УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2020 Том 20 № 5 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2020 Vol. 20 No 5 ISSN 2226-1494 http://ntv.ifmo.ru/en/
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

УДК 004.054 doi: 10.17586/2226-1494-2020-20-5-708-713

МЕТОД ОПРЕДЕЛЕНИЯ УПАКОВАННЫХ И ЗАШИФРОВАННЫХ ДАННЫХ ВО ВСТРОЕННОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

А.Н. Югансон

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: a_yougunson@itmo.ru

Информация о статье
Поступила в редакцию 23.07.20, принята к печати 30.08.20
Язык статьи — русский

Ссылка для цитирования: Югансон А.Н. Метод определения упакованных и зашифрованных данных во встроенном программном обеспечении // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 5. С. 708–713. doi: 10.17586/2226-1494-2020-20-5-708-713

Аннотация

Предмет исследования. Исследование встроенного программного обеспечения на предмет наличия дефектов безопасности может быть затруднено применением различных приемов антиотладки (шифрование) и упаковщиков кода (сжатия). Представлен обзор существующих инструментов для определения методов антиотладки. Недостатки существующих решений заключаются в применении сигнатурных методов анализа исполняемых файлов, что ограничивает область их применения количеством известных сигнатур. Существующие статистические методы, основанные на энтропийном анализе файлов, дают неоднозначный результат. Для определения способа преобразования данных предложен метод обнаружения упакованных и зашифрованных данных в исполняемом файле встроенного программного обеспечения. **Метод.** Экземпляр встроенного программного обеспечения представляется в виде конечной последовательности байтов, где каждый байт может принимать одно из 256 возможных значений. Предложенный метод совмещает подходы, основанные на использовании критерия согласия Пирсона для проверки гипотезы о равномерном распределении байтов в файле, а также применение метода Монте-Карло для аппроксимации числа π с целью вычисления характеристик распределения байтов в файле. Чем выше точность аппроксимации числа π и чем ближе распределение байтов в файле к равномерному, тем вероятнее применение алгоритмов шифрования для преобразования данных. **Основные результаты.** Показано, что предложенные критерии более чувствительны к отклонениям равномерно распределенной случайной величины, чем энтропийный анализ. Применение этих подходов к экспериментальной выборке файлов с различным размером, которые были подвергнуты различным алгоритмам сжатия/шифрования, показало корреляции, благодаря которым, с высокой степенью уверенности, можно утверждать, какому алгоритму (сжатия или шифрования) было подвергнуто встроенное программного обеспечение. **Практическая значимость.** Представлен подход для определения упакованных и зашифрованных данных, полученных в результате использования различных приемов антиотладки. Предложенный метод применим как в рамках анализа вредоносного программного обеспечения, так и в рамках поиска и идентификации дефектов безопасности во встроенном программном обеспечении.

Ключевые слова
встроенное программное обеспечение, статистические тесты, энтропийный анализ, критерий согласия Пирсона, метод Монте-Карло, приемы антиотладки, информационная безопасность

doi: 10.17586/2226-1494-2020-20-5-708-713

DETERMINATION OF PACKED AND ENCRYPTED DATA IN EMBEDDED SOFTWARE

A.N. Iuganson

ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: a_yougunson@itmo.ru

Article info
Received 23.07.20, accepted 30.08.20
Article in Russian

For citation: Iuganson A.N. Determination of packed and encrypted data in embedded software. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 5, pp. 708–713 (in Russian). doi: 10.17586/2226-1494-2020-20-5-708-713

