DOI 10.37882/2223–2966.2022.11–2.06

ИССЛЕДОВАНИЕ СКРЫТОГО КАНАЛА ПЕРЕДАЧИ ДАННЫХ, ФОРМИРУЕМОГО МАГНИТНЫМ ПОЛЕМ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

INVESTIGATION OF THE HIDDEN DATA TRANSMISSION CHANNEL FORMED BY THE MAGNETIC FIELD OF THE CENTRAL PROCESSOR

A. Vasiliev
S. Ryzhikov
I. Agureev
B. Zagartdinov

Summary. This article discusses a hidden magnetic data transmission channel between a computer isolated from the external environment, into which a malicious program is embedded, and a nearby smartphone that acts as a receiver. Based on this study, it was revealed that the malware regulates the workloads on the processor cores and thus, by generating different levels of the magnetic field from the central processor, encodes the transmitted data.

Keywords: malicious program, hidden data transmission channel, information leakage.

Васильев Андрей Савельевич
Старший преподаватель
Национальный исследовательский университет «МЭИ»
universe@mpei.ac.ru
Рыжиков Сергей Сергеевич
Национальный исследовательский университет «МЭИ»
universe@mpei.ac.ru
Агуреев Иван Александрович
Национальный исследовательский университет «МЭИ»
universe@mpei.ac.ru
Загартдинов Булат Назимович
Национальный исследовательский университет «МЭИ»
me@vaire.lt

Аннотация. В данной статье рассмотрен скрытый магнитный канал передачи данных между изолированным от внешней среды компьютером, в который внедрена вредоносная программа, и расположенным рядом смартфоном, который выполняет функции приемника. На основании данного исследования выявлено, что вредоносная программа регулирует рабочие нагрузки на ядра процессора и тем самым, формируя разные уровни магнитного поля от центрального процессора, кодирует передаваемые данные.

Ключевые слова: вредоносная программа, скрытый канал передачи данных, утечка информации.

Введение

Современные процессоры компьютеров являются, как правило, многоядерными и энергозатратными, то есть мгновенная нагрузка на процессор напрямую влияет на динамические изменения его энергопотребления. Регулируя нагрузку на центральный процессор (ЦП), можно влиять на его энергопотребление и, следовательно, контролировать уровень создаваемого магнитного поля. Преднамеренно запуская и останавливая рабочую нагрузку ЦП, можно генерировать магнитное поле на необходимой частоте. [1, 2]

Чувствительным элементом, фиксирующим уровень магнитного поля, может выступать магнитный датчик смартфона. Лежащий на столе около ноутбука или системного блока смартфон выглядит достаточно безобидно. Магнитные датчики (магнитометры) интегрированы практически во все современные смартфоны и планшеты. Их основная функция заключается в измерении магнитных полей Земли с целью определения ориентации телефона в трехмерном пространстве. Типичный магнитометр содержит три магнитных датчика (для трех перпендикулярных осей X, Y и Z). Чипы магнитометра обычно включают в себя встроенный акселерометр, который помогает регулировать измерения магнитного датчика. В отличие от оборудования сотовой связи, Wi-Fi, Bluetooth и NFC магнитные датчики не считаются интерфейсами связи и к ним можно получить доступ с основными разрешениями.

