Spectre ve Meltdown Saldırıları

10.04.2021
Video

* 2018 yılı başında ortaya çıktı.
  ↳ 30 yıldır olan bir açıktır.
  ↳ Bu açığı kaldırmak başarımı azaltmıştır, yavaşlama meydana gelmiştir. (%30 oranında başarım düşüşü)

Meltdown → Intel mimarisi özelinde
Spectre → Tümünü kapsar

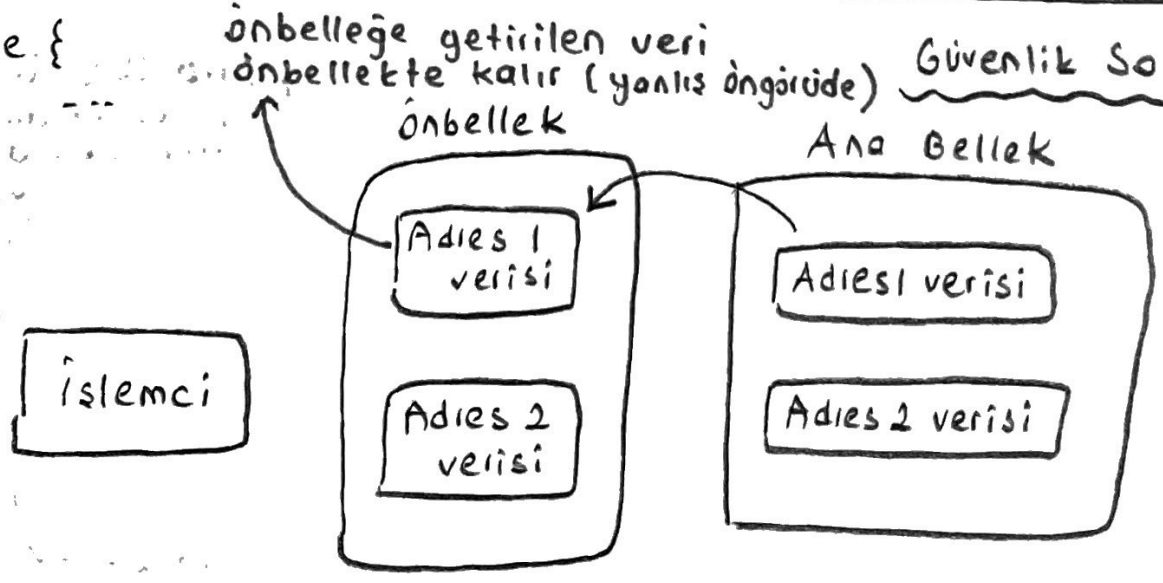
Spectre

Öngörüye Dayalı Yürütme ★

Öngörüye dayalı yürütme günümüz işlemcilerinde yüksek başarım için kullanılan bir yöntemdir.

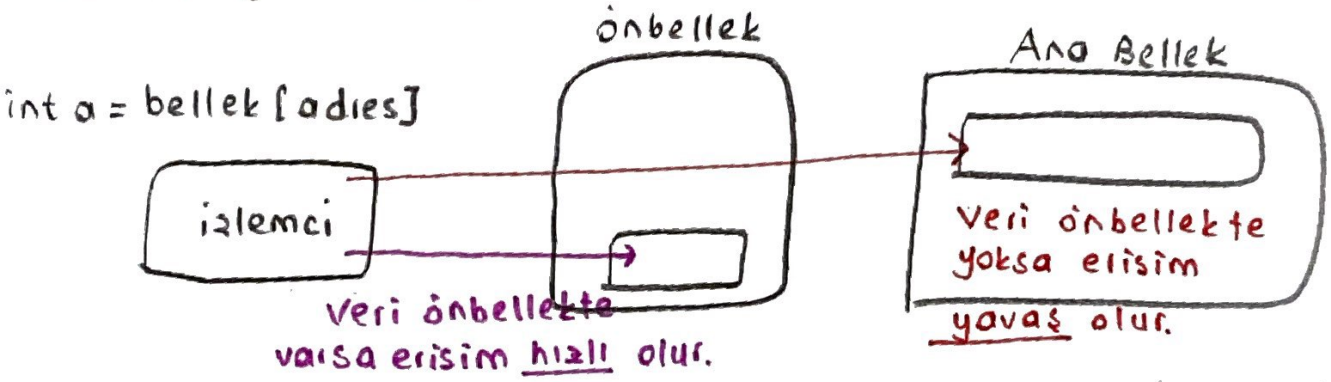
```
if (Kosul #1) {
    int a = bellek[adres1];
    ---
}
else if (Kosul #2) {
    int a = bellek[adres2];
    ---
}
else {
    ---
}
```