Abstract

Subject of Research. Embedded software research for security faults can be handicapped by various anti-debugging techniques (encryption) and code wrappers (compression). The paper presents an overview of existing tools for definition of anti-debugging techniques. The disadvantages of existing solutions lie in the use of signature-based methods for analysis of executable files, that limits the scope of their application to the number of the known signatures. The existing statistical tests based on the entropy analysis of files give an ambiguous result. To determine the data conversion technique, a method is proposed for detection of packed and encrypted data in an executable firmware file. **Method.** The embedded software is represented as a finite sequence of bytes, where each byte can take one of 256 possible values. The proposed method combines the approaches based on the use of Pearson's chi-squared test to check the hypothesis of a uniform distribution of bytes in a file, as well as the use of the Monte Carlo method to approximate the number π in order to calculate the characteristics of the distribution of bytes in a file. The higher approximation accuracy of the number π and the closer the distribution of bytes in the file to a uniform one is, the more likely is the application of encryption algorithms for data transformation. **Main Results.** It is shown that the proposed criteria are more sensitive to deviations of a uniformly distributed random variable than the entropy analysis. Applying of these approaches to an experimental sample of files with various sizes, which were compressed/encrypted with a variety of algorithms, have shown correlations, that with a high degree of confidence give the possibility to state which algorithm (compression or encryption) the embedded software was subjected to. **Practical Relevance.** An approach is presented for determination of packed and encrypted data obtained as a result of the use of various anti-debugging techniques. The proposed method is applicable both in the analysis of malicious software and in the search and identification of security defects in embedded software.

Keywords

embedded software, statistical tests, entropy analysis, Pearson's chi-squared test, Monte Carlo method, anti-debugging techniques, information security

Введение

Под встроенными системами понимается множество устройств, предназначенных для взаимодействия с физическим миром через ряд периферийных устройств. Основными компонентами встроенной системы являются: аппаратное обеспечение; коммуникационные интерфейсы; загрузчик; ядро операционной системы; файловая система; приложения уровня пользователя.

В настоящей работе под встроенным программным обеспечением (ПО) понимается совокупность программных компонентов, которые обеспечивают управление встроенной системой на уровне аппаратной части.

Встроенное ПО представляет огромный интерес для злоумышленников: отчетливо виден рост числа успешных атак на встроенные системы (например, Stuxnet [1, 2], Mirai [3, 4]). Данный рост обусловлен в первую очередь повсеместным распространением встроенных систем (начиная от носимой электроники и заканчивая беспилотными транспортными средствами), автоматизацией процесса производства (переходом индустриальных компаний к киберфизическим системам), а также пренебрежением принципами SDL (Secure Development Lifecycle — жизненный цикл безопасной разработки) со стороны разработчиков встроенного ПО в угоду скорейшему выводу продукта на рынок.

Многие исследователи заняты поиском и анализом уязвимостей встроенного ПО [5, 6]. Для оценки защищенности встроенного ПО необходимо изучить состав программных компонентов и исследовать их на предмет наличия дефектов безопасности. Однако исследование зачастую затруднено применением различных приемов антиотладки (шифрование) и упаковщиков кода (сжатия) со стороны производителей. Как показывает практика [7, 8], используемый принцип обеспечения безопасности встроенного ПО через неясность (security through obscurity), является малоэффективным.

Таким образом, прежде чем приступить к анализу защищенности встроенного ПО, необходимо определить, каким алгоритмам сжатия или шифрования были подвергнуты компоненты встроенного ПО.

Обзор применяемых подходов определения упакованных и зашифрованных данных

Рассмотрены инструменты, в которых решаются задачи, схожие с задачей определения упакованных и зашифрованных данных во встроенном ПО. К таким инструментам относятся:

— binwalk[1] — инструмент, разработанный для помощи в анализе и извлечении образов встроенного ПО и других бинарных объектов;
— signsrch[2] — инструмент, позволяющий определить используемые алгоритмы сжатия или шифрования популярных протоколов и форматов исполняемых файлов;
— offzip[3] — инструмент, предназначенный для распаковки данных формата zip (zlib, gzip и др.).

Одним существенным недостатком представленных инструментов является то, что они основаны на сигнатурном анализе файлов, реализованном в виде анализа заголовков исполняемых файлов, что ограничивает область их применения количеством известных сигнатур. Таким образом, внутренняя структура содержимого файлов не учитывается.

[1] Исходный код программы binwalk [Электронный ресурс]. Режим доступа: https://github.com/ReFirmLabs/binwalk, свободный. Яз. англ. (дата обращения: 02.05.2020).

[2] Официальный сайт разработчика программы signsrch [Электронный ресурс]. Режим доступа: http://aluigi.altervista.org/mytoolz/signsrch.zip, свободный. Яз. англ. (дата обращения: 02.05.2020).

