СИГНАТУРНОЕ СКАНИРОВАНИЕ И ЭНТРОПИЙНЫЙ АНАЛИЗ ИСПОЛНЯЕМЫХ ФАЙЛОВ ПРОГРАММ

Блинов В.В.

Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
Минск, Республика Беларусь

Савенко А.Г. – м.т.н., старший
преподаватель кафедры ИСиТ

Аннотация. В работе представлено разработанное программное средство сигнатурного сканирования и энтропийного анализа исполняемых файлов программ. Энтропийный анализ выполняется по формуле Клода Шеннона. Разработанное программное средство позволяет наглядно визуализировать результаты своей работы, а также содержит встроенный редактор баз правил, позволяющий как использовать готовые существующие правила, так и создавать и редактировать собственные правила выполнения энтропийного анализа.

Ключевые слова: сигнатурное сканирование, энтропийный анализ, исполняемый файл, вредоносные программы, кибербезопасность.

Необходимость идентификации объектов возникает при решении многих прикладных задач, в частности, задачи выявления вредоносного программного обеспечения (ПО). Получить нужный результат можно сравнивая структуру файла с известными подозрительными программами. В ряде задач, возникающих при использовании современных информационных технологий, требуется сравнение файлов или их частей. К таким задачам относятся, например, классификация, обнаружение незаконного использования данных, поиск дублирующихся участков программного кода. Также получение информации о компиляторе

(которым было скомпилировано исследуемое программное обеспечение), упаковщике, системе лицензирования (протекторе). К одной из важных практических задач, для решения которой может проводиться сравнение файлов, относится автоматическое обнаружение вредоносного ПО и получение дополнительной информации о ПО для дальнейшего анализа. Современной компьютерной вирусологии известно множество вредоносных программ, каждая из которых представляет собой незначительную модификацию одного и того же исполняемого кода. Это значительно усложняет работу антивирусных программ, основанных на сигнатурном анализе: фактически необходимо проанализировать каждую из модификаций.

Классификации вредоносных программ на семейства, каждое из которых содержит похожие файлы, представляющие собой модификации одного и того же кода, значительно облегчают борьбу с вирусами, так как позволяет для установления факта принадлежности программы какому-либо семейству проводить анализ, используя только один или небольшое число представителей этого семейства. Такая идентификация может быть выполнена на основе автоматического сравнения файлов и обнаружения среди них похожих [1].

Для решения задачи сигнатурного сканирования и энтропийного анализа исполняемых файлов программ было спроектировано и разработано программное средство «dSign».

К основным функционалом разработанного программного является:

– сигнатурное сканирование файла;
– расчет энтропии файла;
– построение гистограммы эмпирической плотности распределения;
– просмотр совпадающих правил;
– редактирование содержимого базы правил;
– просмотр содержимого базы правил;
– просмотр лога (окна вывода);
– проверка программы и базы правил на наличие обновлений.

Сигнатура представляет собой уникальную для каждого ПО последовательность байт, которая однозначно идентифицирует определенную программу. Сигнатура сама по себе не несёт никакого смысла и может вызвать недоумение, встретившись в коде программы без соответствующего контекста или комментария, при этом попытка изменить его на другое, даже близкое по значению, может привести к абсолютно непредсказуемым последствиям. В UNIX-подобных операционных системах тип файла обычно определяется по сигнатуре файла, вне зависимости от расширения его названия.

Разработанное программное средство «dSign» имеет возможность расчёта энтропии файла по формуле Клода Шеннона и вывод данных, которые использовались при расчете в таблицу. Любой компьютерный файл, как известно, состоит из байтов. Байт может принимать значения от 0 до 255. Информационная энтропия – это статистический параметр, который показывает вероятность встречаемости определённых байтов в файле [2].

Формула Клода Шеннона для расчета энтропии исполняемого файла выглядит следующим образом:

$$\mathrm{H}(x) = -\sum_{i=1}^{n} p_i \times \log_2 p_i$$

где p_i – вероятность наступления i-го исхода.

Энтропия бывает:
термодинамическая;
алгоритмическая;
информационная;
дифференциальная;
топологическая [3].

Построение гистограммы эмпирической плотности распределения является одним из этапов энтропийного анализа. Гистограмма эмпирической плотности распределения строиться следующим образом. По оси X (оси абсцисс) будут показаны значения байта (0 - 255), по оси Y (оси ординат) будет показана его частота по отношению к загруженному исполняемому файлу. Примеры гистограммы эмпирической плотности распределения представлены на рисунках 1 и 2.

