Домашняя работа №6

**Spectre**

**Цель работы:** знакомство с аппаратной уязвимостью Spectre.

**Инструментарий и требования к работе:** рекомендуется использовать C, C++.

**Задание и варианты**

Необходимо прочитать данные из некоторого региона памяти без прямого обращения к нему используя уязвимость Spectre v1.

Код должен стабильно читать данные на современных процессорах Intel и AMD.

В качестве <данных> выступает строка символов, которая записывается в некоторую область памяти (статический, динамический массив), к которой дальше нет обращений в программе.

Вывод: результат чтения данных через Spectre v1.

Формат вывода свободный. Всё, что выводит программа, необходимо пояснить в практической части.

**Вариант 1. Своя функция**

В качестве уязвимого кода использовать свою функцию.

**Вариант 2. Системная функция**

Использовать в качестве уязвимого кода какую-то системную функцию или из стандартной библиотеки.

Код уязвимой функции нужно приложить к коду написанной программы (достаточно фрагмента, который используется атакующим кодом, но с сохранением компилируемости и работоспособности атаки).

**Содержание отчета**

1. Теоретическая часть: описание уязвимости Spectre/Meltdown;

2. Описание работы написанного кода (пункт 2 из Порядка выполнения, экспериментальная часть);

3. Листинг кода с указанием компилятора/интерпретатора (подробнее **Оформление кода в отчёте**).

**Примечания:**

1.    Файл с отчётом подгружаем в саму форму: <https://vk.cc/bWUBaZ>;

2.  В поле «Ссылка на отчет» требуется ссылка на ответ, а не на диск, где лежит отчет. Данное поле заполняется, если по каким-либо причинам не удалось приложить файл в форму. Также можно подстраховаться на случай, если файл приложится некорректно, заполнив это поле. Если ссылка на отчет приложена неверно и в форме нет приложенного файла, то отчет не принимается;

3. В поле «Ссылка на код» можно вставлять как ссылку на файл (исходного кода либо архив с исходниками), так и на диск (где именно вы будете хранить файлы не принципиально, главное – открытый доступ по ссылке до обозначенного времени) или репозиторий (git – репозиторий должен быть закрытым и расшаренным со мной (RonoveRaum));

4.    «Шаблон отчета»:<https://vk.cc/aAWqZm>;

5. **Важно:** будет оцениваться как правильность реализации, так и стабильность получаемых результатов на разном железе.

**Дополнительные сведения (код)**

1.    Аргументы программе передаются через командную строку:

**hw6.exe <данные> [<имя\_выходного\_файла>]**

Если указано последнее, то результат работы пишется в этот текстовый файл.

2. Корректно выделяется и освобождается память, закрываются файлы, есть обработка ошибок: не удалось открыть файл, формат файла не поддерживается.

Если программе передано значение, которое не поддерживается – следует сообщить об ошибке;

3.    В программе можно вызывать только стандартные библиотеки (например, <bits/stdc++.h> таковой не является и ее использование влечет за собой потерю баллов). То есть сторонние библиотеки использовать нельзя.

4.    Если программа использует библиотеки, которые явно не указаны в файле с исходным кодом (например, <algorithm>), то за это также будут снижаться баллы.

**Оформление кода в отчёте**

1. Никаких скринов кода – код в отчет добавляется только текстом;

2. Шрифт: Consolas (размер 10-14 на ваше усмотрение);

3. Выравнивание по левому краю;

4. Подсветка кода допустима. Текст должен быть читаемым (а не светло-серый текст, который без выделения на белом не разобрать);

5. В раздел Листинг код вставляется полностью в следующем виде:

**<Название файла>**

<Его содержимое>

Файлы исходных кодов разделяются новой строкой.

Например,

**main.cpp**

int main()

{

return 0;

}

**tmain.cpp**

int tmain()

{

   return 666;

}

6. Фон белый (актуально для тех, у кого копипаста кода идет вместе с фоном темной темы из IDE).