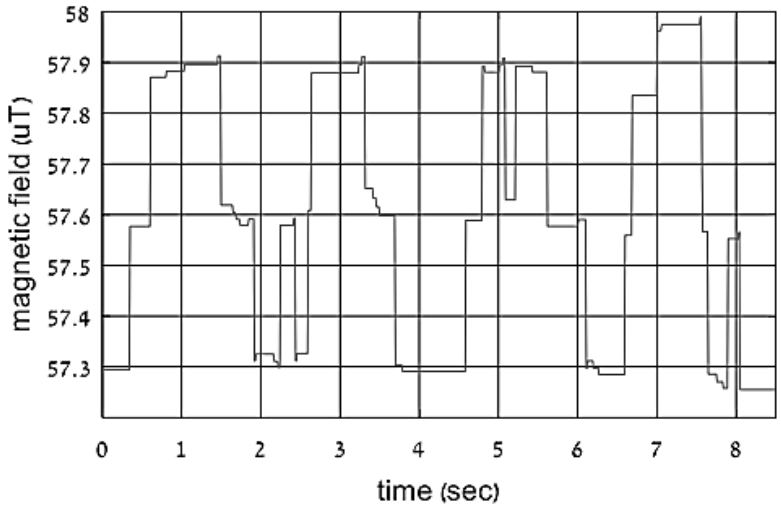

Рис. 1. Модуляция двоичной последовательности 10101010 посредством изменения загрузки ЦП.

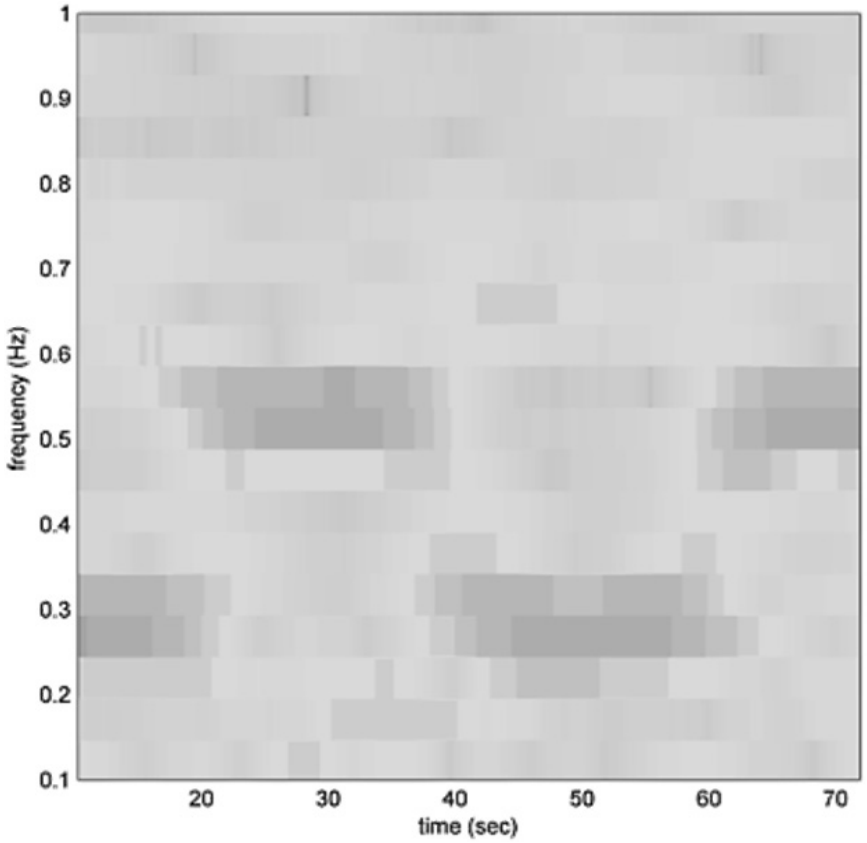

Рис. 2. Модуляция двоичной последовательности 1010 двумя частотами (0,25 Гц и 0,5 Гц).

Проблема нарушения конфиденциальности с использованием различных датчиков в мобильных устройствах не является новой. Данные акселерометра могут быть использованы для вычисления геолокации пользователя на основе специальных алгоритмов обработки, в обход запрета на использование GPS в мобильном приложении. [3]

В [1] приводятся два подхода к организации скрытого канала передачи данных, которые основаны

