УДК 004.056.57 Вестник УрФО № 1(35) / 2020, с. 35–47

Гибилинда Р. В.

DOI: 10.14529/secur200104

КЛАСТЕРИЗАЦИОННЫЙ МЕТОД ИДЕНТИФИКАЦИИ ВОЗДЕЙСТВИЙ НА ФАЙЛЫ С ПРИМЕНЕНИЕМ АЛГОРИТМА К-СРЕДНИХ, ИСПОЛЬЗУЕМЫЙ ПРИ РАССЛЕДОВАНИИ ИНЦИДЕНТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В статье представлен кластеризационный метод идентификации воздействий на файлы, используемый при расследовании инцидентов информационной безопасности. Предлагаемый метод основан на применении алгоритма кластеризации k-средних с адаптированным алгоритмом автоматического определения оптимального количества кластеров, которые описывают воздействия на файлы. В статье рассмотрен процесс подготовки входных данных, полученных из записей журнала изменений тома $UsnJrnl, а также алгоритм выявления сложных комплексных воздействий на файлы, основанный на поиске взаимосвязей между кластерами. Предлагаемый кластеризационный метод имеет ярко выраженный автоматизированный характер, что позволяет специалисту, проводящему расследование инцидента информационной безопасности, ускорить процесс выявления и ликвидации последствий инцидента.

Ключевые слова: *расследование инцидентов информационной безопасности, воздействие на файл, кластеризация.*

Gibilinda R. V.

A CLUSTERING METHOD FOR IDENTIFYING FILE IMPACTS BASED ON THE K-MEANS ALGORITHM USED IN INFORMATION SECURITY INCIDENTS INVESTIGATION

The article presents a clustering method for identifying file impacts used in information security incidents investigation. The proposed method is based on application of k-means clusterization algorithm with adapted automatic optimal cluster number determination algorithm. Precisely defined clusters amount allows to group data to describe file impacts. The article discusses preparation process of input data obtained from $UsnJrnl volume changes log records, as well as the algorithm for identifying complex file impacts based on the search for relationships between clusters. The proposed clustering method has a pronounced automated character, which allows a specialist that carries out an information security incident investigation to speed up the process of identifying and eliminating the consequences of an incident.

Keywords: *information security incident response, file impact, clusterization.*

Развитие информационных технологий (ИТ) и укрепление их влияния на повседневную деятельность человеческого общества зачастую сопровождается повышением интереса злоумышленников к совершению противоправных действий в сфере ИТ. Рост объема информации, обрабатываемой в информационных системах (ИС), а также непрерывное развитие злоумышленниками методов и средств совершения киберпреступлений неизбежно приводят к необходимости постоянного совершенствования применяемых мер обеспечения информационной безопасности (ИБ).

Существующие методы и средства противодействия угрозам ИБ, направленные на обеспечение конфиденциальности, целостности и доступности информации, в силу ограниченности функционала, ошибок конфигурации и эксплуатации, не всегда могут полностью решить поставленные задачи по защите информации. Выявленные злоумышленниками недостатки применяемых методов и средств противодействия угрозам ИБ способствуют совершению киберпреступлений, результатом которых являются инциденты ИБ. Расследование инцидентов позволяет установить причины их возникновения, ликвидировать последствия, а также сформировать рекомендации по усовершенствованию мер обеспечения ИБ.

В процессе расследования инцидента ИБ с применением различных методов и средств [1] собирается и анализируется значительное количество данных, способствующих установлению причин инцидента, его последствий, а также действий злоумышленника: запуск и активность процессов, воздействия на файлы, установка сетевых соединений и передача данных и др. Указанные данные хранятся в нескольких источниках — «массивах данных»: журналы аудита [2], журналы событий операционной системы [3], записи о последних открытых файлах и запущенных программах, журналы средств защиты информации и др.

Одной из важных составляющих расследования инцидента ИБ является анализ массивов данных, в частности, идентификация воздействий на файлы. Это связано с тем, что

в ИС информация хранится в виде файлов. В рамках статьи под идентификацией воздействий на файлы будем понимать процесс, в результате которого определяется порядок изменения параметров, характеризующих файл.

Массивы данных, обработка и анализ которых лежат в основе процесса расследования инцидента ИБ, обладают рядом недостатков:

• отсутствие исчерпывающего и единоформатного набора параметров, характеризующих файл;

• возможность искажения данных массива даже в процессе функционирования операционной системы;

• отсутствие возможности автоматизированного анализа некоторых массивов данных.

Совокупность указанных недостатков порождает необходимость поиска новых методов и средств анализа данных, содержащихся в массивах, для достижения полноценного результата в расследовании инцидента ИБ. В работе [4] авторами рассмотрены проблемы формализации набора параметров, характеризующих файл, и верификации1 массива данных.

В тех ситуациях, когда данных в массиве немного, возможен ручной анализ. Тем не менее, анализ «вручную» обладает рядом недостатков:

• человеческий фактор – специалист, проводящий анализ массивов данных может упустить из виду информацию, связанную с произошедшим инцидентом;

• начало инцидента могло произойти значительно раньше, чем основная совокупность воздействий на файлы, обрабатываемые в ИС, в результате чего потенциальный «источник инцидента» не попадет в выборку по временному интервалу.

В работе [5] для идентификации воздействий на файлы использовался журнал изменений тома $UsnJrnl в качестве массива данных, который обладает рядом преимуществ:

• полнота — журнал содержит подробную информацию о действиях, совершенных по отношению к файлам;

• достоверность — системный файл $UsnJrnl защищен операционной системой от несанкционированных изменений со стороны пользователя;

• простота обработки данных, что играет немаловажную роль в последующей автоматизации их анализа.

В рамках настоящей статьи автором предлагается кластеризационный метод идентификации воздействий на файлы с использованием алгоритма k-средних при обработке записей журнала изменений тома $UsnJrnl.

