DOI: 10.25586/RNU.V9187.21.03.P.156 УДК 342

С.А. Нестерович, Ю.И. Купцова

О НЕКОТОРЫХ ВОЗМОЖНОСТЯХ ОБНАРУЖЕНИЯ СКРЫТОГО ВРЕДОНОСНОГО КОДА

Рассмотрен способ обнаружения скрытого вредоносного кода с помощью анализа энтропии. Если лицо, совершающее злой умысел, внедряет вредоносный код в оригинальный файл, кодирует, сжимает его, то это действие увеличит энтропию. Скомпилированный файл любой программы содержит некоторые участки кода, которые в большинстве своем распределяются равномерно. В том случае, когда используется запутывание или кодирование кода, данная равномерность имеет свойство нарушаться. Анализ энтропии – это базовая оценка объекта тестирования, позволяющая сделать вывод о том, какую секцию или часть файла нужно анализировать, чтобы понять, является ли угрозой объект в целом.

Ключевые слова: анализ, энтропия, вредоносное программное обеспечение, обфускация, видоизмененный файл.

S. Nesterovich, Y. Kuptsova

ABOUT SOME POSSIBILITIES OF DETECTING HIDDEN MALTIC CODE

The article discusses how to detect hidden malicious code using entropy analysis. If a person who commits evil, intent, introduces malicious code into the original file, encodes, compresses it, etc., then this action will increase entropy. The compiled file of any program contains some sections of code, which are mostly evenly distributed. When code entanglement or coding is used, this uniformity has the property of being broken. Entropy analysis is a basic assessment of a test object, which allows you to conclude which section or part of a file you need to analyze to understand whether the object as a whole is a threat.

Keywords: analysis, entropy, malicious software, obfuscation, modified file.

Введение

При решении многих задач по информационной безопасности необходимо анализировать различные устройства на предмет наличия видоизмененных файлов. Такие файлы могут быть изменены специально вручную или вредоносным программным обеспечением и впоследствии могут представлять определенную угрозу.

Одной из таких задач является оценка этих файлов. Для этого могут применяться различные способы: известные криптографические хеши, а также классические контрольные суммы, которые облегчают задачу нахождения одинаковых файлов. Можно один раз посчитать хеш-функцию от файла, чтобы в дальнейшем быстро обнаруживать идентичные ему. При равенстве хешей с некоторой степенью уверенности можно судить об идентичности файлов в зависимости от значения уровня коллизий [1] конкретной хеш-функции.

Малейшее изменение файла приводит к тому, что хеш-функция меняется. Это важно, особенно когда решаются задачи по поиску программного обеспечения (ПО), которое может нанести вред устройству. Если использовать обфускацию (запутывание) кода программы, это также приведет к изменению хеш-функции [2]. Вышеописанное происходит из-за свойств функции хеша, а именно из-за «лавинного эффекта» [1]. Этот эффект характеризуется тем, что изменения выходных данных взаимосвязаны с изменениями данных хеш-функции. Следовательно, если цель злоумышленника – избежать обнаружения

Сергей Александрович Нестерович
кандидат технических наук, доцент кафедры информационных технологий Московской академии Следственного комитета Российской Федерации, Москва. Сфера научных интересов: информационные технологии, системы, основанные на знаниях, информационная безопасность. Автор 23 опубликованных научных работ.
E-mail: sirial_2005@mail.ru

Купцова Юлия Ильинична
кафедра информационных технологий Московской академии Следственного комитета, Москва. Сфера научных интересов: информационные технологии, вычислительные распределенные системы, информационная безопасность.
E-mail: Kuptsova.yulia3@yandex.ru

при помощи криптографических хешей, то ему достаточно произвести незначительные изменения вредоносного файла.

Информационная энтропия

Также имеется проблема, которая связана с проведением идентификации похожих файлов. Например, вредоносная программа была немного изменена, чтобы избавиться от сигнатуры и не считаться вредоносной. Однако при таком изменении больше половины файла изменения не затронут. Этот недостаток решается методом, который основан на частотном анализе энтропии файлов.

Информационная энтропия – это понятие из прикладной теории информации, которое обозначает меру неопределенности или непредсказуемости информационной системы [4]. Стоит отметить, что данное понятие имеет более широкий смысл и дает возможность для проведения анализа некоторых данных. Слово «анализ» здесь используется в контексте предположений, основанных на том, что энтропия взаимосвязана с информацией.

