Артём О. Калинкин[1], Светлана А. Голуб[2], Игорь Ю. Коркин[3], Данил Н. Пятовский[4],
Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, Москва, 115409, Россия
[1]*e-mail: kalinkin8884@gmail.com, https://orcid.org/0000-0002-7785-2979,*
[2]*e-mail: glb.svtln@gmail.com, https://orcid.org/0000-0002-2395-0661,*
[3]*e-mail: igor.korkin@gmail.com, https://orcid.org/0000-0001-7640-2792,*
[4]*e-mail: danil.piat@mail.ru, https://orcid.org/0000-0002-7280-9218*

ОБНАРУЖЕНИЕ ПРОГРАММ-ШИФРОВАЛЬЩИКОВ НА ОСНОВЕ ДАННЫХ МЕХАНИЗМА ТРАССИРОВКИ СОБЫТИЙ И ПРИМЕНЕНИЯ МЕТОДА МАШИННОГО ОБУЧЕНИЯ

DOI: http://dx.doi.org/10.26583/bit.2022.3.07

Аннотация. В настоящее время многие аналитические компании отмечают значительный рост компьютерных инцидентов с использованием программ-шифровальщиков. Данный вид вредоносного программного обеспечения частично или полностью изменяет файлы пользователей, вынуждая заплатить выкуп для восстановления файлов. По результатам проведённого в работе сравнительного анализа существующих способов обнаружения программ-шифровальщиков были выявлены их недостатки и принято решение о разработке нового средства обнаружения с использованием аппарата машинного обучения. Анализ недавних атак с использованием программ- шифровальщиков позволил выявить ряд особенностей взаимодействия с файловой системой, присущих только вредоносным программам данного вида. Для сбора информации о таких событиях выбран механизм Windows Event Tracing for Windows (ETW), который предустановлен во все современные сборки Windows и имеет широкие возможности по регистрации различных событий ОС. Для выбора подходящего алгоритма машинного обучения были протестированы следующие алгоритмы: одноклассовый метод опорных векторов (One Class Support Vector Machines), алгоритм изолирующего леса (Isolation Forest) и алгоритм локального уровня выброса (Local Outlier Factor). В качестве основного был выбран алгоритм Isolation Forest. С использованием механизма ETW были получены два набора данных для легитимных программ, и для программ-шифровальщиков. Для безопасной генерации наборов данных, соответствующих программам-шифровальщикам, был разработан демонстрационный прототип ПО, который реализовал основные алгоритмы взаимодействия с файловой системой популярных программ-шифровальщиков. Полученная выборка была поделена в отношении 30% и 70% для обучающего и тестового наборов соответственно. Данные из тестовой выборки не участвовали в процессе обучения. Программная реализация системы обнаружения выполнена на языке Python. Разработанная система была успешно опробована с использованием недавних программ-шифровальщиков WannaCry и TeslaCrypt, а также популярных образцов легитимного ПО VeraCrypt, TrueCrypt, 7z, СУБД Oracle.

Ключевые слова: программы-шифровальщики, машинное обучение, поиск аномалий, Event Tracing for Windows (ETW).

Для цитирования: КАЛИНКИН, Артём О. и др. ОБНАРУЖЕНИЕ ПРОГРАММ-ШИФРОВАЛЬЩИКОВ НА ОСНОВЕ ДАННЫХ МЕХАНИЗМА ТРАССИРОВКИ СОБЫТИЙ И ПРИМЕНЕНИЯ МЕТОДА МАШИННОГО ОБУЧЕНИЯ. Безопасность информационных технологий, [S.l.], т. 29, № 3, с. 82–93, 2022. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1437. DOI: http://dx.doi.org/10.26583/bit.2022.3.07.

Artem O. Kalinkin[1], Svetlana A. Golub[2], Igor Y. Korkin[3], Danil N. Pyatovskiy[4]
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashirskoe shosse, 31, Moscow, 115409, Russia
[1]*e-mail: kalinkin8884@gmail.com, https://orcid.org/0000-0002-7785-2979,*
[2]*e-mail: glb.svtln@gmail.com, https://orcid.org/0000-0002-2395-0661,*
[3]*e-mail: igor.korkin@gmail.com, https://orcid.org/0000-0001-7640-2792,*

[4]*e-mail: danil.piat@mail.ru, https://orcid.org/0000-0002-7280-9218*

Ransomware detection based on machine learning models and Event Tracing for Windows