Журнал $UsnJrnl представляет собой разреженный файл, расположенный в каталоге $Extend. Он состоит из двух потоков данных: $Max и $J. Первый содержит служебные данные о журнале и не представляет интереса в рамках расследования инцидента. Второй, напротив, является для специалиста, проводящего расследование инцидента ИБ, ценным источником информации, так как состоит из набора записей об изменениях, произошедших с файлами как в отношении содержимого, так и служебной информации [6].

Согласно [4], параметры, характеризующие файл j, представлены вектором:

$$V_j = \left\{ I_j, D_j, N_j, C_j, X_j \right\}, \qquad (1)$$

где:

• I_j – идентификатор файла – уникальное числовое значение, содержащееся в служебной информации о файле, используемое драйвером файловой системы для однозначного определения файла;

• D_j – идентификатор родительского каталога файла – уникальное числовое значение, используемое драйвером файловой системы для установления однозначного соответствия между файлом и каталогом, в котором файл расположен;

• N_j – имя файла – битовая строка, используемая драйвером файловой системы для представления файла пользователю;

• C_j – содержимое файла – битовая строка, являющаяся информацией, хранимой в файле;

• X_j – иная служебная информация о файле – набор числовых значений, являющихся служебной информацией о файле, зависящий от типа файловой системы.

Записи журнала $UsnJrnl содержат в явном виде только часть параметров (компонентов вектора V_j), тем не менее, поле записи R_j может быть косвенно использовано для определения изменений в компонентах C_j и X_j в рамках решения задачи верификации воз-

[1] В рамках статьи под верификацией массива данных будем понимать процесс выявления комбинаций параметров, характеризующих файл, возникновение которых невозможно в процессе штатного заполнения массива данными.

действия на файл. Формат записи представлен в таблице 1. Курсивом выделены поля, информация в которых позволяет идентифицировать файл и действия, осуществленные по отношению к нему. Полужирным шрифтом выделены поля, которые являются основой для предлагаемого автором кластеризационного метода.

Таблица 1

Формат записи журнала изменений тома $UsnJrnl

Смещение, байт	Размер, байт	Описание
0x00	4	Размер записи
0x04	2	Версия записи
0x06	2	Версия программного обеспечения, которым запись создана
0x08	8	Идентификатор файловой записи (Ij)
0x10	8	Идентификатор родительского каталога (Dj)
0x18	8	**Номер записи (Uj)**
0x20	8	**Временная отметка создания записи (T)**
0x28	4	Идентификатор действия (Rj)
0x2C	4	Тип источника записи
0x30	4	Идентификатор безопасности
0x34	4	Атрибуты файла
0x38	2	Длина имени файла *
0x3A	2	Начало имени файла в записи
0x3C	*	Имя файла (Nj)

В рамках настоящего исследования был проанализирован порядок формирования записей и выявлена зависимость, описываемая выражением:

$$\Psi = f(\tau, K_j), \qquad (2)$$

где Ψ – множество значений номеров записей журнала изменений тома $UsnJrnl, имеющих отношение к файлу *j*, причем $\forall j\, U_j \in \Psi$; K_j – количество записей журнала, имеющих отношение к файлу *j*; τ – множество временных интервалов различной величины, в рамках которых осуществляются действия по отношению к файлу *j*, описываемые записями журнала изменений тома $UsnJrnl, причем $T \in \tau$.

Пример зависимости (2), аппроксимированный кусочно-заданной функцией, представлен на рис. 1. По оси абсцисс отложены отнормированные к минимальному значения *T*, а по оси ординат – отнормированные к минимальному значения U_j нескольких записей журнала $UsnJrnl. Окружностями обозначены значения (T, U_j) каждой записи для файла *j*.

Выявленная зависимость обладает следующими свойствами:

• нелинейная – за фиксированный интервал времени может быть совершено произвольное количество воздействий на файлы, что повлечет за собой появление соответствующего количества записей журнала $UsnJrnl;

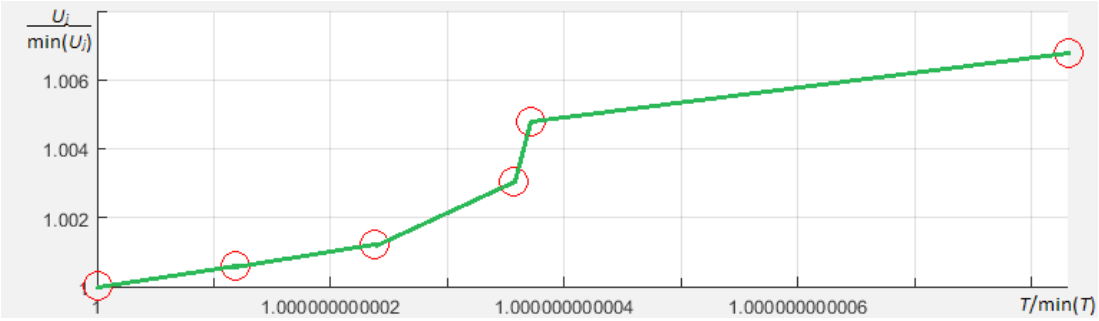

Рис. 1. Зависимость отнормированных к минимальному значений Uj от количества записей журнала $UsnJrnl в нескольких отнормированных к минимальному временных интервалах

• возрастающая – номер записи постоянно увеличивается. Несмотря на наличие ограничения в размерности поля U_j (8 байт), ис-

черпать его ресурс при повседневной активности пользователя затруднительно.

Нелинейность позволяет определить период наибольшей активности пользователя/процесса по отношению к файлу. Крутизна графика функции в выбранном фиксированном временном интервале характеризует интенсивность воздействий на файлы. Рост значения номера записей U_j говорит об интенсивности работы с файлами в целом. При совершении простых воздействий крутизна графика функции незначительна. К ним можно отнести работу с документами в формате txt, простые файловые операции, такие как создание/удаление/переименование файла. Сложные комплексные воздействия на файлы существенно увеличивают крутизну графика. Примерами могут служить редактирование «офисного» документа в текстовом процессоре Microsoft Word, разархивирование файлов и т. д. Изменяемое значение крутизны графика и нелинейность позволяют рассмотреть использование математических алгоритмов для проведения анализа выявленной зависимости.