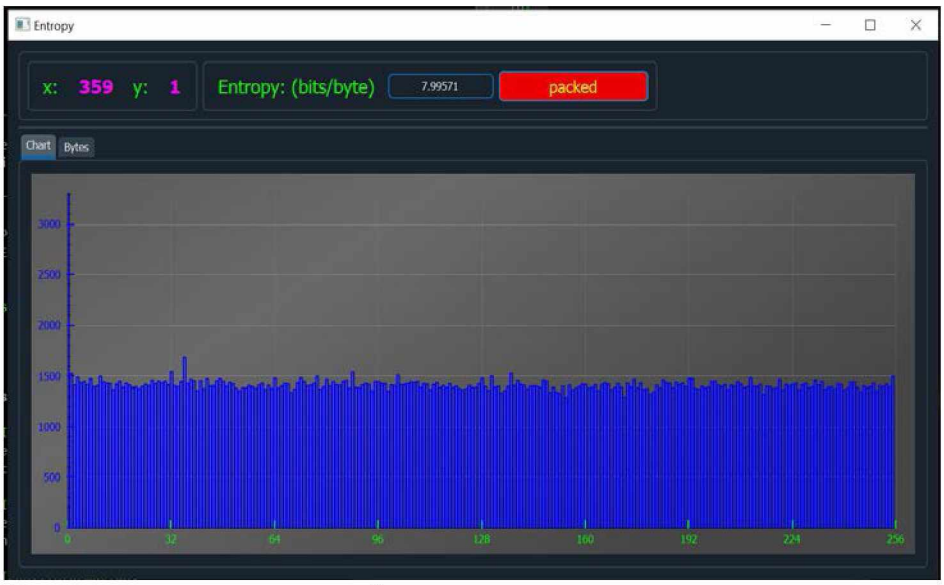

Рисунок 1 – Гистограмма эмпирической плотности распределения для упакованного файла

Рисунок 2 – Гистограмма эмпирической плотности распределения для не упакованного файла

Пример отображения рассчитанной энтропии и данных, используемых при её расчёте представлен на рисунке 3.

Редактор базы правил для энтропийного анализа поддерживает синтаксис популярного инструмента с открытым кодам YARA. Каждое новое правило в YARA начинается в ключевого слова «rule» и затем следует идентификатор правила. Идентификаторы правил должны соответствовать тем же лексическим конвенциям, как в языке программирования C, они могут содержать любой символ английского алфавита и символ подчеркивания «_», но первый символ не может быть цифрой. Идентификаторы правил устойчивы к регистру и не могут превышать длину в 128 символов.

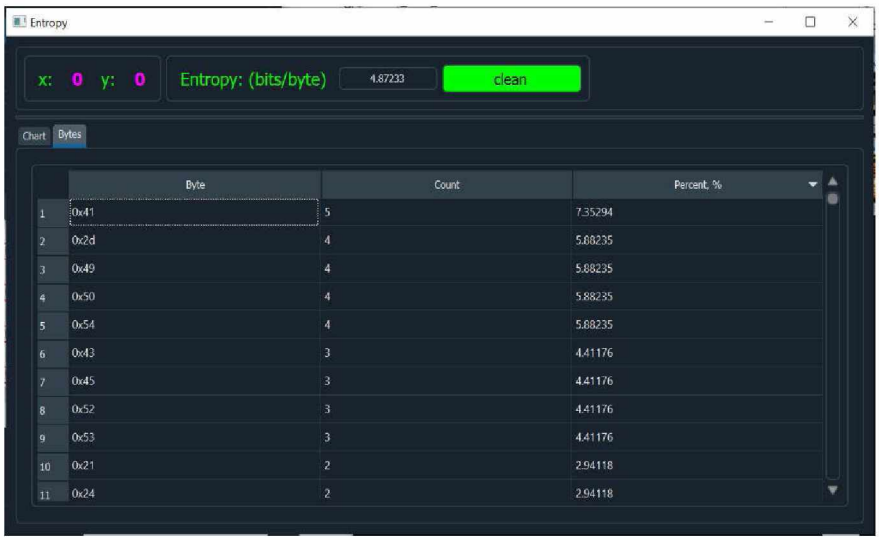

Рисунок 3 – Просмотр рассчитанной энтропии и данных использующихся при её расчете

Правила обычно состоят из двух секций: определение строк (сигнатур) «strings» и условия «condition». Секция с определением строк может отсутствовать, если правило не зависит от какой-либо строки, но секция условий требуется всегда. Секция определения строк – это место, где определяются строки(сигнатуры), от которых зависит правило и являются его частью. Каждая строка имеет идентификатор включающий в себя символ «$» за которым следует последовательность из букв английского алфавита и символа подчеркивания «_», данные идентификаторы могут быть использованы в секции условия «condition» для ссылки на соответствующую строку. Строки могут быть определены в текстовой или шестнадцатеричной форме.

Текстовые строки берутся в двойные кавычки, как в языке программирования C. Шестнадцатеричные строки берутся в фигурные скобки, они состоят из последовательности шестнадцатеричных чисел, которые могу появляться последовательно или разделяться пробелами. Десятичные цифры не допускаются в шестнадцатеричных строках.

Секция условий «condition», которая содержит логику правила. Секция условий «condition» должна хранить логическое выражение, указывающее, при каких обстоятельствах файл или процесс удовлетворяет правилу. Обычно условие ссылается на идентификаторы строк, ранее определенных в секции «strings», в таком случаи строковый идентификатор действует как логическая переменная, которая принимает значение true, если строка была найдена в памяти файла или процесса и false, если это не так [4].

