Građa računala

Sigurnost hardvera
(dodatna prezentacija)

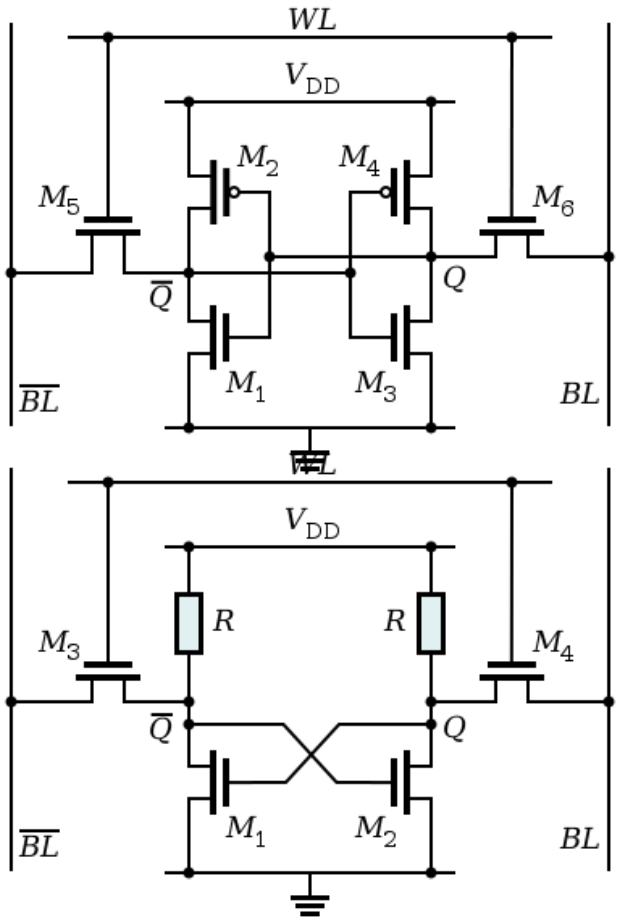
Generalna sigurnost hardvera (The semiconductor security war)