Предлагаемый метод базируется на реализации процесса автоматизированного объединения нескольких записей в совокупность с целью получения воздействия на файл. Для этого возможно использование различных типов алгоритмов машинного обучения: кластеризации, классификации, нейронных сетей и др. В рамках исследования были рассмотрены алгоритмы кластеризации *k*-средних, *k*-медоидов, DBSCAN.

DBSCAN [7], несмотря на возможность работы с кластерами произвольной формы, требует точного задания двух параметров: ε и minPts. В случае неоптимального задания этих параметров данные либо не будут кластеризованы, либо сольются в один кластер. Алгоритм показал отрицательные результаты при группировке данных с различной плотностью распределения в пространстве. Модификации алгоритма, например, Generalized DBSCAN или Hierarchical DBSCAN существенно уступают в скорости работы по сравнению с *k*-средних и *k*-медоидов в рамках решаемой задачи.

Алгоритм *k*-медоидов похож на *k*-средних как по принципу работы, так и по скорости выполнения, отличие заключается лишь в порядке выбора центра кластера – центром выбирается одна из точек кластера, а не его «центр масс». В связи с особенностью выбора центра кластера, алгоритм *k*-медоидов показал отрицательные результаты в тех случаях, когда точки в пространстве распределены в формах половины и четверти окружности, что не является редкостью в связи с особенностями зависимости (2).

Для автоматизации процесса идентификации воздействий на файлы в рамках настоящего исследования был выбран алгоритм кластеризации *k*-средних по соотношению скорость работы/сложность подготовки входных данных/эффективность полученных результатов.

Объем записей в журнале $UsnJrnl может превышать 600 тысяч, что, в свою очередь, затрудняет использование алгоритма *k*-средних для группировки записей в рамках предлагаемого метода идентификации воздействий на файлы при расследовании инцидентов информационной безопасности – продолжительность анализа записей может достигать нескольких часов в зависимости от конфигурации компьютера. Для повышения эффективности применения алгоритма *k*-средних в отношении решаемой задачи необходимо подготовить входные данные, выбрать оптимальное количество кластеров, которые представляют собой воздействия на файлы, и проанализировать связи между кластерами для выявления сложных комплексных воздействий.

Для повышения скорости анализа автором предлагается использовать в качестве входных данных алгоритма *k*-средних «порции» записей из $UsnJrnl, предварительно разделенных согласно идентификаторам файлов I_j. Помимо выборки данных по I_j необходимо также учесть особенность работы драйвера файловой системы (ФС) NTFS, а именно активное использование им существующих освобожденных файловых записей вместо выделения новых. Указанная особенность может привести к получению неверного результата.

Для предотвращения ошибок анализа записей журнала $UsnJrnl из-за особенностей работы драйвера ФС NTFS при разделении записей по идентификаторам файлов I_j предлагается алгоритм разделения записей журнала $UsnJrnl на блоки (алгоритм № 1):

1. Выбрать записи в соответствии с идентификатором I_j.

2. Создать буфер «разделенных» записей – буфер 1 и назначить его текущим.

3. Провести в цикле последовательную

проверку выбранных записей на наличие значения R_j = 0x00000200, свидетельствующего об удалении файла.

4. Если значение 0x00000200 в поле R_j записи не найдено, то поместить запись в текущий буфер.

5. В противном случае создать следующий буфер (буфер 2), назначить его текущим. Если значение 0x00000200 в поле R_j записей появляется m раз, то создать m+1 буферов, чтобы разделить записи на несколько частей в соответствии с особенностями работы драйвера ФС NTFS.

6. Поместить запись в текущий буфер.

7. Разделенные на части записи сохранить в виде блоков для последующего анализа.

Схема алгоритма представлена на рис. 2.

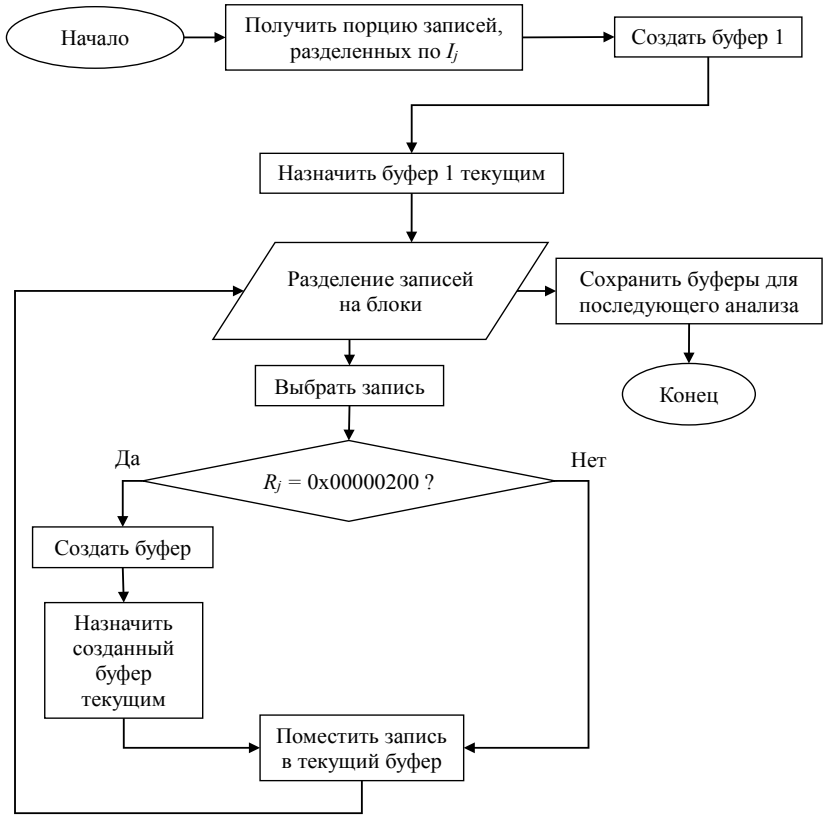

Рис. 2. Алгоритм №1 для учета особенностей драйвера ФС NTFS

После разделения записей в связи с особенностями работы драйвера ФС NTFS значения полей U_j и T каждой разделенной части следует отнормировать к соответствующему минимуму. Нормировка необходима для того, чтобы проводить анализ зависимости (2) в числовых значениях одного порядка.

