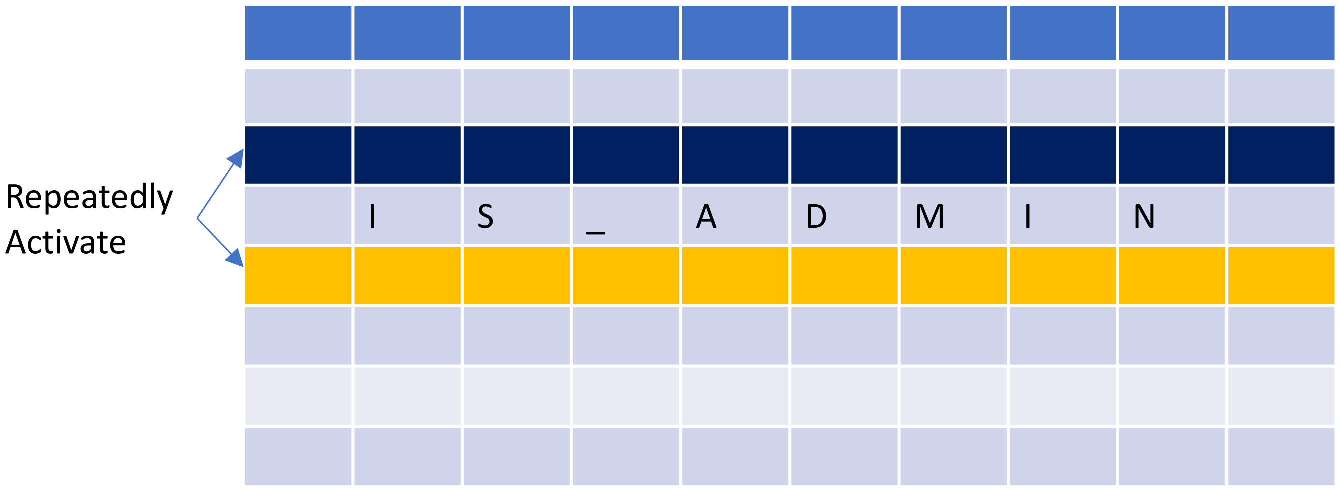
Fifth week tutorial

Yuvraj Patel

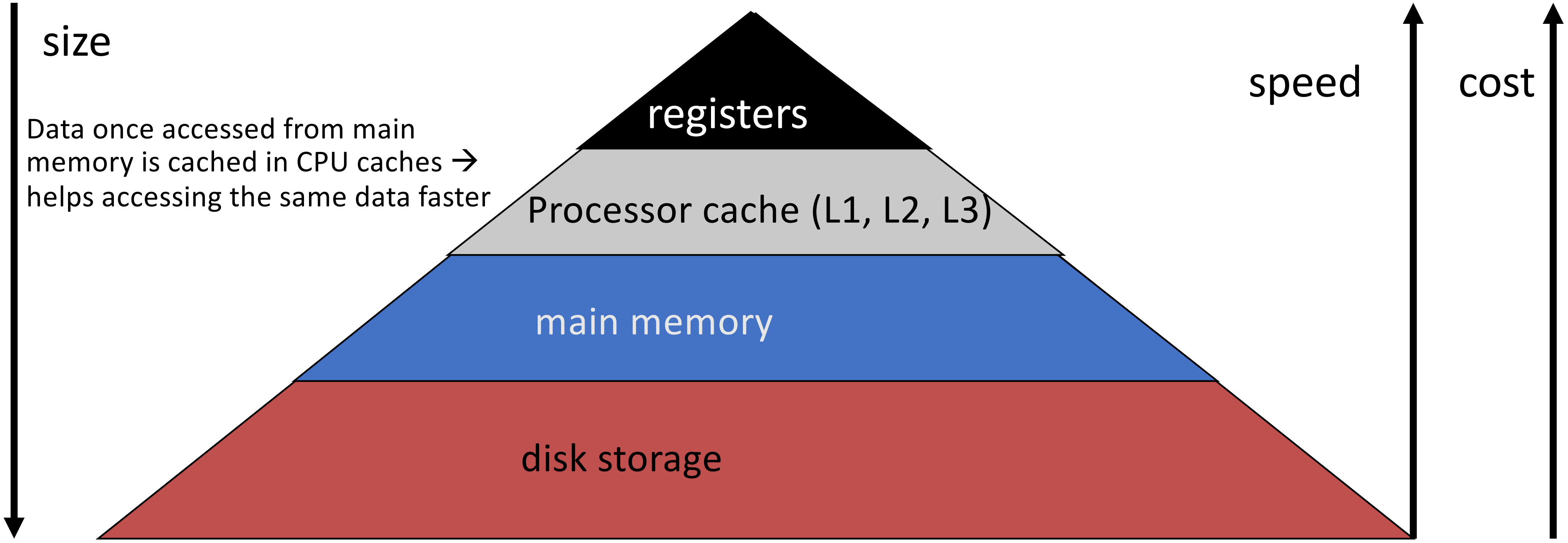
Question 1

A Rowhammer attack requires the rapid access of rows within the CPU’s Random Access Memory. However, all modern systems have caches that act as a fast buffer for frequently accessed memory. Does such a cache make a Rowhammer attack impossible, given that it typically reduces the rate of access to main memory?

Rowhammer



Memory Hierarchy



Data once accessed from main memory is cached in CPU caches → helps accessing the same data faster

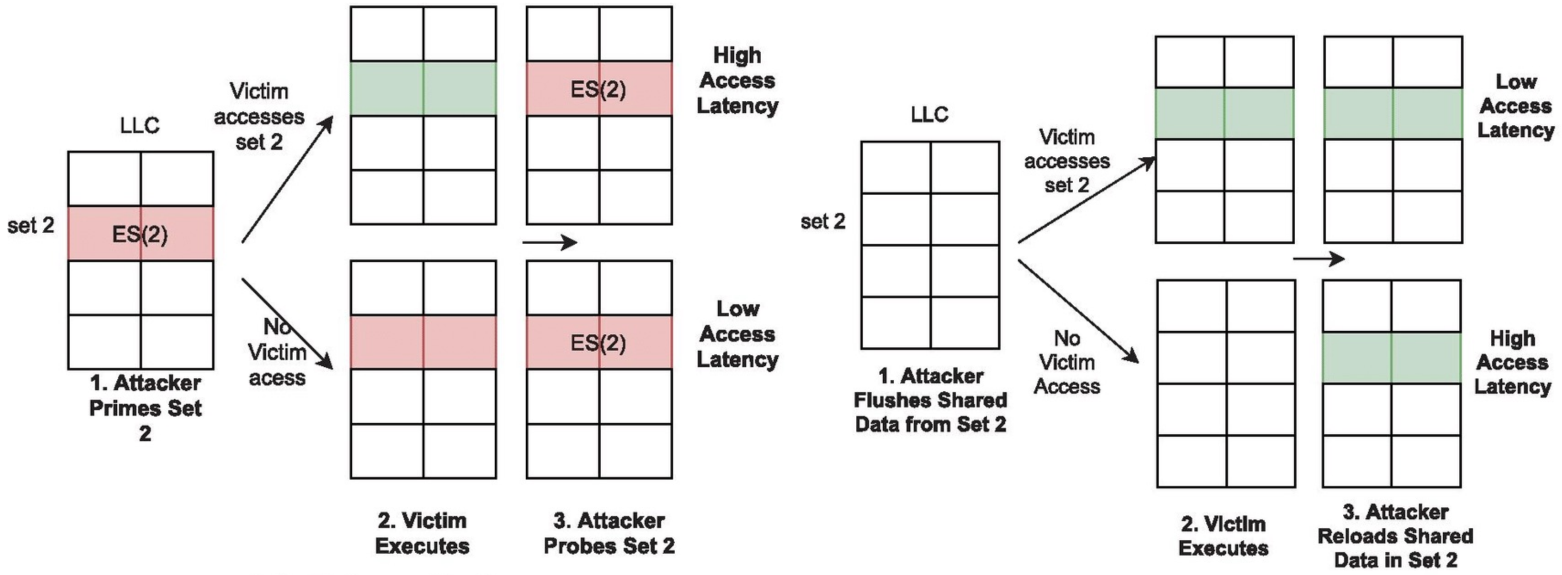
Question 1

A Rowhammer attack requires the rapid access of rows within the CPU’s Random Access Memory. However, all modern systems have caches that act as a fast buffer for frequently accessed memory. Does such a cache make a Rowhammer attack impossible, given that it typically reduces the rate of access to main memory?

- You must work out some way to bypass the cache, whether directly with a cache flush or indirectly with a suitable access pattern

Question 2

In side-channel attacks, what is the difference between “Flush and Reload” and “Prime and Probe”? Is one of these more general than the other?



(a) Prime+Probe

(b) Flush+Reload

Question 2

In side-channel attacks, what is the difference between ”Flush and Reload” and ”Prime and Probe”? Is one of these more general than the other?