★ Kosul 1 yanlış görüldü!
↳ yanlış olduğunu anlayana kadar arkadan gelen buyrukları işlenir ve önbelleğe alınır.
↳ yapılan değişiklikler geri alınarak buyruk geçersiz kılınır fakat; önbelleğe gelen veri geçersiz kılınamaz.



Önbellek Saldırıları (Cache side channel attacks)

→ Önbelleğe erişim süresi kullanılarak yapılan saldırılara verilen genel addır.



→ Bir adrese eristiğimizde veri gelene kadar geçen süreyi ölçerek bu adresin verisinin önbellekte olup olmadığını anlaya biliriz.

Spectre Saldırı Mekanizması

1) Dallanma öngörücü istenen şekilde eğitilir.

```
if ( x < array1_size)
    y= array2 [array1[x] * 4096];
```

* x birçok kez array1_sizeden küçük olursa bu dallanmanın içindeki kod yürütülecek olarak öngörülür.

* Öncelikle küçük x değerleri verilir ve dallanma öngörücü hep doğru tahmin eder, daha sonra x'e çok büyük değer verilerek yanlış öngörü sağlanır.

→ Array1[x] değerine göre okunmaması gereken bir adres okunur.

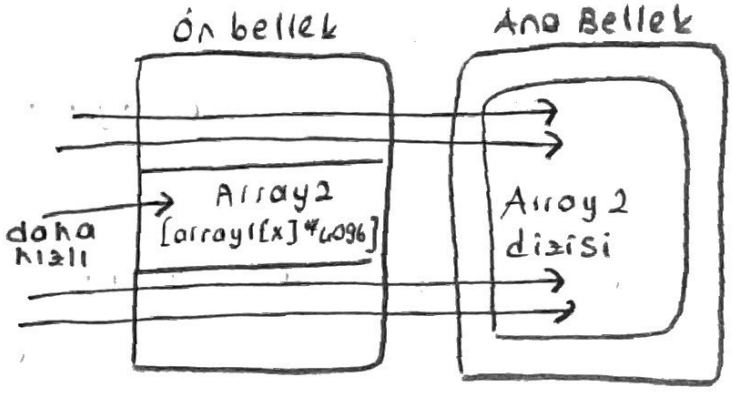
2) Daha sonra x değiştirilir.
→ Ama; hala önbelleğe erişim yetkisi bulunmaz!

3) Array2 dizisinin elemanlarına erişilir ve elemanlardan hangisinin önbellekten geldiği bulunur.

→ Bunun için; tüm dizi taranır.

→ Aranan veri önbellekte olduğundan dolayı, veriye daha çabuk erişilir. (Diğerleri ana bellekte)

```
for i=0 to array2_size:
  addr = &array2[i]
  time1 = __rdtscp();
  y = *addr;
  time2 = __rdtscp() - time1;
  if (time2 <= CACHE_HIT_THRESHOLD)
      // attack successfull
```



* istenilen veriye ulaşmak amacıyla zaman yaratmak için; dallanmanın uzun sürede çözülmesi için değişkenler belleğe atılır. Böylece daha çok beklenir. (Load işlemleri)

→ öngörüye dayalı sırasız yürütüm yapan tüm işlemciler Spectre saldırılarından etkilenmiştir.

Çözümler

Spectre bir saldırı sınıfı olarak tanımlandığı için tamamını çözmek mümkün değil.