Необходимым условием работы алгоритма *k*-средних является указание желаемого количества кластеров *k* для группировки входного множества значений. От выбора значения *k* зависит корректность получаемых результатов. Пример задания верного количества кластеров указан на рис. 3, где видно, что точки в пространстве были сгруппированы и обозначены цветом оптимальным образом. При установке значения *k*, отличного от 3, для группировки тех же данных, полученный результат окажется некорректным.

Существуют несколько методов, позволяющих определить оптимальное значение *k*: метод «локтя», удовлетворение информационным критериям Акаике или Шварца, метод «силуэтов», построение дендрограммы кластеризации, метод градиентного спуска и др. [8]. Описание достоинств и недостатков указанных методов выходит за рамки данной статьи. Для решения задачи выбора оптимального значения *k* выбран принцип минимальной длины описания (MDL) [9–11].

Принцип MDL основывается на следующей идее: «Любая закономерность в задан-

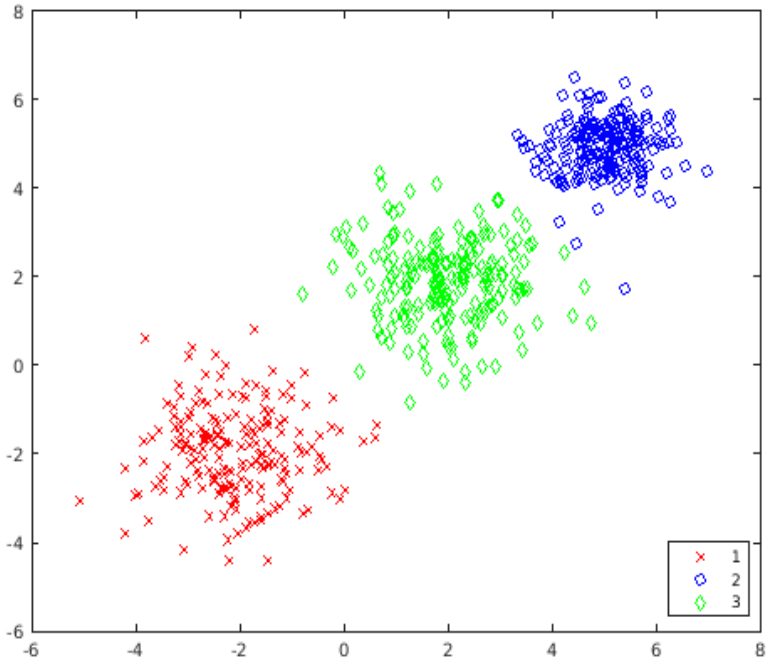

Рис. 3. Разделение точек на плоскости при значении $k = 3$

ном наборе данных может быть использована для сжатия данных, то есть описания данных с использованием меньшего набора символов, чем нужно для описания данных буквально» [11]. В роли количественной оценки величины, которая необходима для описания набора данных, выступает длина описания L(x). В соответствии с принципом MDL, нахождение минимального значения L(x) позволяет определить оптимальный способ описания данных. В статье автор предлагает использовать принцип MDL и нахождение минимального значения L(x), то есть, в отношении решаемой задачи, оптимального количества кластеров k, которые описывают воздействия на файлы.

В работе [12] для расчета длины описания исследуемых случайно распределённых значений предлагается использовать следующую формулу:

$$L(x) = -\sum_{x \in X} \log_2 p(x) + \frac{1}{2} P \log_2 |X|, \quad (3)$$

где X – вектор исследуемых значений, *p(x)* – плотность вероятности исследуемого значения *x*, *P* – количество переменных параметров в формуле плотности вероятности.

В формуле (3) длина описания складывается из «стоимостей» описания. Первое слагаемое является логарифмической функцией правдоподобия исследуемых значений и описывает «стоимость» описания этих значений. Второе слагаемое представляет собой «стоимость» описания мощности множества значений, заданных в соответствии с предполагаемым/выбранным законом плотности вероятности.

Формула (3) была адаптирована в работе [13] для поиска оптимального значения кластеров в алгоритме k-медоидов. Адаптированная формула описывается выражением:

$$L(x) = -\sum_{x \in X} \log_2 p(\|x - c_x\|) + (\frac{1}{2} P + k) \log_2 |X|, \quad (4)$$

где X – вектор элементов исследуемых значений, $p(\|x - c_x\|)$ – плотность вероятности расстояния между исследуемым значением x (точки с заданными координатами) и ближайшим к нему центром кластера c_x, *P* – количество переменных параметров в формуле плотности вероятности, *k* – количество кластеров.

Формула (4) может быть использована для расчета минимальной длины описания в целях определения оптимального значения k. По мнению автора статьи, формула (4) обладает избыточностью, которая усложняет решение задачи определения оптимального значения k для алгоритма k-средних.

В ходе настоящего исследования проведен анализ влияния слагаемых формулы (4) на определение оптимального значения k в рамках предлагаемого метода и получены следующие результаты:

1. Данные, которые попадают на вход алгоритма *k*-средних, предварительно разделяются на «порции», что существенно уменьша-

ет абсолютную величину слагаемого $\frac{1}{2}P\log_2|X|$ и его вклад в значение длины описания.

