И. Сенновский

i.sennovskiy@bi.zone

Новая техника атак на основе Meltdown. Использование спекулятивных инструкций для детектирования виртуализации

1. **Введение**

Атака, представленная в данной работе, базируется на побочном канале по кешу, используемому в атаке Meltdown. Meltdown использует спекулятивное исполнение для доступа к памяти, которая не должна быть доступна для злоумышленника без специальных привилегий. Атака отличается от Meltdown, она не использует порог времени доступа к памяти через кеш. Это возможно, потому что процессор исполняет определенные инструкции заранее для ускорения выполнения кода. Meltdown утилизирует чтение из буферов, контролируемых злоумышленником, при спекулятивном исполнении таким образом, что злоумышленник может использовать замеры времени доступа к памяти в качестве побочного канала.

2. **Виртуализация**

Технология VT-x в процессорах Intel позволяет гипервизору выбрать, произойдет ли VMEXIT (переключение контекста на гипервизор) при определенных инструкциях, например ***rdtsc***. Большинство сред виртуализации в стандартной конфигурации настраивают перехват ***rdtsc*** по умолчанию, например Virtualbox, VMWare, Hyper-V, Parallels на гипервизоре от Apple и от Parallels. Поскольку VMEXIT фактически означает переключение контекста, то инструкции, которые генерируют VMEXIT, исполняются дольше, чем если бы они исполнялись в невиртуализованной среде.

3. **Атака**

Создается буфер размером в несколько страниц. Затем вместо спекулятивного доступа к областям памяти с целью получения данных спекулятивно исполняется инструкция ***rdtsc*** и результат её исполнения

используется для доступа к определенной части выделенного ранее буфера. При спекулятивном выполнении производится доступ только к определенной части выделенного буфера, что позже позволяет отличить случаи спекулятивного доступа от случайных ошибок. После завершения выполнения функции, содержащей спекулятивное исполнение кода, номер страницы памяти с самым низким временем доступа добавляется в статистику. После во всём буфере сбрасывается кеш. Ниже приведены функции, которые используются для срабатывания спекулятивного исполнения и доступа к памяти в 32-битных версиях Windows:

```
_declspec(naked) void herring() { //Эта функция используется для
	__asm {                          //срабатывания спекулятивного
		xorps xmm0, xmm0         //исполнения в функции speculate
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		sqrtpd xmm0, xmm0
		movd eax, xmm0
		lea esp, [esp+eax+4]
		ret
	}
}

_declspec(naked) void __fastcall speculate(const char* detector) {
	__asm {              //Эта функция спекулятивно исполняет rdtsc и
		Mfence       //обращается к странице, соответствующей
		mov esi, ecx //возвращенному rdtsc значению
		call herring
		rdtsc                                   //Эти инструкции
		and eax, 7                              //исполняются
спекулятивно
		or eax, 32                              //*
		shl eax, 12                             //*
		movzx eax, byte ptr [esi+eax]           //*
	}
}
```

Для успешной реализации атаки эти действия нужно повторять, чтобы найти распределение кэшируемых страниц. Необходимо выполнить столько повторов, чтобы набрать достаточно статистических данных: во время описываемого теста использовалось 10 000 итераций. Затем рассчитывается количество промахов мимо выбранного региона памяти. В виртуализованных средах, где включен перехват ***rdtsc***, доля таких промахов составляет от 50 до 99 процентов. На невиртуализованных системах она меньше одного процента. Эта информация представлена на рисунке 1 (чем темнее регион памяти, тем больше попаданий в него зафиксировано). При тестировании в качестве невиртуализованных систем использовались macOS, Ubuntu, Debian и Windows, а в качестве гостевых систем - Ubuntu, Debian и Windows. Аналогичная атака возможна при помощи инструкции ***rdmsr***.