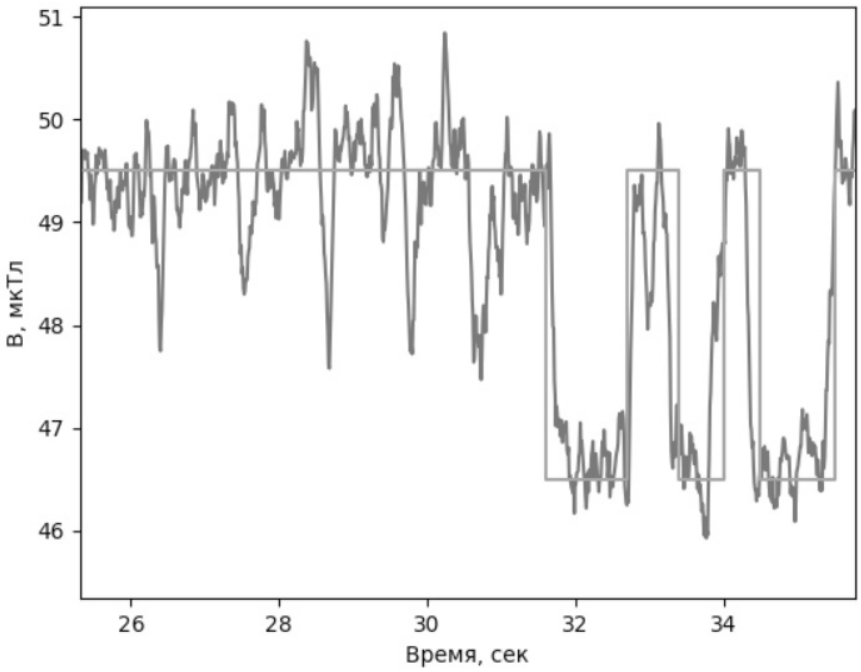

Рис. 3. Зависимость магнитного поля от модулирующей нагрузки

на наличии/отсутствии изменяемого магнитного поля с некоторой частотой (рис. 1) и на частотной манипуляции, когда рабочие нагрузки на различные ядра ЦП стандартного настольного ПК формируются с разными частотами (рис. 2).

Принимающий смартфон находился на расстоянии 5 см от передающего компьютера, при этом уровень шума составляет ~ 57,3 мкТл, а логической «1» ~ 57,9 мкТл. Передача данных осуществлялась короткими пакетами, содержащими преамбулу для определения факта начала передачи на фоне шумовой составляющей магнитного поля, непосредственно данные и контрольную сумму для обнаружения ошибок.

Эксперимент

Представляет интерес исследование характеристик данного скрытого канала передачи данных с использованием ноутбука в качестве передатчика в разнородном магнитном поле.

Поскольку максимальное расстояние передачи ограничено, применение сложных методов модуляции нецелесообразно, так как с увеличением сложности модуляции уменьшается различие уровней амплитуд сигнала из-за помех и инерционности влияния на магнитное поле. Физическое ограничение максимального расстояния между передатчиком и приемником усугубляется влиянием дополнительных источников магнитного поля, в том числе со стороны компонентов самого ноутбука.

В период ожидания, в отличие от максимальной загруженности, нагрузка на ЦП и магнитное поле вблизи него нестабильны. На рисунке 3 представлена зависимость величины магнитного поля от модулирующей нагрузки.

В качестве модулирующей функции выбрана широтно-импульсная модуляция, где логическая «1» представлена более широким импульсом, а «0» — более коротким. Дополнительно скорость передачи данных может быть увеличена за счет кодирования пар или троек бит в импульсы различной ширины.

После выбора модулирующей функции, необходимо формализовать метод демодуляции — трансформацию аналогового сигнала, получаемого с датчика магнитного поля, обратно в цифровой при помощи математических преобразований.