2. Абсолютное значение длины описания *L(x)* не важно для определения оптимального количества кластеров, т.к. из полученных значений *L(x)* выбирается минимальное.

3. В ходе экспериментов расчет *L(x)* проводился для нескольких законов плотности вероятности (Probability Density Function – PDF), в отличие от работы [13]. Необходимость выбора из нескольких PDF обусловлена тем, что при расчете *L(x)* в рамках предлагаемого метода анализа записей журнала $UsnJrnl предполагаемое распределение величин расстояний между значениями *x* (точек с координатами *(T, U_j)*) и центрами ближайших кластеров c_x (рассчитанных алгоритмом *k*-средних точек с координатами (*T', U_j'*)) задается специалистом. Влияние на распределение величин расстояний оказывают типы воздействий на файлы, а также интенсивность воздействий. Результаты экспериментов по выбору наилучших значений параметров PDF представлены в таблице 2.

ных экспериментов, автором предлагается адаптированная формула для расчета длины описания, используемая при определении количества кластеров *k*:

$$L(x) = -\sum_{m=1}^{M}\log_2 p(\|x_m - c_m\|) + k\log_2 M \ , \quad (5)$$

где M – количество записей журнала $UsnJrnl с отнормированными значениями *T* и U_j в «порции», $p(\|x_m - c_m\|)$ – плотность вероятности расстояния между исследуемым значением xm (точки с координатами $\langle T, U_j \rangle_m$) и ближайшим к нему центром кластера cm (точки с рассчитанными координатами $\langle T', U_j' \rangle_m$).

Для определения оптимального количества кластеров с учетом формулы (5) и проведения кластеризации «порции» записей журнала $UsnJrnl необходимо воспользоваться алгоритмом определения значения *k* (алгоритм № 2):

1. Выбрать «порцию» записей журнала $UsnJrnl, предварительно обработанных с использованием алгоритма № 1.

2. Выбрать PDF с оптимальными параметрами согласно таблице 2.

Таблица 2

Значения параметров PDF, используемых при определении оптимального количества кластеров

PDF	Параметр	Диапазон значений параметра / Наилучшее значение параметра
Нормальное распределение	μ	0-2 / 0
	σ	0.5-1.5 / 1
Гамма-распределение	α	0.5;1;2 / 1
	β	0.8-1.1 /0.9
Распределение Стьюдента	ν	1-5 / 1
Распределение χ^2	ν	1-5 / 2
Распределение Фишера	$\nu 1$	1-2 / 1
	$\nu 2$	1-5 / 2

В ходе экспериментов было установлено, что для идентификации большинства воздействий на файлы достаточно использовать PDF нормального распределения с оптимальными значениями параметров, указанными в таблице 2, при расчёте значения *L(x)* для определения оптимального количества кластеров *k*.

С учетом анализа результатов проведен-

3. Определить количество записей *M* в «порции».

4. Запустить цикл с изменением значения *k* от 1 до *M*. На каждой итерации цикла:

a. С использованием алгоритма *k*-средних для текущего значения *k* вычислить центры кластеров cm текущей «порции» записей;

b. С использованием формулы (5) рассчитать значение *L(x)*;

c. Если значение *L(x)* текущей итерации меньше или равно значению, полученному на предыдущей итерации, то продолжить выполнение цикла. В противном случае прервать выполнение – минимальное значение *L(x)* найдено.

d. Определить оптимальное значение *k*, соответствующее минимальному *L(x)*.

тате проведенной кластеризации «порции» записей журнала $UsnJrnl выделены два воздействия на файл *j*, состоящие из 12 и 4 записей журнала изменений тома $UsnJrnl соответственно. Центры кластеров, описывающих выделенные воздействия, обозначены плюсами.

Полученные в ходе применения алгорит-

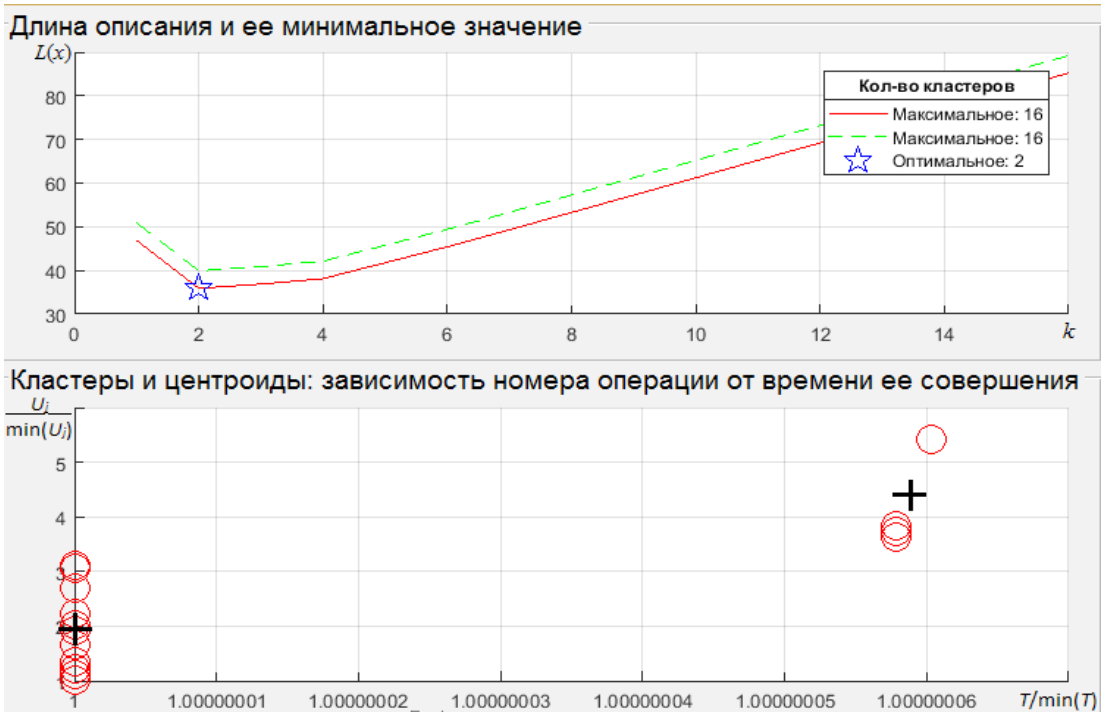

Рис. 4. Пример определения оптимального количества кластеров для идентификации воздействия на файл

5. Сохранить результаты кластеризации проведенной «порции» записей журнала $UsnJrnl при оптимальном значении *k*.