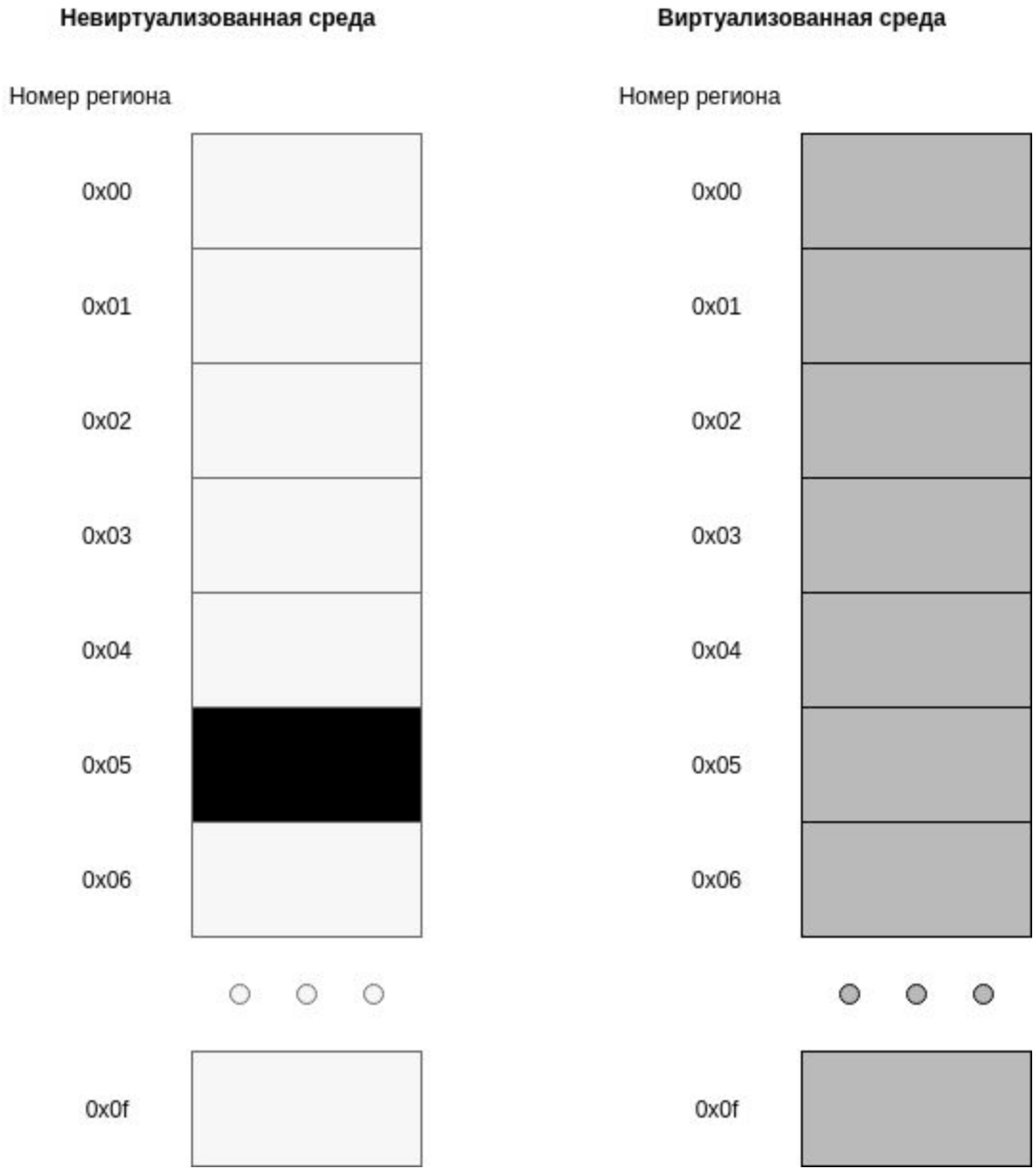

Рисунок 1 - Распределение кэшированных страниц в различных средах

4. **Объяснение атаки**

Атака использует спекулятивное исполнение инструкций, чтобы заставить процессор раскрыть информацию об исполнении ***rdtsc***. В невиртуализованной среде ***rdtsc*** исполняется на самом процессоре, который просто возвращает счетчик. В виртуализованной среде, где для бита «RDTSC exiting» выставлено

значение MSR IA32_VMX_PINBASED_CTLS, исполнение ***rdtsc,*** по сути, является переключением контекста, которое выполняется слишком долго.

На момент обнаружения уязвимости доступная внутренняя документация процессоров Intel не содержала данных, которые позволили бы точно объяснить, что происходит. Есть два предположения: либо процессор решает, что ***rdtsc*** будет выполняться слишком долго, и не исполняет его, пока ход исполнения не дойдет до него напрямую, либо все инструкции, которые вызывают VMEXIT, не исполняются спекулятивно. В виртуализованной среде ***rdtsc*** и инструкции, сразу следующие за ним, не выполняются спекулятивно, а в невиртуализованной - выполняются.

5. **Выводы и направления будущих исследований**

Описываемая атака использует новую технику кеширования на основе Meltdown для создания побочного канала, но вместо обращения к привилегированным регионам памяти она раскрывает информацию о режиме работы процессора. Существует несколько известных методов детектирования виртуализации, однако все они сильно зависят от инструкции ***rdtsc*** в качестве таймера, что позволяет «умному» гипервизору их обмануть, подменив возвращаемые значения. Такую атаку также можно ограничить, но если внести небольшие изменения в код, то подмена времени со стороны гипервизора не сможет повлиять на результат. Возможно, PoC такой версии будет опубликован позже.

Возможно сделать вывод, что в виртуализованных средах с перехватываемым ***rdtsc*** вариация атаки, введенной в этой работе, позволяет определить наличие виртуализации, а при отсутствии перехвата возможно использование ранее известных методов, например измерение скоростей работы с TLB кэшами.

Данная атака позволяет просто и быстро детектировать виртуализацию в средах со стандартными настройками или в средах, которые намеренно используют перехват ***rdtsc*** с целью защитить себя от детектирования

виртуализации. Эта атака успешно была протестирована на виртуализированной песочнице: эксперты обнаружили песочницу, не выдав себя.

PoC можно найти в репозитории https://github.com/bi-zone/rdtsc-checkvirt.