Понятие «информационная энтропия» было основано Шенноном. Она являлась некоторым разнообразием, которые представляли собой данные. Ряд данных связан с энтропией. Чем больше он гладкий, тем меньше сама энтропия. В случае, когда ряд является идентичными значениями, энтропия будет нулевой [3].

Увеличение различных значений влечет за собой возрастание энтропии этого признака, а также содержание информации, которая в нем имеется. В связи с этим энтропия – одна из мер насыщенности информацией данных. Информационная насыщенность данных будет высокой при высокой энтропии набора, и наоборот.

Чтобы вычислить энтропию, используется метод под названием «скользящее окно». Суть данного метода состоит в том, что осуществляется счет конфигураций, которые являются уникальными.

Имеются измерения признака проявления для каждого возможного значения байта (от 0 до 255). Данные значения располагаются на определенном интервале. Чтобы оценить значение параметра статистики, выбираются некоторые точки, которые имеют конкретное значение. Полученный результат будет присвоен точке на профиле, которая полностью совпадает с положением, имеющимся у центральной точки окна. В том случае, если «окно» будет перемещаться по профилю со значением, равным расстоянию между

рядом стоящими наблюдениями на профиле, то получится значение, которое равно оценки среднего в каждой точке этого профиля.

Далее данные частоты (fi) рассчитываются по формуле

$$H(x) = -\sum_{i=0}^{255} f_i \log_2 (f_i).$$

Анализ энтропии состоит в следующем. Скомпилированный файл любой программы содержит некоторые участки кода, которые в большинстве своем распределяются равномерно. В том случае, когда используется запутывание или кодирование кода, данная равномерность имеет свойство нарушаться. В файле образуются высокоэнтропийные области, а также области, которые меньше подвергаются запутыванию или кодированию.

Характерное качество алгоритма сжатия – распределение частот, которые встречаются в байте кода. Данный фактор станет заметным при проведении анализа. Эти файлы будут отличаться высокой степенью энтропии, которая близка к значению максимума. Иными словами, маленькая избыточность в файле обусловлена высокой энтропией.

Если энтропия более семи для файлов, которые сжаты, закодированы или зашифрованы при практическом применении, то практически со стопроцентной гарантией можно заявить о том, что применялось преобразование кода. Файлы, которые не подвергались преобразованию, имеют энтропию от 2 до 6.

Представим, что существует некоторый объект. В данный объект лицо, совершающее злой умысел, внедряет вредоносный код, при этом помещает нагрузку, которая является полезной, в оригинальный файл, кодирует, сжимает некоторый фрагмент данных. Это действие увеличит энтропию.

Вирусный аналитик проверит образец на наличие энтропии и поймет, запакован этот образец или обфусцирован, а затем на основе полученных данных выберет метод анализа данного объекта.

Примеры инструментов для подсчета энтропии

Имеются разные инструменты, которые совершают подсчет энтропии. Наиболее популярный из них Detect It Easy (DIE).

Будем использовать именно данную программу, а также разбирать примеры популярного образца Agent Tesla.

Detect It Easy – дает возможность понять, какой у файла тип компилятора, проекта и установщика.

Agent Tesla является модульным ПО, которое предназначено для шпионажа. Данное программное обеспечение охватывает модели malwareas-a-service под видом кейлоггера.

Этот «шпион» дает возможность извлечь, а затем передать на сервера данные, которые хранятся в браузерах и буферах обмена, осуществлять скриншоты экранов, которые в итоге получают злоумышленники.

Ниже представлены скриншоты, на которых виден интерфейс. На них имеется следующая информация:

1) Тип файла;
2) Оценка энтропии;
3) Статус файла (запакованный файл или нет);
4) Статус искомого файла и подсчёт энтропии;
5) График энтропии.

Также представлены графики энтропии для образца Agent Tesla и его разновидности (рис. 1, 2).