- Prime-and-Probe is about forcing the victim to evict your data, thus increasing time on a measurement of the cache
- Evict-and-Reload is about measuring which lines the victim has brought in (and thus requires shared memory between victim and attacker), thus decreasing time of a load when the target line is re-accessed

Question 3

To eliminate Spectre attacks, your boss suggests you buy a processor without a cache. How effective do you think this might be, how feasible, and would there be any disadvantage?

Real-world example

Naïve way of making tea

1. Washing teacups (assuming dirty, only one teacup available)
2. Boil water and put it on table
3. Make tea
4. Drink

Real-world example

Naïve way of making tea

1. Washing teacups (assuming dirty, only one teacup available)
2. Boil water and put it on table
3. Make tea
4. Drink

Efficient way of making tea

1. Washing teacups (assuming dirty, only one teacup available)

< Parallelly wash cups and boil water; Out of order execution>

1. Boil water and put it on table
2. Make tea
3. Drink

Real-world example

Naïve way of making tea

1. Washing teacups (assuming dirty, only one teacup available)
2. Boil water and put it on table
3. Make tea
4. Drink

Efficient way of making tea

1. Washing teacups (assuming dirty, only one teacup available)

< Parallelly wash cups and boil water; Out of order execution>

1. Boil water and put it on table
2. Make tea
3. Drink

What if you break the ”only” teacup? What happens to the boiled water?

Question 3

To eliminate Spectre attacks, your boss suggests you buy a processor without a cache. How effective do you think this might be, how feasible, and would there be any disadvantage?

- Cacheless designs are way too slow
- In-order cores are Spectre proof even with a cache but still too slow
- Most buyers (other than majors like Google) have to take what they’re given

Question 4

Does SGX eliminate Spectre attacks? If so, how? If not, what does SGX achieve? Is SGX only beneficial in a data centre, rather than on end-consumer hardware?

Question 4

Does SGX eliminate Spectre attacks? If so, how? If not, what does SGX achieve? Is SGX only beneficial in a data centre, rather than on end-consumer hardware?

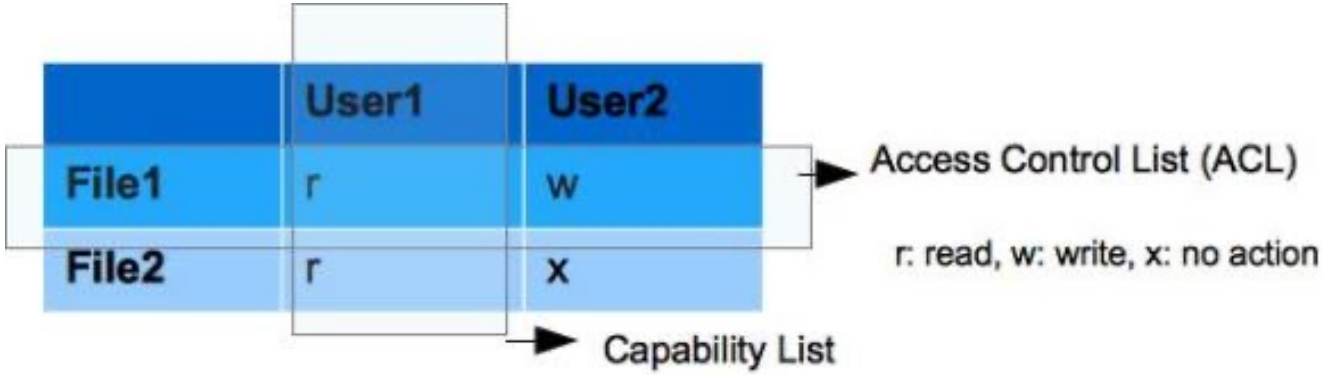
- No, and SGX hasn’t really been successful and is now marked as obsolete
- SGX was brought in for Blu-Ray disks in 2016 but consumer devices now use different tech such as TrustZone and Blu-Ray has been replaced by streaming services

Question 5

“All access control is just a matrix matching users to files, so capabilities and access control lists are exactly equivalent.” How accurate is this statement?

Question 5

“All access control is just a matrix matching users to files, so capabilities and access control lists are exactly equivalent.” How accurate is this statement?