Пример определения оптимального количества кластеров и кластеризации записей журнала $UsnJrnl показан на рис. 4. На верхней оси координат: пунктирной линией показан график значений *L(x)*, рассчитанных для нескольких значений *k* по формуле (4); сплошная линия показывает график значений *L(x)*, рассчитанных по формуле (5); звездой отмечено оптимальное значение *k*. На нижней оси координат: окружностями обозначены значения xm – точки с координатами $\langle T, U_j \rangle_m$, полученные из записей журнала $UsnJrnl, плюсами – центры кластеров c_m. Примечание: для повышения наглядности результатов, выполнение цикла алгоритма № 2 не прерывалось при нахождении минимального значения *L(x)*.

Как видно из графиков верхней части рисунка, слагаемое $\frac{1}{2}P\log_2|X|$ не вносит существенного влияния в значение *L(x)*. В результате

мов № 1 и № 2 результаты свидетельствуют о том, что объединение записей журнала $UsnJrnl в кластеры, описывающие события над файлами, сокращает, в некоторых случаях существенно, объем анализируемых данных в ходе расследования инцидента ИБ.

При предварительной подготовке данных алгоритмом № 1 было сделано допущение о возможности разделения записей журнала $UsnJrnl по идентификаторам файловых записей. Такое допущение позволило существенно уменьшить время обработки данных с применением алгоритма *k*-средних, но в то же время снизило точность идентификации сложных комплексных воздействий, состоящих из нескольких записей журнала $UsnJrnl, в том числе имеющих отношение к различным файлам. Для того, чтобы корректно идентифицировать сложные комплексные воздействия, необходимо проанализировать связи между полученными кластерами.

Для проведения анализа взаимосвязей между кластерами в процессе идентифика-

ции сложных комплексных воздействий необходимо последовательно обработать все кластеры в целях поиска файлов с одинаковыми именами с использованием алгоритма № 3:

1. Определить полученное количество кластеров по результатам работы алгоритмов № 1 и № 2.

2. В цикле осуществить последовательный перебор кластеров. На каждой итерации цикла:

a. Выбрать кластер, извлечь из его элементов (записей журнала $UsnJrnl) все имена файлов;

b. Произвести поиск совпадений имен файлов в остальных кластерах;

c. Отсортировать кластеры по значению поля T записи журнала $UsnJrnl, расположенной в начале каждого кластера;

d. Последовательно объединить кластеры, если временной интервал между конечным элементом текущего кластера и начальным элементом следующего не превышает значения *t1*;

e. Предложить объединение кластеров, если временной интервал между конечным элементом текущего кластера и начальным элементом следующего не превышает значения *t2*;

f. Пометить все кластеры, которые участвовали в объединении на данной итерации цикла, как обработанные.

3. Вывести предложения об объединении кластеров, полученные на шаге 2.e алгоритма, специалисту, проводящему расследование инцидента ИБ.

Значения параметров *t1* и *t2* задаются в рамках предложенных значений специалистом, проводящим расследование инцидента ИБ, исходя из ожидаемой точности объединения кластеров записей журнала $UsnJrnl: чем больше значение *t1* и *t2*, тем больше кластеров будет объединено. Диапазоны рекомендуемых значений параметров:

• *t1*: 200 – 800 мс, оптимальное значение 400 мс;

• *t2*: 2 – 8 с, значение по-умолчанию 3 с.

Интерфейс программного обеспечения с предложением об объединении кластеров представлен на рис. 5.

На представленном рисунке столбцы таблиц соответствуют следующим полям записей журнала $UsnJrnl:

• столбец «ИФЗ» содержит идентификатор файла I_j, равный 43 и 44 в двух кластерах;

• столбец «ИРК ФЗ» содержит идентификатор родительского каталога D_j, равный 5 в обоих кластерах;

• столбец «Файловая операция» содержит идентификаторы действий R_j, совершенных по отношению к файлам;

• столбец «Имя файла» содержит имена файлов N_j;

• столбец «USN» содержит номера записей U_j.

Из рисунка видно, что в процессе объединения кластеров записи журнала $UsnJrnl о двух файлах с идентификаторами I_j, равными 43 и 44, попадут в объединенный кластер и будут принадлежать одному сложному комплексному воздействию на файл.