В качестве тестового примера использовался сканнер и загруженный в него файл (в данном случаи EICAR файл). Правило, определяющее EICAR файл, представлено на рисунке 4:

```
rule eicar_av_test {
    strings:
        $eicar_regex = /^X5O!P%@AP\[4\\PZX54\(P\^\)7CC\)7\}\$EICAR-
STANDARD-ANTIVIRUS-TEST-FILE!\$H\+H\*\s*$/
    condition:
        all of them
}

rule eicar_substring_test {
    meta:
        description = "Standard AV test, checking for an EICAR substring"

    strings:
        $eicar_substring = "$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!"

    condition:
        all of them
}
```

Рисунок 4 – Правило для определения EICAR файла

Пример редактирования правила в синтаксисе YARA представлен на рисунке 5.

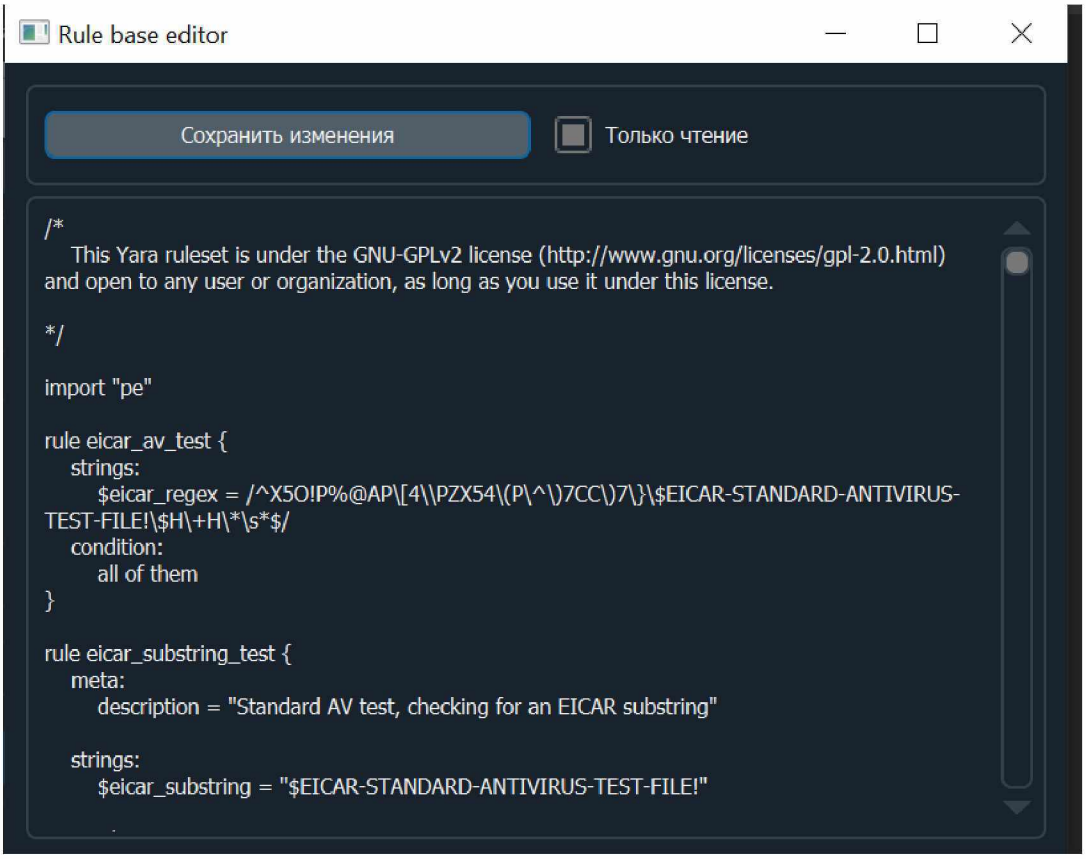

Рисунок 5 – Просмотр содержимого базы правил и возможность редактирования

Сигнатурное и энтропийное сканирование исполняемого файла выполняется параллельно в разных потоках.

При разработки программного средства «dSign» для анализа исполняемых файлов использовался язык программирования C++ с использованием Qt framework. Для работы с правилами dSign использует возможности YARA framework. Построение гистограммы осуществляется с помощью вспомогательного виджета QCustomPlot.

Основное преимущество разработанного программного средства заключается в том, что программа по сравнению с аналогами проста в использовании, имеет возможность редактирования базы правил прямо из программы, имеет современный и удобный интерфейс, визуализирует построение гистограммы эмпирической плотности распределения.

Список использованных источников:

1. Идентификация типа файла на основе структурного анализа [Электронный ресурс]. – Режим доступа : https://cyberleninka.ru/article/v/ identifikatsiya-tipa-fayla-na-osnove-strukturnogo-analiza
2. Что такое энтропия файла. [Электронный ресурс]. – Режим доступа : https://soltau.ru/index.php/themes/kompyutery-i-programmy/item/467-chto-takoe-entropiya-fajla.
3. Введение в понятие энтропии и ее многоликость. [Электронный ресурс]. – Режим доступа : https://habr.com/ru/post/305794/.
4. Writing YARA rules. [Electronic resource]. – Mode of access : https://yara.readthedocs.io/en/stable/writingrules.html.