[3] Официальный сайт разработчика программы offzip [Электронный ресурс]. Режим доступа: http://aluigi.altervista.org/mytoolz/offzip.zip, свободный. Яз. англ. (дата обращения: 02.05.2020).

С другой стороны, для выявления приемов антиотладки можно воспользоваться статистическими тестами NIST (The National Institute of Standards and Technology) для определения упакованных, зашифрованных и случайных данных [9–12]. Установлено, что решение данной задачи, исходя из значения энтропии, дает неоднозначный результат, так как диапазон значений энтропии для различных форматов данных (.pdf, .mp3, .avi) пересекается с диапазоном значений энтропии для зашифрованных и запакованных данных [13–15]. Для решения данной проблемы предлагается использовать дополнительные критерии для определения характера распределения байтов в файле: критерий согласия Пирсона и аппроксимация числа π методом Монте-Карло.

Описание предлагаемого метода

Представим исполняемый файл встроенного ПО S в виде конечной последовательности байтов $\{x_1, x_2, x_3, \ldots, x_N\}$, длины N, где $x_i \in X = \{0, 1, 2, 3, \ldots, 255\}$, $|X| = 256$ — мощность множества X, а N — размер встроенного ПО в байтах. Необходимо проверить, удовлетворяет ли распределение байтов в двоичном файле встроенного ПО свойствам равномерно распределенной случайной величины.

Для определения неоднородности данных использована мера усредненной информативности испытания, учитывающая вероятность отдельных исходов:

$$H = -\sum_{i=0}^{255} P(i)\log_2 P(i),$$

где $P(i) = N_i/N$ – вероятность появления байта со значением i в файле размера N.

Поскольку каждый байт в файле может иметь одно из 256 возможных значений, файл случайных данных будет иметь равномерное распределение значений байта от 0 до 255 включительно.

Для проверки гипотезы о равномерном законе распределения, необходимо построить интервальный вариационный ряд. Величина интервала определяется:

$$h = R/m,$$

где R — размах варьирования случайной величины (для исполняемых файлов $R = 255$); m — число интервалов, определяемое по формуле Стерджесса:

$$m = 1 + 3{,}32 \log_{10} N.$$

Оценка математического ожидания в этом случае:

$$\bar{x} = \sum_{j=1}^{m} \acute{x}_j f_j^*,$$

где $\acute{x}_j$ — середина интервала; f_j^* — частота попадания результатов наблюдения x_i в заданный интервал; j — номер интервала.

Оценка среднего квадратического отклонения:

$$S = \sqrt{\sum_{j=1}^{m} (\acute{x}_j - \bar{x})^2 f_j^*}.$$

Для проверки близости теоретического и эмпирического распределений частот появления байта в двоичном файле предлагается использовать критерий согласия Пирсона, вычисляемый по формуле:

$$\chi^2 = \sum_{j=1}^{m} \frac{(f_j^* - f_j)^2}{f_j},$$

где f_j^* — эмпирические частоты попадания результатов наблюдения x_i в заданный интервал; f_j — теоретические частоты попадания результатов наблюдения x_i в заданный интервал.

Теоретические частоты равномерного распределения байта в файле описываются выражениями:

$$f_1 = \frac{1}{b^* - a^*}(x_1' - a^*),$$

$$f_2 = f_3 = \ldots f_{m-1} = \frac{1}{b^* - a^*}(x_j' - x_{j-1}'),$$

$$f_m = \frac{1}{b^* - a^*}(b^* - x_{m-1}'),$$

где a^* и b^* — оценки границ интервала, в котором наблюдались все возможные значения результатов наблюдения x; x_j' – граница интервала j.

Оценить границы интервала можно по формулам:

$$a^* = \bar{x} - \sqrt{3}S,$$

$$b^* = \bar{x} + \sqrt{3}S.$$

Сравнение эмпирических и теоретических частот с помощью критерия Пирсона происходит при уровне значимости 0,05 и количестве степеней свободы $k = m - 3$. Очевидно, что чем значительнее отличие эмпирических и теоретических частот, тем χ^2 больше, и наоборот, если расхождение незначительно, то χ^2 должно быть малым.