→ önbellek saldırılarında kullanılan özel buyrukların engellenmesi ile çözülmeye çalışılmıştır.

→ Önbelleği boşaltan clflush buyruğunun kısıtlanması

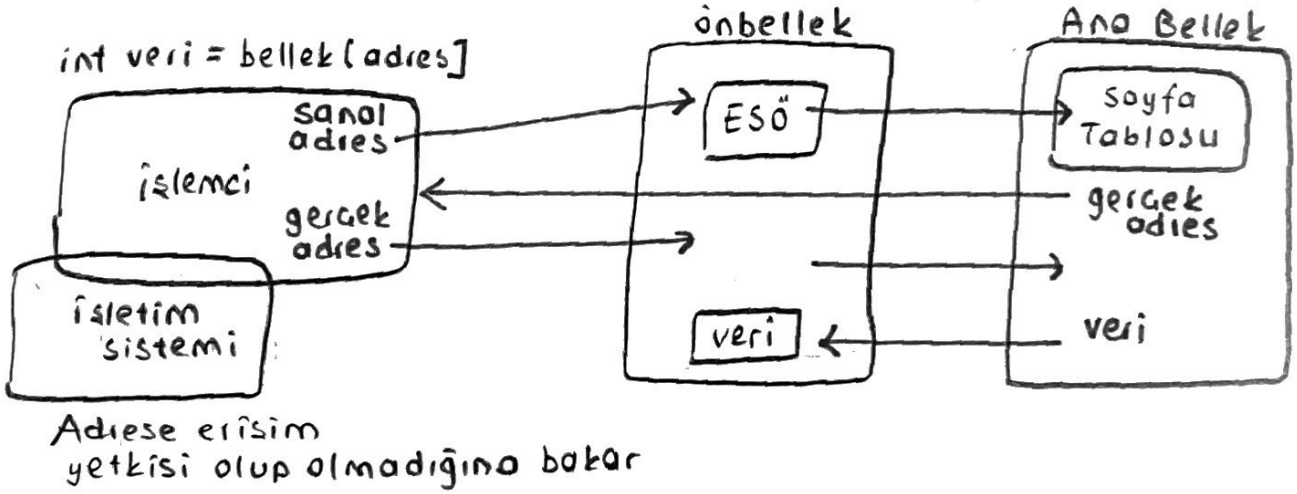
→ Bellek işlemleri tamamlanmadan yürütmeye devam edilmemesini sağlayan mfence buyruğunun kısıtlanması

Retpoline = Adresi hesaplanan dallanma buyruklarını kullanan saldırı vektörünü adres hesabını öne alarak durdurma çözümü (sadece bir versiyonu çözer)

* önbelleğe yapılan saldırılar ortadan kaldırılsa dahi başka bir bilgi sızdırma yolu ile saldırı gerçekleştirilebilir.

Meltdown → ESÖ ile ilgilidir.

→ Bir adrese erişim gerçekleşirken o sayfaya erişim izni olup olmadığına da bakılır.



→ İşletim sisteminin denetimi bitmeden veri önbelleğe getirilir.

* Eğer o sayfaya erişim izni yoksa hata verilir.

* Sayfaya erişim izni varsa veri okunur.

→ yine önbellekte bu verinin bulunması sebebiyle veri sızdırılabilir ?