Формулу скользящего среднего по n элементам из множества значений показаний P датчика магнитного поля для сглаживания шумов в выборке можно представить как

$$F_{\text{ср}}: \langle N_0, Q^n, N, P \rangle \rightarrow Q, F_{\text{ср}}(t) = \begin{cases} \frac{\sum_{i=0}^{n-1} w_i \cdot p_{t-i}}{\sum_{i=0}^{n-1} w_i}, t \geq n, \\ p_0, t < n; \end{cases} \quad (1)$$

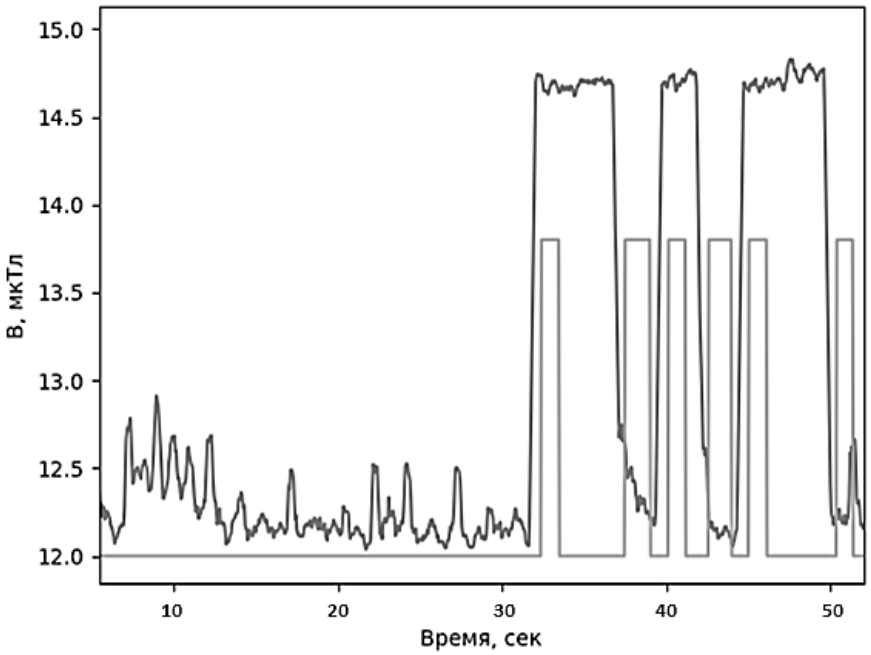

Рис. 4. Вид принимаемого сигнала до и после процедур фильтрации и декодирования.

где t — номер измерения от начала старта измерений,

p_i — i-й элемент из множества значений показаний P датчика магнитного поля,

w_i — весовой коэффициент i-го значения из вектора весов W,

n — размерность окна усреднения.

Для упрощения расчётов в (1) все весовые коэффициенты могут быть приняты за 1

$$F_{\text{ср}}(t) = \begin{cases} \frac{1}{n}\sum_{i=0}^{n-1} p_{t-i}, t \geq n-1, \\ p_0, t < n-1. \end{cases} \quad (2)$$

В последующем единичный весовой вектор может быть заменен более подходящим в зависимости от статистических наблюдений.

После усреднения значений требуется выделить изменения магнитного поля, используемые для кодирования — отношение приращения функции $\Delta F_{ср}$ к приращению аргумента Δt. Для масштабирования результатов используется возведение в квадрат

$$F_{\text{пр}}(t) = k \cdot \theta\left(\left(\frac{\Delta F_{\text{ср}}(t)}{\Delta t}\right)^2 - k\right), \quad (3)$$

где θ — функция Хевисайда;

k — пороговое значение детектирования изменения сигнала.

После преобразования входного сигнала необходимо повторно провести процедуру фильтрации шумовых импульсов для извлечения полезного сигнала

$$F_{\text{рез}} = \min_X F_{\text{пр}}, \quad (4)$$

где X — множество последовательных промежутков времени.

На рисунке 4 приведен график изменения формы сигнала до (синий цвет) и после его преобразования (оранжевый цвет).

Предлагаемая процедура декодирования сводится к детектированию промежутков времени между парами импульсов. Первая пара импульсов (между 30 и 40 секундами) кодирует значение «1», а следующая пара импульсов кодирует «0». Промежуток между этими парами импульсов в процедуре декодирования не участвует. Непосредственно перед передачей данных необходимо проводить калибровку принимающего устройства с использованием заранее известной последовательностью битов. За счет такой процедуры может быть достигнута универсальность относительно уникальных характеристик переходных процессов каждой ПЭВМ.