Метод Монте-Карло для аппроксимации числа π используется для вычисления характеристик распределения байтов в файле, где файл с байтами является генератором случайных чисел. Выдвигается следующая гипотеза: чем точнее вычисленное значение числа π к эталонному, тем равномернее распределены байты в файле.

Вычисление данного критерия происходит следующим образом: в квадрат со стороной $a = 2R$ вписывается круг радиуса R так, чтобы центры круга и квадрата совпадали. Точки со случайными координатами (z, y) вписываются в квадрат. Геометрически, вероятность того, что точка попадет в круг, описывается выражением:

$$P_т = \frac{s_{кр}}{s_{кв}} = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4}.$$

Эмпирически, вероятность попадания точки в круг вычисляется с помощью формулы:

$$P_э = \frac{N_{кр}}{N_{кв}},$$

где $N_{кр}$ — множество точек, попавших в круг; $N_{кв}$ — множество всех точек. Для вычисления координат, интерпретируем последовательность

$x_i x_{i+1} x_{i+2} x_{i+3} x_{i+4} x_{i+5} x_{i+6} x_{i+7}$ из 8 Б в исполняемом файле следующим образом:

— $z = x_i 2^{24} + x_{i+1} 2^{16} + x_{i+2} 2^8 + x_{i+3}$ — координата z;

— $y = x_{i+4} 2^{24} + x_{i+5} 2^{16} + x_{i+6} 2^8 + x_{i+7}$ — координата y.

Точка попадает круг, если

$$z^2 + y^2 \leqslant R^2,$$

где радиус круга $R = 256^4 - 1$.

Таким образом, возможно измерить процент ошибки в приближенном значении числа π, чтобы оценить равномерность распределения байтов в файле.

Описание эксперимента

Для оценки применимости предложенных методов сформирована выборка исполняемых файлов встроенного ПО фирмы ZyXeL[1] в количестве 1 524 экземпляра. Загруженные исполняемые файлы распакованы с помощью штатных средств операционной системы. Далее сформированы выборки для каждого алгоритма упаковки данных (BZ2, GZ, LZMA, ZIP) и шифрования (AES256, режим шифрования CBC). Для каждого файла в выборке вычислены значения энтропии (H), критерий согласия Пирсона (χ^2) и число π. Результаты статистических тестов представлены в таблице.

Результаты эксперимента

Значение энтропии для зашифрованных данных лежит в диапазоне от 7,999382 до 8,000000, тогда как энтропия сжатых файлов — в диапазоне от 7,923749 до 7,999999 (рис. 1). Это говорит о том, что метод обнаружения зашифрованных данных на основе энтропийного анализа может привести к ложноположительным результатам, за счет попадания в область определения зашифрованных файлов некоторых типов сжатых файлов.

Среднее значение критерия согласия Пирсона для зашифрованных файлов составляет 255. Это в несколько раз меньше, чем для сжатых файлов (рис. 2). Данный критерий чрезвычайно чувствителен к отклонениям равномерно распределенной случайной величины, поэтому его использование для обнаружения зашифрованных файлов является обоснованным.

После выполнения серии тестов можно сделать вывод, что аппроксимация числа π по методу Монте-Карло является эффективным критерием для обнаружения зашифрованных данных. Зашифрованные файлы позволяют вычислить число π с точностью до двух знаков после запятой, тогда как сжатые файлы — с точностью в один знак после запятой (рис. 3).

Применение совокупности критериев в рамках предложенного метода к экспериментальной выборке файлов различных размеров, которые были подвергнуты алгоритмам сжатия и шифрования, показало следующие корреляции:

— отклонение нулевой гипотезы о равномерном распределении байтов в файле при большом проценте ошибок в вычислении числа π методом Монте-Карло (ошибка более 0,008 %) являются явными признаками сжатия;

— очень точные вычисления числа π методом Монте-Карло (ошибка менее 0,008 %) в случае подтверждения нулевой гипотезы о равномерном распределении байтов в файле являются верными признаками шифрования.