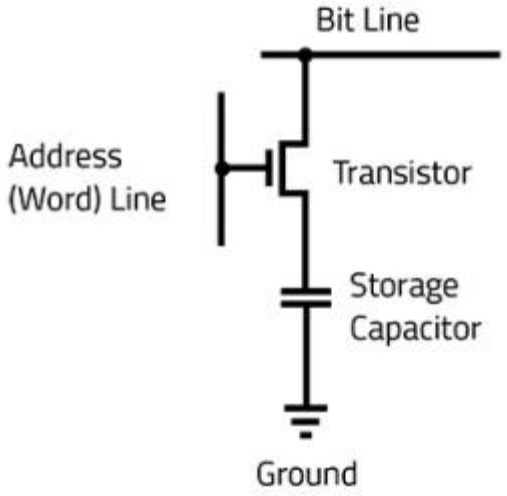
RowHammer napadi

Uobičajene floskule

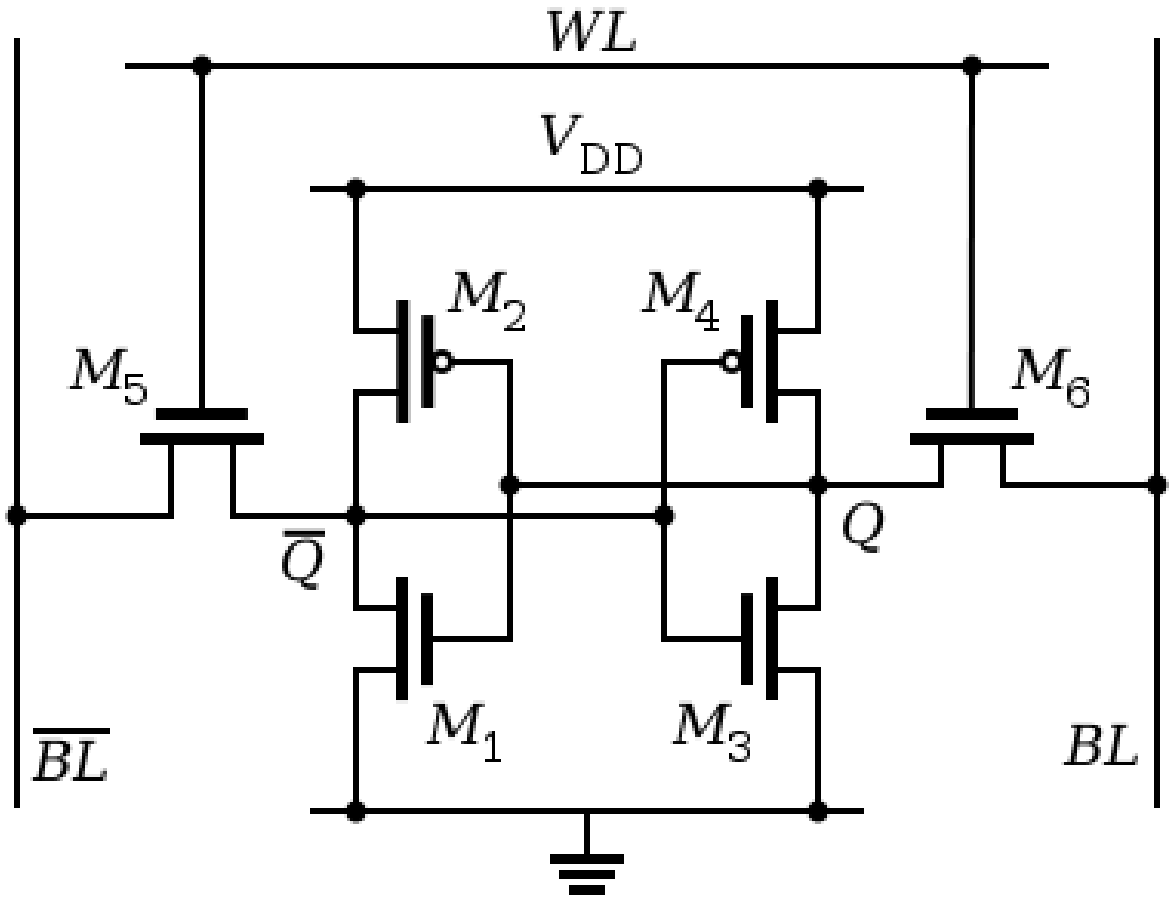
- “Zaaaašto kompliciramo – idemo rješiti problem memorije tako da maknemo “sporu” DRAM memoriju, I idemo sve držati u brzoj, SRAM memoriji I cache memoriji”
- Razmislimo zašto je ovo nepraktično I nemoguće za izvesti

SRAM vs DRAM

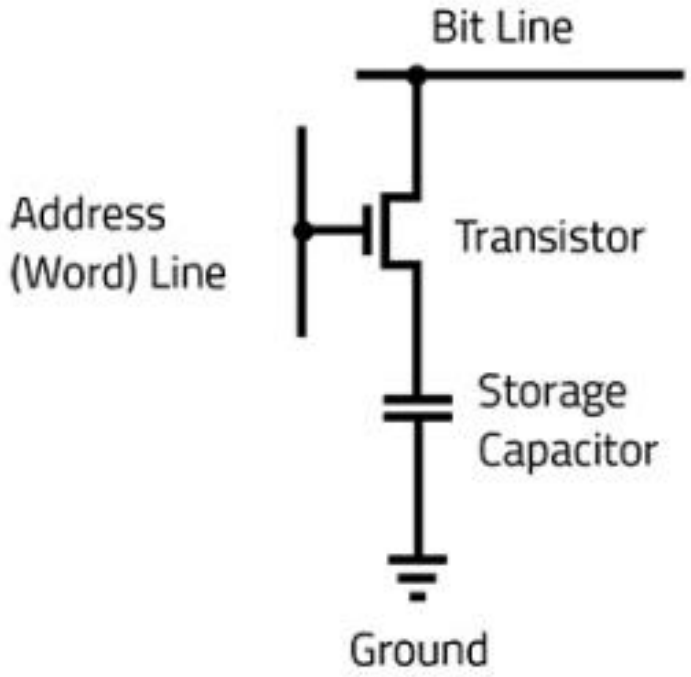

WL
V_{DD}
M_2
M_4
M_5
M_6
$\overline{Q}$
Q
M_1
M_3
$\overline{BL}$
BL
WL
V_{DD}
R
R
M_3
M_4
$\overline{Q}$
Q
M_1
M_2
$\overline{BL}$
BL


