В результате объединения IDS Snort и МСЭ IPTables получается двухуровневая система защиты: на первом уровне iptables проверяет входящий пакет на соответствие своим правилам фильтрации, если пакет получил разрешение на прохождение через межсетевой экран, его проверяет система обнаружения вторжений на наличие вредоносного кода в теле входящего пакета. Представлен скрипт конфигурирующий Snort_inline и IPTables для совместной работы.

ОБ ОДНОЙ МОДИФИКАЦИИ АЛГОРИТМА XOR

И.В. Молчанов, М.А. Калугина

Обратимость логической операции исключающего ИЛИ позволила успешно использовать ее в синхронном шифровании. Однако основанный на ней классический алгоритм XOR рекомендуется применять лишь однократно из-за его уязвимости к криптоанализу при достаточно большом количестве перехваченных зашифрованных данных.

Реализованный в утилите для Windows алгоритм на основе простого XOR предназначен для приложений, использующих уникальный ключ для шифрования каждого конкретного блока информации. Он состоит из шифрования данных простым алгоритмом XOR и добавления «шума» с использованием специальной рекуррентной последовательности псевдослучайных чисел. Эта последовательность формируется заранее по следующей схеме: $a_0=A$, $a_1=B$, $a_{i+1}=\alpha \bullet a_{i-1}+ \beta \bullet a_i$, − где A, B, α, β – псевдослучайные числа, которые вычисляются до исполнения.

Ключ генерируется как последовательность псевдослучайных байтов или как ключевая фраза, задаваемая пользователем (данная реализация включает псевдослучайный ключ на основе вычисления количества байтов в шифруемом файле). Частью ключа являются значения A, B, α, β.

Шифрование. Пусть M , K , E – массивы символов (байтов) в файле ввода, в ключе и в файле вывода соответственно. Тогда $E_i = M_i$ xor K_i, если i не совпадает с каким-либо элементом последовательности, $E_i = T$ – псевдослучайный символ, если совпадает. Расшифровка. Пусть E, K, M – массивы символов (байтов) в файле ввода (зашифрованная информация), в ключе и в файле вывода соответственно. Тогда $M_i = E_i$ xor K_i, если i не совпадает с каким-либо элементом последовательности, E_i игнорируется, если совпадает.

Таким образом, попытка узнать длину ключа путем циклического сдвига (первый шаг по взлому XOR) не осуществима из-за разных дистанций между a_i и a_{i+1}. Кроме того, при разработке утилиты особое внимание было уделено не только реализации данного алгоритма, но и распространению такого подхода на асинхронный алгоритм шифрования.

ИСПОЛЬЗОВАНИЕ КОДОВ ХЭММИНГА ДЛЯ ОБНАРУЖЕНИЯ И ИСПРАВЛЕНИЯ ОШИБОК В DWDM СЕТЯХ

Б.А. Монич

В системах оптических магистралей, которые работают на терабитных и мультигигабитных скоростях, используется уплотненного волнового мультиплексирования (DWDM). Алгоритм обнаружения и исправления ошибок методом кодов Хэмминга на DWDM транспортной системы Optix OSN 8800 компании Huawei был впервые опробован на одном из 10G каналов. В Республике Беларусь данная транспортная система также является частью сети передачи данных. Функции выявления ошибок реализуются на транспортировке в составе блока избыточной служебной информации, по которой смотрят степень достоверности принятой информации.

Для обнаружения ошибок разработаны и применяются различные алгоритмы. Среди них основными являются следующие.

1. Контроль по паритету. Представляет собой наиболее простой метод контроля данных. Метод заключается в суммировании по модулю 2 всех бит контролируемой информации. Существуют горизонтальный и вертикальный контроль по паритету.

2. Циклический избыточный контроль. Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R.

Библиотека БГУИР

3. Вертикальный и горизонтальный контроль по паритету. Исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы.

4. Коды Хэмминга. Самостоятельно находят и исправляют ошибки, даже изолированные, то есть искаженные отдельные биты которые могут быть разделены большим количеством правильных битов.

Построение кодов Хэмминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным, ($S = 0$ – ошибок нет, $S = 1$ – однократная ошибка). Матрица преобразования соответствующей размерности умножается на матрицу-столбец кодового слова и каждый элемент полученной матрицы-столбца берется по модулю 2. На аппаратном уровне, в Optix OSN 8800, коды Хэмминга реализуются в модуле OTN tributary unit, на плате TN52TOG.

ИСПОЛЬЗОВАНИЕ ЖУРНАЛА ИЗМЕНЕНИЙ ФАЙЛОВ NTFS В КОМПЬЮТЕРНО-ТЕХНИЧЕСКОЙ ЭКСПЕРТИЗЕ

О.Р. Мысливец

При проведении компьютерно-технической экспертизы, а именно при анализе файловой системы NTFS, часто встает необходимость в поиске удаленных файлов и их анализе. Несмотря на то, что методы поиска удаленных файлов и их восстановления для файловой системы NTFS [1] не представляют особой сложности, не всегда возможно полностью восстановить файл для последующего анализа. В подобных случаях, если не удается восстановить файл полностью либо частично, существует возможность получить информацию о том, хранился ли файл в файловой системе в определенный промежуток времени и определить действия, проводимые над файлом. Информацию такого рода можно получить из журнала изменений файлов NTFS. Наряду с именами файлов и информацией об их MFT-записях, данный метафайл хранит в себе информацию о действиях, произведенных над файлами, а также временные метки произведенных действий [2]. С помощью данной информации можно однозначно установить факт хранения и использования некоторого файла, вплоть до получения полной последовательности произведенных над файлом действий за определенный промежуток времени. Недостатками данного подхода можно считать возможное отсутствие метафайла, вследствие его преднамеренного удаления или отключение опции журналирования для конкретного тома, а также ввиду ограничений на размер журнала изменений файлов, некоторые записи в журнале могут быть перезаписаны более новыми, что может повлиять на конечный результат. Несмотря на все недостатки данного подхода, в ходе исследований выяснилось, что в процессе анализа журнала изменений файлов NTFS существует вероятность получения информации о файлах, удаленных более 3 месяцев назад, но и не более 6 месяцев с момента удаления при условии активного использования накопителя информации.

Литература

1. Кэрриэ, Б. Криминалистический анализ файловых систем/ Кэрриэ Б. – СПб.: Питер, 2007. – 470 с.

2. USN_RECORD_V2 structure [Электронный ресурс] / Microsoft TechNet – Режим доступа: https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa365722(v=vs.85).aspx – Дата доступа: 17.05.2017.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОГО ХРАНЕНИЯ И ЗАЩИТЫ ИНФОРМАЦИИ БАЗЫ ДАННЫХ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Е.А. Нестерович, С.А. Зайкова

В современных условиях любая деятельность сопряжена с оперированием большими объемами информации, которое производится широким кругом лиц. Защита данных от несанкционированного доступа является одной из приоритетных задач при проектировании любой информационной системы. Следствием возросшего в настоящее время значения информации стали высокие требования к конфиденциальности данных. Системы управления базами данных (СУБД), в особенности реляционные, стали доминирующим инструментом в этой области. Обеспечение информационной безопасности СУБД приобретает решающее

Библиотека БГУИР