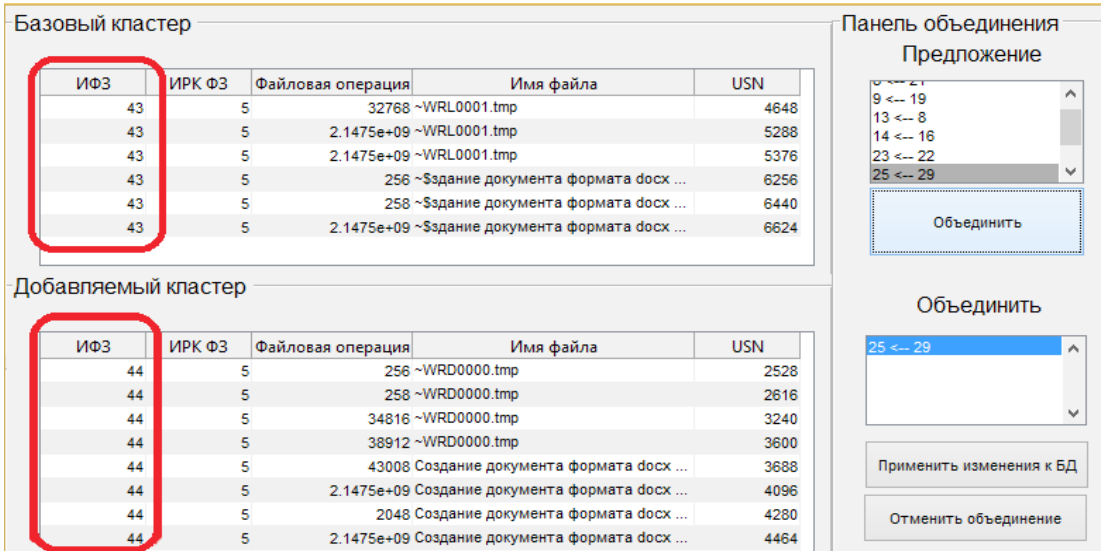

Рис. 5. Интерфейс программного обеспечения с предложением по объединению кластеров

Полученные с применением алгоритмов №№ 1 – 3 воздействия на файлы позволяют сократить объем обрабатываемой в рамках расследования инцидента ИБ информации. В ходе анализа может возникнуть вопрос достоверности полученных воздействий на файлы с позиций корректности результатов работы алгоритмов и отсутствия искажений в массиве данных. Не смотря на то, что используемый массив данных – журнал изменений тома $UsnJrnl защищен от изменений средствами операционной и файловой систем, существует возможность внесения искажений в массив с применением загрузки с внешнего носителя.

Верифицировать полученные воздействия на файлы возможно с применением событийной модели [4], на вход которой подаются:

• идентификаторы файла I_{j1} и I_{j2};

• идентификаторы родительского каталога D_{j1} и D_{j2};

• имена файла N_{j1} и N_{j2};

• признак изменения содержимого Z.

Для функционирования модели должна быть выставлена начальная маркировка μ, соответствующая равенству заданных входных параметров и установке признака изменения содержимого. Параметр Z можно определить по таблице, указанной в работе [5], исходя из значения поля R_j записи, принадлежащей верифицируемому кластеру. Параметр τ вычисляется как разница между значениями полей T смежных записей журнала $UsnJrnl, принадлежащих верифицируемому кластеру.

В результате проведения верификации специалист, проводящий расследование инцидента ИБ, проверяет последовательность записей в кластере на предмет наличия в них искажений. Стоит отметить, что записи в журнале $UsnJrnl формируются в соответствии с собственным алгоритмом драйвера файловой системы NTFS и не зависят от особенностей работы событийной модели. В связи с этим, может возникнуть несоответствие набора входных параметров искомому воздействию на файл в случае сложного комплексного воздействия. Специалист может убедиться в отсутствии искажений путем проверки последовательности значений полей U_j – номера записей при переходе между файлами в рамках одного воздействия либо смежные, либо отличаются на 1-2 записи.

Предлагаемый кластеризационный метод идентификации базируется на последовательном применении рассмотренных алгоритмов №№ 1 – 3. Для идентификации воздействий на файл предложенным методом необходимо последовательно выполнить несколько шагов:

1. Считать массив данных – журнал изменений тома $UsnJrnl, сохранив содержащиеся в нем записи в базу данных.

2. Создать множество выборок данных, разделив их по идентификаторам файлов I_j;

3. К каждой выборке данных применить алгоритм № 1, чтобы подготовить данные для проведения кластеризации алгоритмом *k*-средних;

4. Кластеризовать каждую подготовленную выборку алгоритмом *k*-средних, определив оптимальное количество кластеров k в соответствии с алгоритмом № 2;

5. Сохранить кластеры, описывающие каждую выборку, в базу данных;

6. Применить алгоритм № 3 в целях поиска взаимосвязей между кластерами в базе данных для повышения точности идентификации сложных комплексных воздействий. Применить/отклонить предложения по объединению кластеров, как результат работы алгоритма № 3.

Верифицировать кластеры с применением событийной модели.

Сформировать список идентифицированных воздействий на файлы, описанных кластерами записей журнала $UsnJrnl.

В сформированном списке найти те воздействия на файлы, которые имеют отношение к расследуемому инциденту ИБ.

Примеры полученных с помощью предлагаемого метода воздействий представлены на рис. 6 и 7.

На рис. 6 представлено воздействие на файл, в рамках которого создается файл taskhsvc.exe. Отличительным свойством указанного воздействия является изменение содержимого исполняемого файла (значения 2 и 3 в столбце «Файловая операция»), что может быть интерпретировано как активность вредоносного программного обеспечения, так и процедура обновления программного обеспечения. Для установления факта, свидетельствующего об активности вредоносного программного обеспечения, следует проанализировать остальные идентифицированные воздействия. Например, воздействие, представленное на рис. 7 говорит о том, что операционная система подверглась заражению

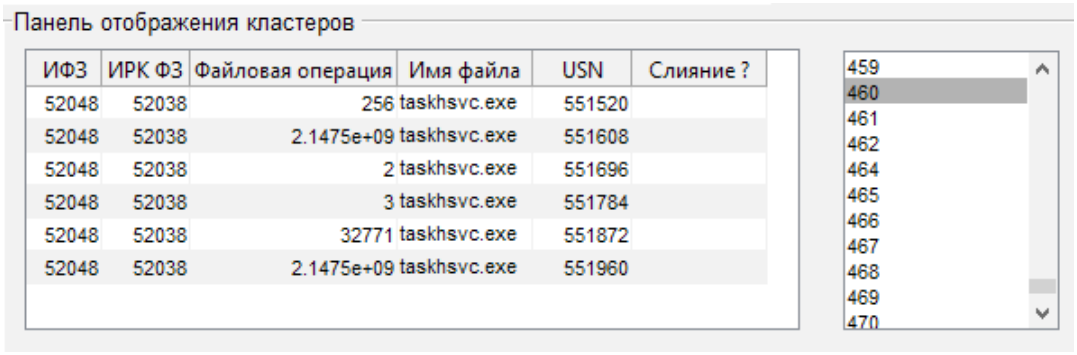

Рис. 6. Пример воздействия: создание исполняемого файла и изменение его содержимого

Рис. 7. Пример воздействия: создание файла с изображением в формате bmp

вредоносным программным обеспечением WannaCry, так как создается файл с характерным именем.

