И.Сенновский
isennovskiy@gmail.com

Новая техника на основе Meltdown. Использование спекулятивных инструкций для проверки виртуализации

1. **Введение**

Атака Meltdown использует спекулятивное исполнения для доступа к памяти, которую в обычном случае непривилегированный злоумышленник не смог бы прочитать. Это возможно, потому что процессор исполняет определенные инструкции заранее для ускорения выполнения кода в целом. Meltdown использует попытки доступа к буферами под контролем злоумышленника при спекулятивном исполнении, изменяя состояние кэша CPU таким образом, что злоумышленник может использовать время запросов к памяти в качестве побочного канала.

2. **Виртуализация**

Технология VT-x в процессорах Intel позволяет гипервизору решить будут ли определенные инструкции перехватываться (произойдет ли VMEXIT), например инструкция ***rdtsc***. Большинство сред виртуализации включают перехват rdtsc по умолчанию, например, это делают Virtualbox, VMWare, Parallels с бэкэндом Apple и Parallels. Сам VMEXIT - это переключение контекста, что будет полезно знать дальше.

3. **Атака**

Первая часть атаки не отличается от Meltdown. Выделяется регион памяти, который будет использоваться для побочного канала. Замеряется время доступа к памяти с кэшем и без него, вычисляется пороговое значение для будущих тестов.

Далее атака отличается от Meltdown. Вместо спекулятивного чтения регионов недоступной злоумышленнику памяти спекулятивно исполняется ассемблерная инструкция ***rdtsc***, после чего результат выполнения этой инструкции из регистров ***EDX:EAX*** ограничивается тремя нижними битами ***EAX*** и производится чтение соответствующей страницы региона памяти выделенного

ранее со смещением в 32 страницы. Таким образом, при успешном выполнении спекулятивных инструкций, будет кэшироваться одна из 8 страниц (всего их 256). Номер страницы с наименьшим временем доступа добавляется в статистику. В конце сбрасывается весь кэш данных страниц.

Для успешной атаки эти действия повторяются много раз (в нашем случае 10000). В итоге производится подсчет количества раз, когда минимальное время доступа было у любой другой страницы, кроме выбранных восьми. В виртуализованной среде, где ***rdtsc*** приводит к VMEXIT, доля случаев доступа к остальным страницам составляет 50-90%. На не виртуализованных системах она меньше процента. При тестировании использовались Windows, Mac OS, Ubuntu, Debian не под виртуализацией и Windows, Ubuntu И Debian под виртуализацией.

4. **Объяснение атаки**

При исполнении CPU решает, что rdtsc в виртуальной среде будет слишком долго выполняться и игнорирует ветвь исполнения, пока не будет точно знать, что ветвь должна выполнится.

5. **Выводы и дальнейшая работа**

Уже существует несколько методов, позволяющих определить виртуализацию, но они в основном опираются на засечении времени выполнения инструкций при помощи rdtsc, что позволяет гипервизору подделать время выполнения инструкций. Хоть данная атака в простой форме так же страдает от этой уязвимости, при небольшом изменении можно её исключить. Если постоянно менять подмножество страниц, которые должны быть закэшированы псевдослучайным образом, то гипервизор не сможет узнать, для каких страниц нужно подделывать время доступа. В дальнейшем будет создан POC, в котором будет использоваться это свойство.

Это ведет к интересному выводу: в случае перехвата ***rdtsc***, который позволяет обмануть тесты, придуманные ранее, данный метод позволяет определить наличие виртуализации, а если перехват выключен, то определение

виртуализации возможно другими методами, например, при помощи сброса TLB кэшей.