samples required	360	≈500000

Targets

- RSA implementation of wolfSSL (CVE-2020-15309).

Require: base b, modulo m, exponent $e = (e_{n-1} \dots e_0)_2$
Ensure: $b^e \pmod m$

```
1: init(R);
2: for i from n − 1 downto 0 do
3:     mul(R[0], R[1], R[e_i]));
4:     modRed(R[e_i]);
5:
6:     sqr(R[2], R[2]);
7:     modRed(R[2]);
8: end for
9: return R;
```

- 96,8% of the multiply operations detected.
- 87.6% precision identifying e_i.

Conclusions

- Prefetching does not work even if the attacker is unprivileged.
  - If she can synchronize with the victim and achieve fast cache evictions.
- TSX works well for the synchronization.
  - The changes made to the cache are not reverted.
- We showed a technique for single access eviction.
  - AIM: Prepare the scenario for the attack.
  - WAIT: For the right moment.
  - SHOOT: Evict data from the cache.
- Demonstration with two real world targets.
  - AES (S-Box Openssl).
  - RSA (WolfSSL).

Thank you for your attention

Source code: https://github.com/greenlsi/CacheSniper

Contact: : Samira.Briongos@neclab.eu, ida.bruhns@uni-luebeck.de

\Orchestrating a brighter world NEC

UNIVERSITÄT ZU LÜBECK

POLITÉCNICA