Bit Line
Address
(Word) Line
Transistor
Storage
Capacitor
Ground

SRAM



DRAM


Bit Line
Address
(Word) Line
Transistor
Storage
Capacitor
Ground

Primjer (Sandy Bridge arhitektura)

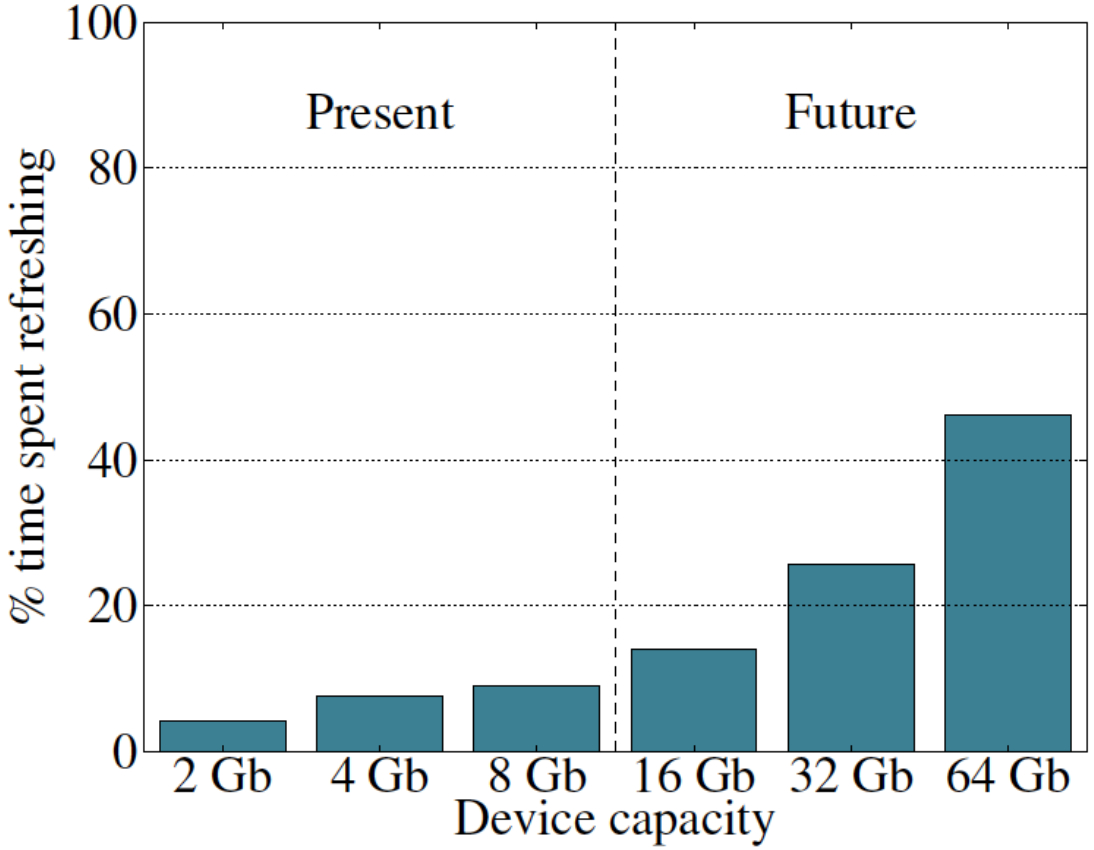
- L1 cache size 32KB cca 3 cycles
- L2 cache size 256KB cca 8 cycles
- L3 cache size 10-20MB cca 35 cycles
- Main memory – whatever GBs are supported/we put in, cca 250 cycles