Рассмотренный метод демодуляции имеет несколько недостатков:

```
class burn_thread {
    std::mutex mutex_;
    std::unique_lock<std::mutex> lock;
public:
    burn_thread():lock(mutex_) {}
    void burn_cpu() {
        while (true) {
            cv.wait(lock);
            while (work) {
                int k = 123;
                volatile int data = 12345678;
                data /= (1234 + k);
            }
        }
    }
};
```

Рис. 5. Реализация потока для загрузки процессора ноутбука

```
void signals()
{
    while (true) {
        for (auto msg : message) {
            for (int j = 7; j >= 0; j--) {
                std::this_thread::sleep_for(std::chrono::milliseconds(DELAY));
                work = true;
                cv.notify_all();
                if (msg[j] == 1) {
                    std::this_thread::sleep_for(std::chrono::milliseconds(2 * DELAY));
                }
                else {
                    std::this_thread::sleep_for(std::chrono::milliseconds(DELAY));
                }
                work = false;
                cv.notify_all();
                std::cout << msg[j];
            }
            std::cout << std::endl;
        }
    }
}
```

Рис. 6. Реализация управляющего потока

```
148         if(one_tick == 0){
149             one_tick = time_values.get(0);
150             zero_tick = time_values.get(2);
151             if(Math.abs(one_tick-time_values.get(4)) > one_tick/5 ||
152                     Math.abs(one_tick-time_values.get(8)) > one_tick/5 ||
153                     Math.abs(one_tick-time_values.get(12)) > one_tick/5){
154                 time_values.remove(0);
155                 one_tick = 0;
156                 zero_tick = 0;
157                 return;
158             }
159             if(Math.abs(zero_tick-time_values.get(6)) > zero_tick/5 ||
160                     Math.abs(zero_tick-time_values.get(10)) > zero_tick/5 ||
161                     Math.abs(zero_tick-time_values.get(14)) > zero_tick/5){
162                 time_values.remove(0);
163                 one_tick = 0;
164                 zero_tick = 0;
165                 return;
166             }
167             if(Math.abs(zero_tick - one_tick) < zero_tick/5){
168                 time_values.remove(0);
169                 one_tick = 0;
170                 zero_tick = 0;
171                 return;
172             }
173             time_values.remove(0);
174             time_values.remove(0);
175             txtView.append("Started message\n");
176             return;
177         }
178         if(Math.abs(one_tick-time_values.get(14)) < one_tick/5){
179             data.add(true);
180             txtView.append("1");
181         }
182         else if(Math.abs(zero_tick-time_values.get(14)) < zero_tick/5){
183             data.add(false);
184             txtView.append("0");
185         }
```

Рис. 7. Фрагмент кода основного блока демодуляции

- трудность обнаружения сигнала близкого к уровню шума;
- сливание соседних пар импульсов, при более короткой паузе между модулирующими последовательностями;
- привязанность окна усреднения, порогового значения детектирования сигнала и других параметров к конкретной выборке данных.

Однако описанные недостатки компенсируются простотой реализации метода, а также минимальностью вычислительной нагрузки на принимающее устройство и независимостью от общего уровня магнитного поля.

Для управления генерацией модулирующей нагрузки на ЦП была разработана программа на C++. В качестве основного алгоритма нагрузки выбран бесконечный цикл с дополнительным вычислением произведения, суммы и деления (рис. 5).

Основную нагрузку в описанном алгоритме на центральный процессор оказывает операция цикла. Поскольку наиболее сложной для современных процессоров командой является операция ветвления, для частичной компенсации этой проблемы используется дополнительный механизм прогнозирования операций ветвлений (branch prediction).

Программа использует эту особенность для повышения энергопотребления процессора. Вследствие цикличности выполняемых операций, механизм предсказания ветвлений заполняет свободные ячейки конвейера, позволяя дополнительно загружать его повторяющимися операциями. Кроме того, эффект значительно усиливается благодаря отсутствию дополнительных точек прерывания выполнения программы в виде каких-либо операций чтения/записи или ожидания.