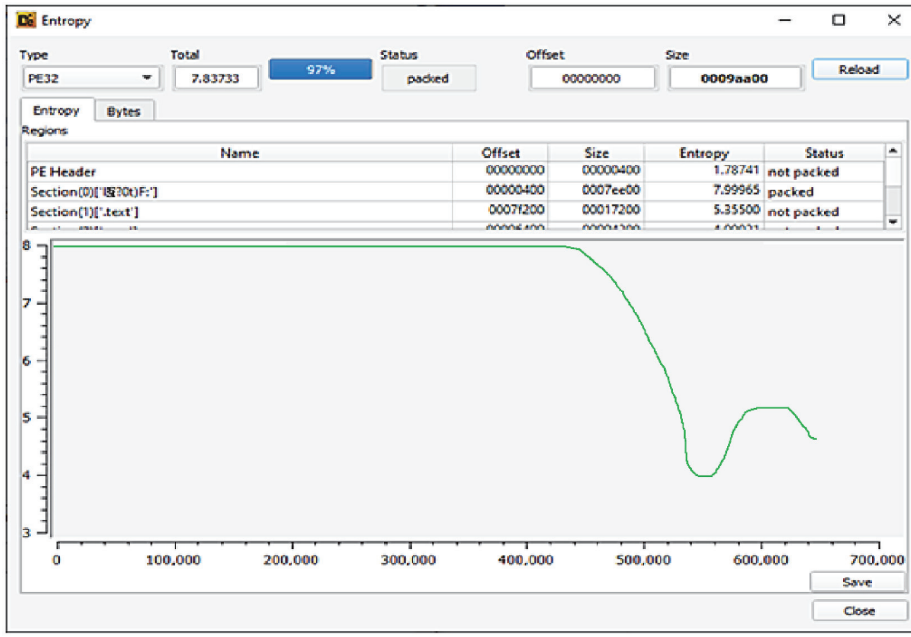

Рис. 1. Результат анализа образца *1* в программе DIE

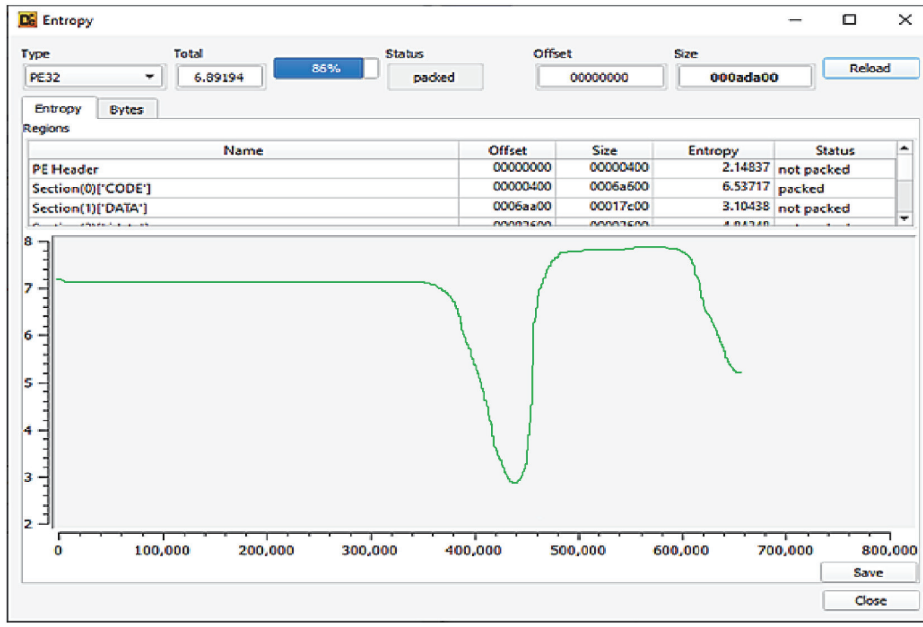

Рис. 2. Результат анализа образца *2* в программе DIE

Обычным примером является то, когда PE-заголовок, но оставшаяся часть файла имеет довольно высокий уровень энтропии. К тому же секцию кода невозможно анализировать, так как она очень походит на случайные значения. Данный фактор говорит о том, что было произведено сжатие файла или его кодировка. Это является стандартным приемом, который используется, чтобы антивирусы не обнаружили вредоносный файл.

Особенностью данного подхода является оценка высоких и низких энтропийных областей, а также селекция для всех областей файла. Оценка энтропии и сравнение полученных данных дает возможность хешировать объекты, а затем объединить их в единый хеш, воспользовавшись фильтром Блума.