	User1	User2
File1	r	w
File2	r	x

Access Control List (ACL)

r: read, w: write, x: no action

Capability List

Question 5

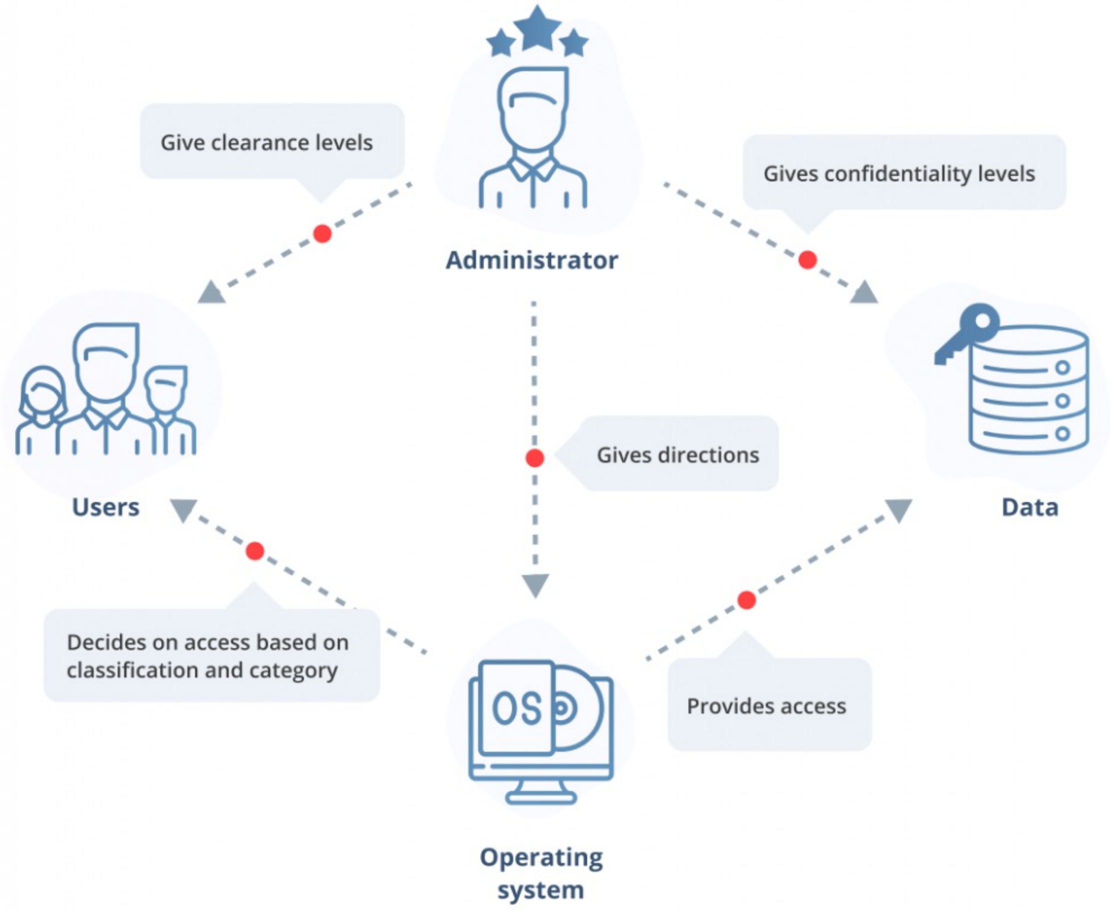
“All access control is just a matrix matching users to files, so capabilities and access control lists are exactly equivalent.” How accurate is this statement?

- There is a big performance difference
  - ACLs are convenient for storing and managing simple rules
  - Capabilities are better at runtime
- Then there are more complex rules
  - Sometimes you need (user, program, data) triples
  - Sometimes you need roles
  - Sometimes users and resources management different

Question 6

“The mandatory access control mechanisms in Android just serve to take control away from the user and place it in the platform holder’s hands. Any benefit to the consumer is just an illusion.” How accurate is this statement?

Mandatory Access Control (MAC)



Question 6

“The mandatory access control mechanisms in Android just serve to take control away from the user and place it in the platform holder’s hands. Any benefit to the consumer is just an illusion.” How accurate is this statement?

- Android (and iOS) have a lot less malware than Windows thanks to MAC
- There’s also the environmental hygiene of the app store ecosystem)

Question 7

Why would Apple only store biometrics in their own proprietary Secure Enclave, rather than using TrustZone and/or allowing access to the iOS kernel more generally? Does this really improve security?

Question 7

Why would Apple only store biometrics in their own proprietary Secure Enclave, rather than using TrustZone and/or allowing access to the iOS kernel more generally? Does this really improve security?

- The Secure Enclave is the equivalent of the TPM or Secure Element in an Android phone
- It’s a minimal design that doesn’t run any third-party software, unlike TrustZone