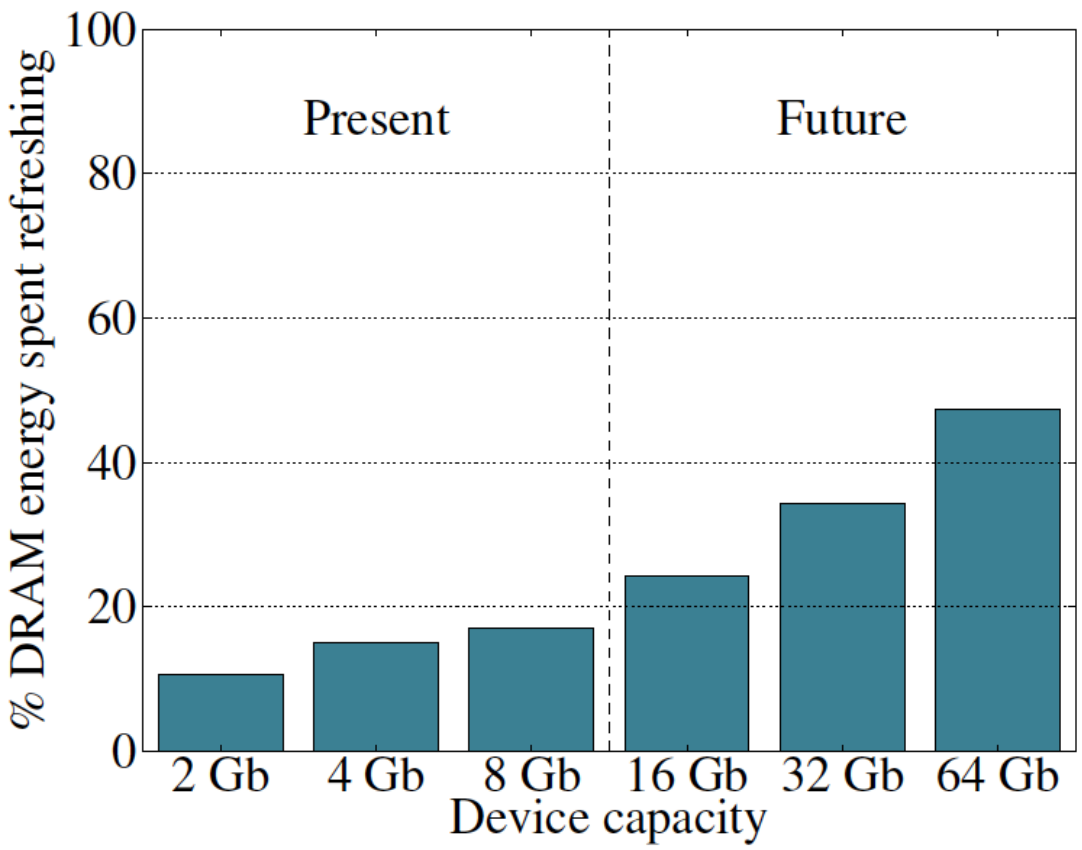
Zašto nas to zanima?