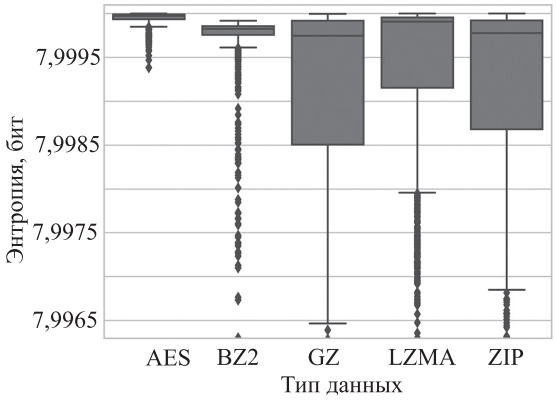

Рис. 1. Энтропия исполняемых файлов встроенного программного обеспечения, подвергнутых различным алгоритмам сжатия/шифрования

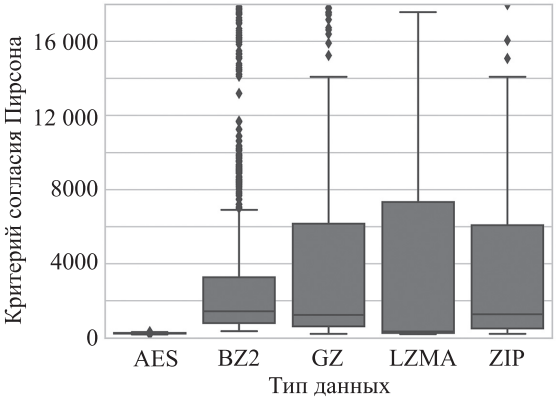

Рис. 2. Критерий согласия Пирсона исполняемых файлов встроенного программного обеспечения, подвергнутых различным алгоритмам сжатия/шифрования

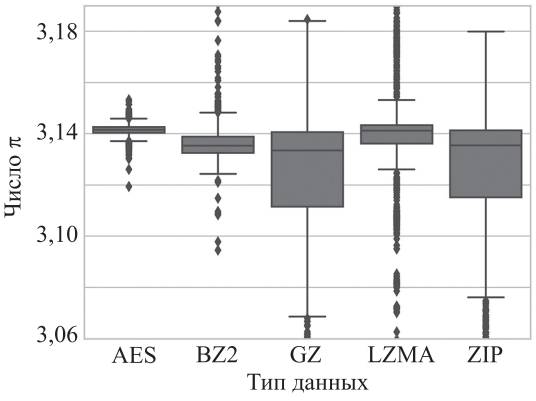

Рис. 3. Аппроксимация числа π методом Монте-Карло на основе исполняемых файлов встроенного программного обеспечения, подвергнутых различным алгоритмам сжатия/шифрования

[1] Index of / [Электронный ресурс]. Режим доступа: ftp://ftp.zyxel.com, свободный. Яз. англ. (дата обращения: 02.05.2020).

Таблица. Основные статистические данные анализа исполняемых файлов встроенного ПО фирмы ZyXeL, подвергнутых различным алгоритмам сжатия и шифрования

Файл в выборке	Оценка математического ожидания ($\bar{x}$)	Оценка среднего квадратического отклонения (S)	0,25-квантиль	0,5-квантиль	0,75-квантиль
Алгоритм шифрования AES					
H	7,999951	0,000061	7,999933	7,999971	7,999989
χ^2	255,440598	22,955680	239,479144	254,520703	270,772624
π	3,141443	0,002455	3,140416	3,141572	3,142601
Алгоритм сжатия BZ2					
H	7,999480	0,002663	7,999757	7,999824	7,999854
χ^2	4 040,869287	7 409,096520	801,360685	1 442,502750	3277,004226
π	3,136734	0,008038	3,132583	3,135350	3,138829
Алгоритм сжатия GZ					
H	7,998933	0,001820	7,998507	7,999748	7,999918
χ^2	19 604,666782	61 413,613152	619,908408	1 249,117301	6 166,157590
π	3,123964	0,030434	3,111571	3,133410	3,140669
Алгоритм сжатия LZMA					
H	7,98952	0,002345	7,999154	7,999907	7,999956
χ^2	19 138,252319	66 166,773188	262,410740	351,780713	7 336,759937
π	3,132686	0,032935	3,136211	3,141248	3,143325
Алгоритм сжатия ZIP					
H	7,999051	0,001607	7,998682	7,999778	7,999921
χ^2	16 967,341792	48 536,247396	519,091977	1281,088337	6087,787837
π	3,125408	0,028862	3,115215	3,135516	3,141345