Описанные выше факторы позволяют операционной системе планировать запущенные потоки так, что ядро, используемое для вычислений, остается в активной фазе все время в процессе нагрузки.

Разработанная реализация алгоритма масштабируется на необходимое количество потоков и контролируется одним управляющим потоком, при помощи примитива синхронизации ***condition_variable*** (рис. 6).

В качестве базовой единицы модуляции используется константа времени DELAY. В процессе своей работы управляющий поток приостанавливается на соответствующее, кратное константе, время.

Реализация управляющей нагрузкой программы позволила автоматизировать генерацию модулирующих сигналов и абстрагироваться от платформы, поскольку выбранный язык программирования и его встроенные библиотеки являются кроссплатформенными.

В качестве устройства приема сигнала может быть использован мобильный телефон со встроенным датчиком магнитного поля, для доступа к которому в современных смартфонах на базе ОС Android не требуется специальных привилегий. Преимуществом возможности использования мобильного телефона в качестве принимающего устройства является обыденность расположения смартфонов вблизи ПЭВМ, что в свою очередь позволяет компенсировать ограниченное расстояние передачи информации.

Для приема и декодирования сигнала от ноутбука разработано приложение для смартфона на базе ОС Android. Алгоритм декодирования сигнала поступающего с датчика магнитного поля смартфона был реализован на основе описанной ранее модели демодулятора, параметры подобраны на основе выборки измерений магнитного поля нескольких ноутбуков в режиме ожидания и под модулирующей нагрузкой:

- выбрано окно усреднения n=30,
- количество последовательных измерений X = 15,
- пороговое значение детектирования k = 150.

Обработка данных с датчика магнитного кода происходит поэтапно. Собирается необходимое для усреднения количество измерений. Усредненные значения записываются в массив, который используется для вычислений отношения приращения аргумента к значению входных параметров. Поскольку частота магнитного датчика фиксированная, вместо времени используется привязка к количеству считываний, одно измерение принимается за условную единицу. В соответствии с моделью приращение возводится в степень и к результату применяется функция активации.

Фрагмент кода, реализующего функционал процедуры демодуляции представлен на рис. 7.

Скорость передачи данных в проводимом эксперименте составляла 1 бит/сек при удалении смартфона от ноутбука на 10 см.

Заключение

Подтверждено существование скрытого магнитного канала передачи данных путем манипуляции нагрузкой на ядра ЦП ноутбука и разработан прототип специального программно-аппаратного комплекса, моде-

лирующего реализацию данной угрозы. Разработано специализированное ПО на основе языка C++ с использованием вспомогательных библиотек для управления величиной загрузки ядер ЦП и на основе языка Java — для декодирования модулированного сигнала из принимаемой последовательности показаний магнитного датчика. Эффективность канала передачи данных путем регулирования величины нагрузки на ядра ЦП зависит от количества ядер ЦП, вида применяемой манипуляции, непосредственно величины загрузки центрального процессора и чувствительности применяемого датчика, фиксирующего уровень изменений магнитного поля.

Представляется перспективным использование внешнего магнитного датчика с характеристиками, превышающими параметры магнитометра, встроенного в смартфон.

ЛИТЕРАТУРА

1. M. Guri. “Magneto: Covert channel between air-gapped systems and nearby smartphones via cpu- generated magnetic fields,” Future Generation Computer Systems, vol. 115–2021 — pp. 115–125
2. M. Guri, B. Zadov, and Y. Elovici. “Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields,” IEEE Transactions on Information Forensics and Security, vol. 15–2019. — pp. 1190–1203
3. Security Researcher Finds Facebook App Tracking iPhone Movements [Электронный ресурс]: https://www.forbes.com/sites/zakdoffman/2021/10/23/apple-iphone-users-delete-facebook-app-after-new-tracking-warning/ (дата обращения: 09.07.2022).

Национальный исследовательский университет «МЭИ»