Objašnjenje je jednostavno

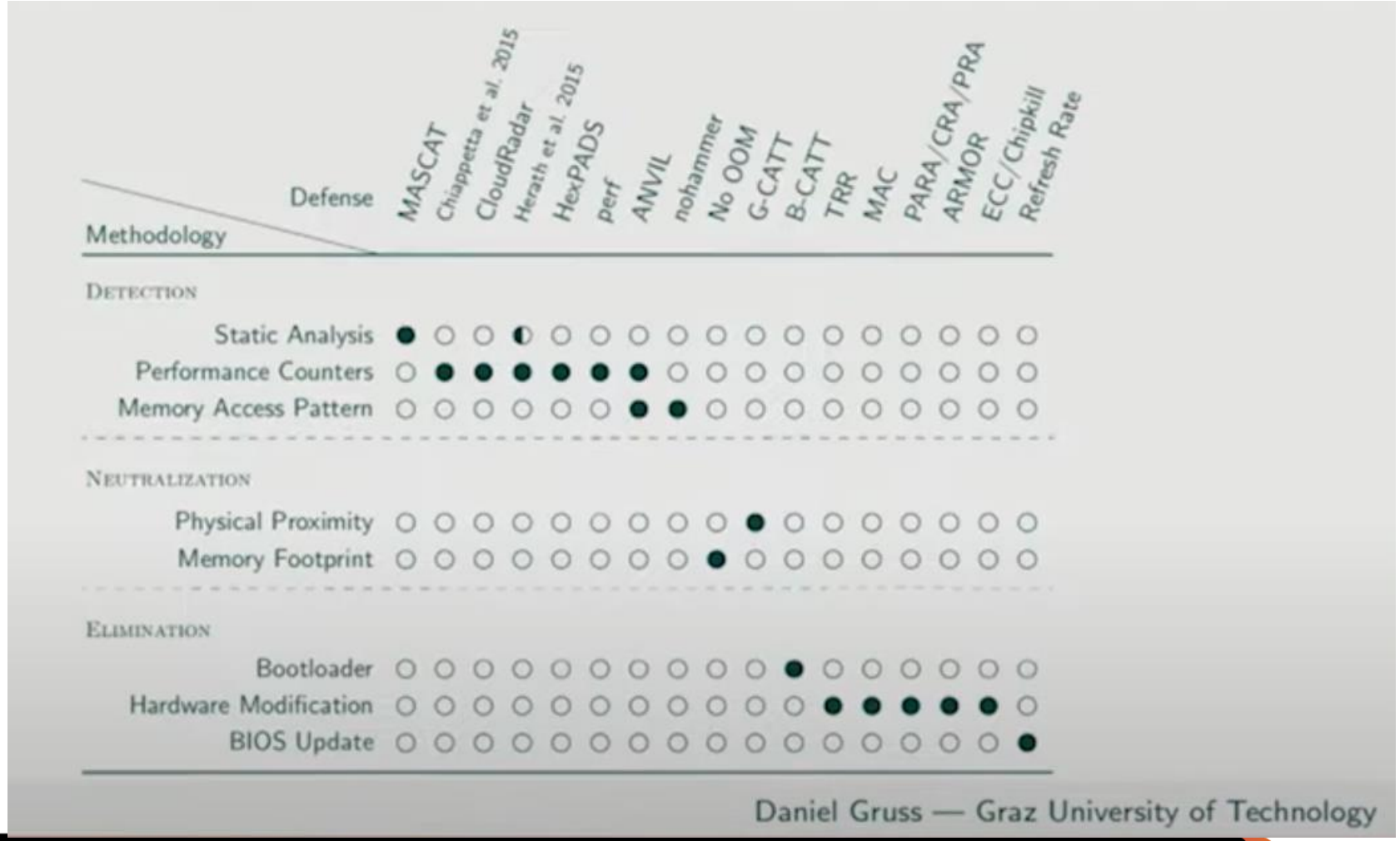
- Već smo ustanovili da je memorija usko grlo
- Razvoj tehnologije za “izgradnju” memorije u smislu latencije nije nešto sa čime se trebamo pohvaliti u zadnjih 3 desetljeća
- Što god da se moglo napraviti u smislu brzine, smanjenja cijene I povećavanja adresnog raspona (32, 64-bit) – to je bila cijena koju smo bili spremni platiti
- Kako napreduje razvoj memorije, memorijske ćelije su sve bliže I bliže, što ima direktan utjecaj na performanse
- Bilo bi “fora” kad bismo mogli *zaobići* osnove fizike, ali samo zato što to jaaaaako želimo, to ne znači da je to I moguće
- Različiti ciljevi dizajna I potrebe tržišta su nas doveli do situacije da postoje ranjivosti kao Spectre, Meltdown I RowHammer
- RowHammer je primjenjiv na sve vrste memorije, poglavito DDR3+ memorije
- Što je novija memorija, situacija je obično gora

Refresh problemi – performanse I potrošnja energije





Sustavi za detekciju, neutralizaciju I eliminaciju RowHamer ranjivosti



Spectre i Meltdown

Kakvi su to napadi?