Заключение

В работе рассмотрены способы определения упакованных и зашифрованных данных на основе результатов статистических тестов. Предложенный метод отличается от разработанных ранее вычислением дополнительных критериев. Используя предложенные критерии, с высокой степенью уверенности можно утверждать, какому алгоритму (сжатия или шифрования) было подвергнуто встроенное программного обеспечение, так как они более чувствительны к отклонениям равномерно распределенной случайной величины.

Разработанный метод применим как в рамках анализа вредоносного программного обеспечения, так и в рамках поиска и идентификации дефектов безопасности во встроенном программном обеспечении.

Литература

1. Langner R. Stuxnet: Dissecting a cyberwarfare weapon // IEEE Security & Privacy. 2011. V. 9. N 3. P. 49–51. doi: 10.1109/MSP.2011.67
2. Falliere N., Murchu L.O., Chien E. W32. stuxnet dossier // White paper, Symantec Corp., Security Response. 2011. V. 5. N 6. P. 29.
3. Antonakakis M., April T., Bailey M., Bernhard M., Bursztein E., Cochran J., Durumeric Z., Halderman J.A., Invernizzi L., Kallitsis M., Kumar D., Lever C., Ma Z., Mason J., Menscher D., Seaman C., Sullivan N., Thomas K., Zhou Y. Understanding the mirai botnet // Proc. 26th USENIX Security Symposium. 2017. P. 1093–1110.
4. Kolias C., Kambourakis G., Stavrou A., Voas J. DDoS in the IoT: Mirai and other botnets // Computer. 2017. V. 50. N 7. P. 80–84. doi: 10.1109/MC.2017.201
5. Cui A. Costello M., Stolfo S.J. When firmware modifications attack: A case study of embedded exploitation // Proc. 20th Annual Network & Distributed System Security Symposium (NDSS). 2013. P. 1–13.
6. Chen D.D., Egeley M., Woo M., Brumley D. Towards automated dynamic analysis for linux-based embedded firmware // Proc. of the Network and Distributed System Security Symposium (NDSS'16). 2016. P. 1–16. doi: 10.14722/ndss.2016.23415

References

1. Langner R. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 2011, vol. 9, no. 3, pp. 49–51. doi: 10.1109/MSP.2011.67
2. Falliere N., Murchu L.O., Chien E. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 2011, vol. 5, no. 6, pp. 29.
3. Antonakakis M., April T., Bailey M., Bernhard M., Bursztein E., Cochran J., Durumeric Z., Halderman J.A., Invernizzi L., Kallitsis M., Kumar D., Lever C., Ma Z., Mason J., Menscher D., Seaman C., Sullivan N., Thomas K., Zhou Y. Understanding the mirai botnet. *Proc. 26th USENIX Security Symposium*, 2017, pp. 1093–1110.
4. Kolias C., Kambourakis G., Stavrou A., Voas J. DDoS in the IoT: Mirai and other botnets. *Computer*, 2017, vol. 50, no. 7, pp. 80–84. doi: 10.1109/MC.2017.201
5. Cui A. Costello M., Stolfo S.J. When firmware modifications attack: A case study of embedded exploitation. *Proc. 20th Annual Network & Distributed System Security Symposium*, 2013, pp. 1–13.
6. Chen D.D., Egeley M., Woo M., Brumley D. Towards automated dynamic analysis for linux-based embedded firmware. *Proc. of the Network and Distributed System Security Symposium (NDSS'16)*, 2016, pp. 1–16. doi: 10.14722/ndss.2016.23415