Фильтр Блума заводит массив битов, размер которых фиксирован под m-значением, а также набор, который состоит из разных функций хеша для k-значений, выдающие значения от нуля до $m - 1$. В том случае, если нужно добавить один элемент к множеству других, для одного элемента происходит расчет значения функции хеша, а затем устанавливаются биты в массиве, которые соответствуют индексам.

Чтобы проверить принадлежность, нужно сосчитать значения функций хеша для потенциального участника, а также удостовериться в том, что все биты находятся в значении единицы, тогда при соблюдении данных условий ответ будет «возможно». Если один из битов не равен единице, то множество элементов будет с ответом «не содержит» или «нет».

На текущий момент аналитика энтропии используется в различных сервисах и предложениях, которые относятся к информационной безопасности. например, анализ данного типа применим в машинном обучении, когда создаются модели, оценивающие файл в антивирусном ПО. Оценка энтропии будет использована при расчете весов, когда идет этап оценивания вредоносного объекта в анти-APT-средствах защиты или в процессе анализа объектов в динамике.

Заключение

Применение на практике энтропии рассмотрено на примере модуля анализа поведения, в составе которого заложен комплекс Group-IB Threat Detection System. Данная система является системой предупреждения от угроз TDS (Threat Detection System), которая предназначена для того, чтобы выявить атаки на первоначальной стадии. Threat Detection System создана и произведена отечественной компанией Group-IB, которая известна в России и на мировом рынках. TBS – это часть системы предупреждения угроз на первоначальной стадии, которая создана с платформой Group-IB Threat Intelligence.

Таким образом, анализ энтропии можно назвать базовой оценкой тестируемого объекта, позволяющей определить, на какую секцию или часть файла следует обратить более пристальное внимание при анализе образца и понять, подозрителен ли образец в целом.

Литература

1. *Alfred J., Menezes Paul C., Van Oorschot and Scott A.* (2001) Vanstone Handbook of Applied Cryptography. Boca Raton, Florida, США: CRC Press, 810 p.

2. *Лифшиц Ю.М.* Запутывание (обфускация) программ. Обзор. СПб.: Санкт-Петербургское отделение математического института им. В. А. Стеклова РАН, 2004. – URL: http://logic.pdmi.ras.ru/~yura/of/survey1.pdf

3. *Паклин Н.Б., Орешков В.И.* Бизнес аналитика: от данных к знаниям (+CO): учеб. пособие. Изд. 2-е, испр. СПб.: Питер, 2013. 704 с.

4. *Чикрин Д.Е.* Теория информации и кодирования: курс лекций. Казань: Казанский университет, 2013. 116 с.
5. *Чумак О.* В. Энтропии и фракталы в анализе данных. М. – Ижевск: Регулярная и хаотическая динамика, Институт компьютерных исследований, 2011. 164 с.

References

1. Alfred J., Menezes Paul C., Van Oorschot and Scott A. (2001) Vanstone Handbook of Applied Cryptography. Boca Raton, Florida, США: CRC Press, 810 p.
2. Lifshits Yu.M. (2004) *Zaputyvanie (obfuskatsiya) programm. Obzor* [Obfuscation of programs. Overview]. St. Petersburg, Sankt-Peterburgskoe otdelenie matematicheskogo instituta im. V. A. Steklova RAN. Available at: http://logic.pdmi.ras.ru/~yura/of/survey1.pdf (in Russian).
3. Paklin N.B., Oreshkov V.I. (2013) *Biznes analitika: ot dannykh k znaniyam (+SO)* [Business analytics: from data to knowledge (+ SB)]. St. Petersburg, Piter Publishing, 704 p. (in Russian).
4. Chikrin D.E. (2013) Teoriya informatsii i kodirovaniya [Information and coding theory]. Kazan, Kazanskiy universitet Publishing, 116 p. (in Russian).
5. Chumak O.V. (2011) *Entropii i fraktaly v analize dannykh* [Entropies and fractals in data analysis]. Moscow – Izhevsk, Regulyarnaya i khaoticheskaya dinamika, Institut komp'yuternykh issledovani Publishing, 164 p. (in Russian).