Meltdown i Spectre napadi

- Netko nam može pokrasti tajne podatke sa sustava, iako:
  - Program I podaci rade u skladu sa pravilima
  - Hardver radi kako treba
  - Ne postoje softverske ranjivosti aplikacije ili operacijskog sustava

Meltdown i Spectre

- Hardverske ranjivosti koje su prisutne na manje-više svim računalnim čipovima koji su proizvedeni unatrag dva desetljeća
- Iskorištavaju spekulativno izvršavanje
  - Reminder: tehnologija koju moderni procesori koriste za dostizanje boljih performansi
  - Izvršava se dio koda prije nego što znamo da li je to potrebno
    - I sami to radimo vrlo često, jer to štedi vrijeme
    - I procesori to rade, iz istog razloga

Spekulativno izvršavanje

- Razmatrajmo ovaj kod

```
if (account-balance <= 0) {
    // do something
} else if (account-balance < 1M) {
    // do something else
} else {
    // do something else
}
```

Pogađa se koji kod će biti izvršen I izvršava se
- Poboljšanje performansi, zato što treba puno vremena za pristup memoriji

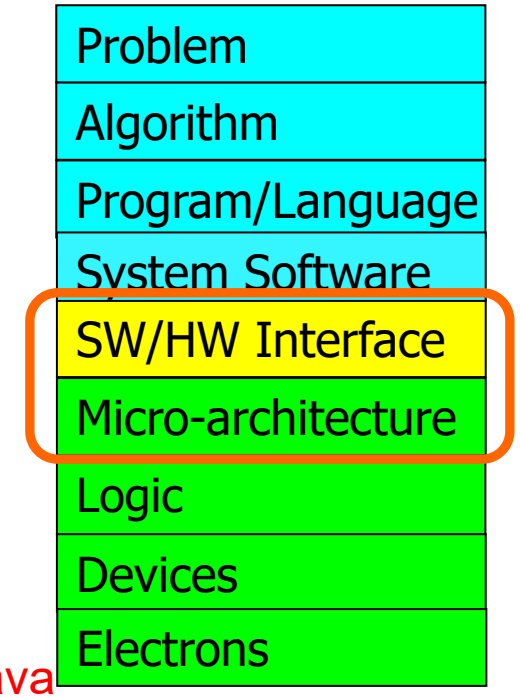
Ako je pogađanje pogrešno, brišu se krive instrukcije I izvršava ispravan kod

Korisnik ne vidi spekulativno izvršavanje

ISA
(Instruction Set Architecture)

Sučelje/”ugovor” između softvera I hardvera.

Programer pretpostavlja da hardware zadovoljava uvjete ISA-e..



Mikroarhitektura
Implementacija ISA-e

Programer(ka) također pretpostavlja da će kod biti izvršen KAKO JE NAPISAN (potpuno pogrešno)

Mikroarhitektura instrukcije izvršava Drugačijim redoslijedom (spekulativno) ali daje identične rezultate koje programer(ka) očekuje.

Meltdown and Spectre

- Netko nam može pokrasti tajne podatke sa sustava, iako:
  - Program I podaci rade u skladu sa pravilima
  - Hardver radi kako treba
  - Ne postoje softverske ranjivosti aplikacije ili operacijskog sustava

- Zašto?
  - Spekulativno izvršavanje ostavlja tragove tajnih podataka u cache memoriji procesora
    - Podaci kojima se ne bi trebalo pristupiti da nema spekulativnog izvršavanja
  - Maliciozni program može provjeravati sadržaj cache memorije da bi došao do zaključka gdje se nalaze tajni podaci kojima ne bi smio moći pristupiti
  - Maliciozni program također može prisiliti drugi program da spekulativno izvrši kod koji ostavlja tragove (tajne podatke)

Cache memorija procesora kao side-channel za napade

- Spekulativno izvršavanje ostavlja podatke u cache memoriji procesora
  - **Gledano iz aspekta dizajna I arhitekture, potpuno korektno ponašanje**
  - **Nuspojava je side-channel – kanal kroz koji netko sa “većim znanjem” može doći do tajnih podataka**

- Kroz cache memoriju “cure” podaci, pošto procesor u cache memoriju privremeno pohranjuje podatke sa kojima radi u fazi spekulativnog izvršavanja

Više o Spectre I Meltdown napadima

Project Zero

News and updates from the Project Zero team at Google

Wednesday, January 3, 2018

Reading privileged memory with a side-channel

Posted by Jann Horn, Project Zero

We have discovered that CPU data cache timing can be abused to efficiently leak information out of mis-speculated execution, leading to (at worst) arbitrary virtual memory read vulnerabilities across local security boundaries in various contexts.

https://googleprojectzero.blogspot.ch/2018/01/reading-privileged-memory-with-side.html

Hvala na pažnji!