7. Costin A., Zaddach J., Francillon A., Balzarotti D. A large-scale analysis of the security of embedded firmwares // Proc. 23rd USENIX Security Symposium. 2014. P. 95–110.
8. Feng Q., Zhou R., Xu C., Cheng Y., Testa B., Yin H. Scalable graph-based bug search for firmware images // Proc. 23rd ACM SIGSAC Conference on Computer and Communications Security. 2016. P. 480–491. doi: 10.1145/2976749.2978370
9. Матвеева В.С. Статистические особенности данных, зашифрованных с помощью программных средств криптографической защиты информации, и способ их обнаружения // Информация и безопасность. 2015. Т. 18. № 1. С. 119–122.
10. Wu Y., Zhou Y., Saveriades G., Agaian S., Noonan J.P., Natarajan P. Local Shannon entropy measure with statistical tests for image randomness // Information Sciences. 2013. V. 222. P. 323–342. doi: 10.1016/j.ins.2012.07.049
11. Lyda R., Hamrock J. Using entropy analysis to find encrypted and packed malware // IEEE Security and Privacy. 2007. V. 5. N 2. P. 40–45. doi: 10.1109/MSP.2007.48
12. Jeong G., Choo E., Lee J., Bat-Erdene M., Lee H. Generic unpacking using entropy analysis // Proc. 5th International Conference on Malicious and Unwanted Software (MALWARE 2010). 2010. P. 98–105. doi: 10.1109/MALWARE.2010.5665789
13. Матвеева В.С. Критерий оценки содержимого файлов различных форматов на предмет их близости к случайным данным // Безопасность информационных технологий. 2015. Т. 22. № 1. С. 106–108.
14. Матвеева В.С. Новый способ различения сжатых форматов файлов от зашифрованных файлов // Проблемы информационной безопасности. Компьютерные системы. 2015. № 4. С. 131–139.
15. Алексеев И.В., Платонов В.В. Выявление зашифрованных исполняемых файлов на основе анализа энтропии для определения меры случайности байтовых последовательностей // Проблемы информационной безопасности. Компьютерные системы. 2016. № 4. С. 74–79.

7. Costin A., Zaddach J., Francillon A., Balzarotti D. A large-scale analysis of the security of embedded firmwares. *Proc. 23rd USENIX Security Symposium*, 2014, pp. 95–110.
8. Feng Q., Zhou R., Xu C., Cheng Y., Testa B., Yin H. Scalable graph-based bug search for firmware images. *Proc. 23rd ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 480–491. doi: 10.1145/2976749.2978370
9. Matveeva V.S. Statistical features of data encrypted by cryptographic information protection software, and their detection method. *Informatsiya i Bezopasnost*, 2015, vol. 18, no. 1, pp. 119–122. (in Russian)
10. Wu Y., Zhou Y., Saveriades G., Agaian S., Noonan J.P., Natarajan P. Local Shannon entropy measure with statistical tests for image randomness. *Information Sciences*, 2013, vol. 222, pp. 323–342. doi: 10.1016/j.ins.2012.07.049
11. Lyda R., Hamrock J. Using entropy analysis to find encrypted and packed malware. *IEEE Security and Privacy*, 2007, vol. 5, no. 2, pp. 40–45. doi: 10.1109/MSP.2007.48
12. Jeong G., Choo E., Lee J., Bat-Erdene M., Lee H. Generic unpacking using entropy analysis. *Proc. 5th International Conference on Malicious and Unwanted Software (MALWARE 2010)*, 2010, pp. 98–105. doi: 10.1109/MALWARE.2010.5665789
13. Matveeva V.S. The criterion for assessing the file content for its proximity to the random data. *IT Security*, 2015, vol. 22, no. 1, pp. 106–108. (in Russian)
14. Matveeva V.S. A new approach to differentiate compressed file formats from encrypted files. *Information Security Problems. Computer Systems*, 2015, no. 4, pp. 131–139. (in Russian)
15. Alekseev I.V., Platonov V.V. Identification of the encrypted executable files based on the entropy analysis for detection value randomness of byte sequences. *Information Security Problems. Computer Systems*, 2016, no 4, pp. 74–79. (in Russian)

Авторы

Югансон Андрей Николаевич — ассистент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57211977888, ORCID ID: 0000-0002-8231-5684, a_yougunson@itmo

Authors

Andrei N. Iuganson — Assistant, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57211977888, ORCID ID: 0000-0002-8231-5684, a_yougunson@itmo.ru