10.04.2021

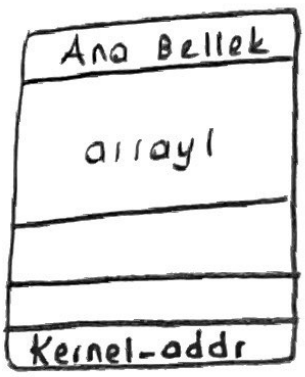
Meltdown saldırı mekanizması

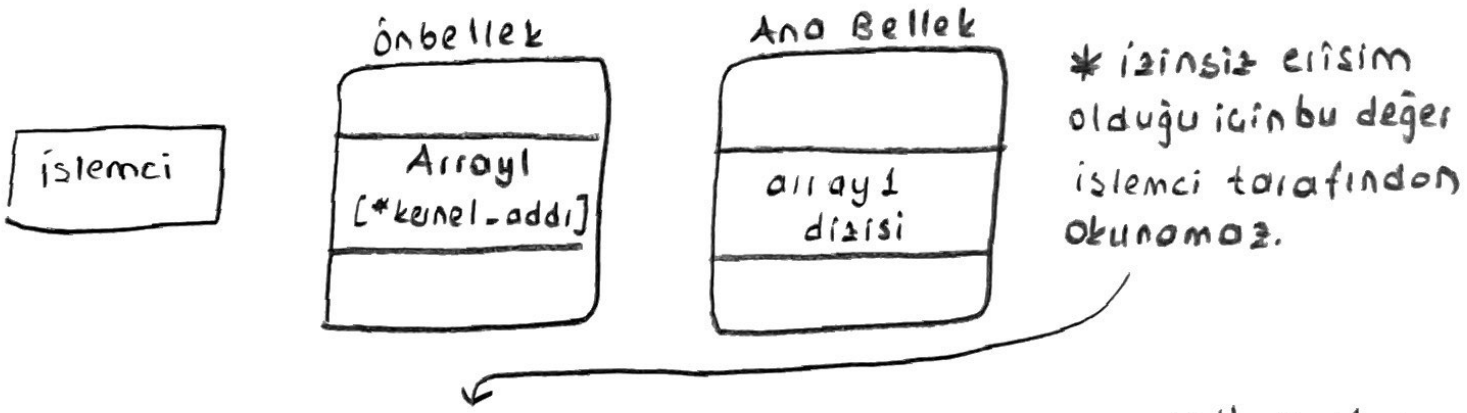
1) önbellek boşaltılır.

```
clflush (array1[tahmin]);
```

2) İstenen adresin okunması sağlanır.

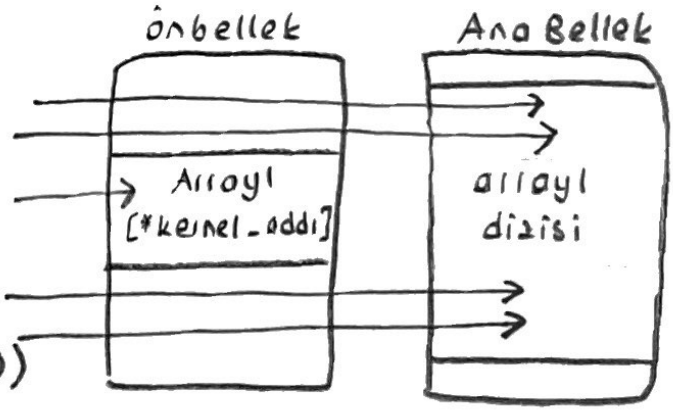
```
int a = array1[*kernel_addr];
```





Ama; önbelleğe array1'in bir elemanı getirilmiş olur.

3) önbellekte hangi verinin olduğuna bakılarak kernel_addr'teki veri bulunur.




```
mfence();
s = rdtsc();
int a = array1[tahmin];
e = rdtscp();
if (e-s < CACHE_MISS_THRESHOLD)
  printf("tahmin doğru!\n");
```

* Genel olarak **Intel** işlemcilerini ve **ARM** Cortex-A75 ve IBM Power'ı etkiledi.

→ AMD işlemcilerinde etkisi olmadığı bildirildi.

Çözümler

KAISER / KPTI (Kernel page-table isolation)

Kullanıcı ile kernel'ın kullandığı alanları izole eden bir yama.

↳ Donanımda bir değişiklik yapılmadı, işletim sistemi koduna ek yapıldı.

* Kullanıcı ile çekirdeğin sayfa tablolarını ayırır. Kullanıcı modundan çekirdeğe ya da tam tersi geçişlerde ESÖ'deki girdiler silinir.

→ Ortalama %5, en kötü durumda ise %30 başarım kaybına sebep olmuştur.