При рассмотрении нескольких воздействий специалист, проводящий расследование инцидента ИБ, может сделать вывод о том, что файл, созданный в рамках идентифицированного на рис. 6 воздействия, является копией WannaCry, замаскированной под штатный сервис операционной системы. Последующий анализ остальных воздействий на файлы поможет выявить «зараженные» файлы, и определить стратегию ликвидации последствий инцидента ИБ.

Предлагаемый кластеризационный метод позволяет идентифицировать воздействия на пользовательские и системные файлы, определять попытки внесения изменений в исполняемые файлы вредоносным программным обеспечением и т.д. Идентификация воздействий на файлы позволяет ускорить процесс расследования инцидента ИБ, а также упростить ликвидацию его последствий.

Метод имеет ярко выраженный автоматизированный характер, что способствует созданию на его основе программного обеспечения, позволяющего специалисту, проводящему расследование инцидента ИБ, сконцентрировать свое внимание на установлении причин инцидента, установлении взаимосвязей между его событиями, нежели на рутинном поиске информации о событиях инцидента. Особенностью метода является возможность проведения верификации анализируемых данных с целью установления попыток их искажения.

Литература / References

Стандарт Банка России СТО БР ИББС-1.3-2016 «Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Сбор и анализ технических данных при реагировании на инциденты информационной безопасности при осуществлении переводов денежных средств» [Электронный ресурс]. URL: http://garant.ru/products/ipo/prime/doc/71457690 (дата обращения: 10.03.2020). [Standart Banka Rossii STO BR IBBS-1.3-2016 «Obespechenie informatsionnoy bezopasnosti organizatsiy bankovskoy sistemy Rossiyskoy Federatsii. Sbor i analiz tekhnicheskikh dannykh pri reagirovanii na intsidenty informatsionnoy bezopasnosti pri osushchestvlenii perevodov denezhnykh sredstv» [Elektronnyy resurs]. URL: http://garant.ru/products/ipo/prime/doc/71457690 (data obrashcheniya: 10.03.2020)].

1. Studiawan H., Payne C., Sohel F. Graph Clustering and Anomaly Detection of Access Control Log for Forensic Purposes // Digital Investigation (2017). 2017.

2. Dwyer J., Marius Truta T. Finding Anomalies in Windows Event Logs Using Standard Deviation // 9th IEEE International on Collaborative Computing: Networking, Applications and Worksharing. 2013. P. 563 -570.

3. Гайдамакин Н. А., Гибилинда Р. В., Синадский Н. И. Событийная модель процесса идентификации воздействий на файлы при расследовании инцидентов информационной безопасности, основанная на математическом аппарате сетей Петри // Вестник СибГУТИ. 2020. № 1. (в печати).

Gaidamakin N., Gibilinda R., Sinadskiy N. File operations information collecting software package used in the information security incidents investigation // IEEE Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT-2020). Yekaterinburg, Russia, May 14 – 15, 2020. (in press). [Gaydamakin N. A., Gibilinda R. V., Sinadskiy N. I. Sobytiynaya model' protsessa identifikatsii vozdeystviy na fayly pri rassledovanii intsidentov informatsionnoy bezopasnosti, osnovannaya na matematicheskom apparate setey Petri // Vestnik SibGUTI. 2020. № 1. (v pechati)].

4. USN_RECORD_V2 – Win32 apps. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/windows/win32/api/winioctl/ns-winioctl-usn_record_v2 (дата обращения: 10.03.2020).

5. Ester M., Kriegel H., Sander J., Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise //Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press. 1996. P.226-231.

6. Determining the number of clusters in a data set. [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set (дата обращения: 10.03.2020).

7. Rissanen J. Modeling by shortest data description // Automatica, Vol. 14 (1978), pp. 465-471.

8. Rissanen J. A Universal Prior for Integers and Estimation by Minimum Description Length // The Annals of Statistics, vol. 11 (1983), no. 2, pp. 416-431.

9. Grunwald P. D., Myung J. I., Pitt M. A. Advances in Minimum Description Length: Theory and Applications. Cambridge, Massachusetts; London, England: MIT Press, 2005. 372p.

10. Roberts S. Novelty detection using extreme value statistics // IEE Proceedings – Vision, Image and Signal Processing, vol. 146 (1999), no. 3, pp.124-129.

11. Using minimum description length to optimize the 'k' in k-medoids [Электронный ресурс]. URL: http://erikerlandson.github.io/blog/2016/08/03/x-medoids-using-minimum-description-length-to-identify-the-k-in-k-medoids (дата обращения: 10.03.2020).

ГИБИЛИНДА Роман Владимирович, ассистент учебно-научного центра «Информационная безопасность», Институт радиоэлектроники и информационных технологий - РТФ Уральский федеральный университет им. первого Президента России Б.Н. Ельцина. 620002, г. Екатеринбург, ул. Мира, 19. E-mail: r.v.gibilinda@urfu.ru

GIBILINDA Roman, assistant of Educational and Scientific Center "Information Security", Institute of Radio electronics and Information Technologies, Ural Federal University named after first President of Russia B.N. Yeltsin. 620002, Yekaterinburg, Mira str., 19. E-mail: r.v.gibilinda@urfu.ru