DOI: http://dx.doi.org/10.26583/bit.2022.3.07

Abstract. Nowadays ransomware cyberattacks are alarmingly increasing. Ransomware is a form of malicious software that locks users' files by modifying it or it's parts. To get the files back the users are supposed to pay ransom. Ransomware are using different types of cryptography, from modern symmetric ciphers to asymmetric ciphers that require the both public key and a private key. The analysis of existing ransomware detection techniques reveals some drawbacks. It was decided to develop a new ransomware detection tool based on machine learning. The analysis of recent ransomware attacks helps to find the behaviour patterns during interactions with file system, which are typical only for ransomware. To collect related OS events, the Windows built-in mechanism named Windows Event Tracing for Windows (ETW) was used. The following machine learning algorithms were checked: One Class Support Vector Machines, Isolation Forest and Local Outlier Factor. Isolation Forest algorithm shows better results. The ETW helps to gain two datasets for legitimate software programs and for ransomware apps. The whole dataset was divided into two parts, training and testing, with the training part to be around 30% of the dataset and the testing one to be 70%. The Python has been used to program the proposed ransomware detection system. The developed system was successfully tested using WannaCry and TeslaCrypt encryption programs, as well as legitimate VeraCrypt, TrueCrypt, 7z, Oracle DBMS software.

Keywords: ransomware, machine learning, anomaly detection, Event Tracing for Windows (ETW).
For citation: KALINKIN, Artem O. et al. Ransomware detection based on machine learning models and Event Tracing for Windows. IT Security (Russia), [S.l.], v. 29, no. 3, p. 82–93, 2022. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1437. DOI: http://dx.doi.org/10.26583/bit.2022.3.07.

Введение

Согласно отчёту, предоставленному SonicWall [1], в 2021 г. было зарегистрировано более, чем в два раза больше атак программ-шифровальщиков чем в 2020 г. и в три раза больше, чем в 2019 г. Можно выделить несколько факторов, способствующих такому резкому росту числа атак программ-шифровальщиков.

Во-первых, появление и активное использование децентрализованных платежных систем с собственными денежными единицами (криптовалют) обеспечили возможность нарушителям получать анонимно выкуп от пользователей, без участия посредников. В результате нарушители получили возможность прямого финансового обогащения.

Во-вторых, появление услуги, в рамках которой разработчики вредоносного ПО предоставляют другим злоумышленникам по подписке программы-шифровальщики и инфраструктуру для управления ими – Ransomware-as-a-Service (RaaS).

Программы-шифровальщики используют самые передовые вредоносные техники для предотвращения обнаружения системой защиты информации, а высокая скорость атаки программ-шифровальщиков значительно затруднят их своевременное обнаружение и удаление.

Можно заключить, что своевременное обнаружение программ-шифровальщиков является актуальной задачей информационной безопасности.

В настоящей работе используется механизм Event Tracing For Windows (ETW), позволяющий получать различные события ОС в режиме реального времени для последующей обработки. Механизм ETW имеет следующие достоинства: переносимость на различные ОС семейства Windows, широкий спектр регистрируемых событий ОС, возможность обработки информации как в режиме реального времени, так и в автономном режиме путём сохранения в файл для последующей обработки.

1. Анализ недавних атак программ-шифровальщиков

Мы провели анализ основных этапов работы программ-шифровальщиков и примеров недавних атак, для установления особенностей программ-шифровальщиков, которые могут быть использованы для их обнаружения.

1.1 Определение программ-шифровальщиков и примеры

Программа-шифровальщик нарушает целостность файла путём изменения файла целиком либо его части. На рис. 1 представлены основные этапы работы программ-шифровальщиков. Далее будет представлен анализ каждого из этапов.

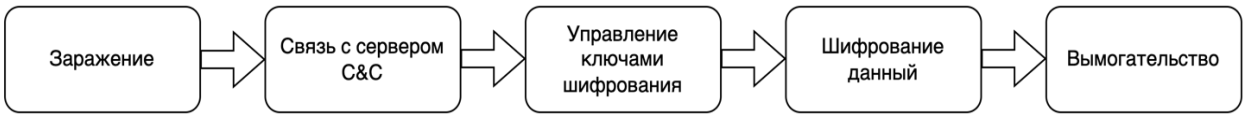

Рис. 1. Этапы атаки программ-шифровальщиков
Fig. 1. Stages of attack of cryptographic programs

I) Этап заражения. На этом этапе происходит доставка и запуск программы-шифровальщика на целевой компьютер пользователя. Нарушители используют следующие векторы атак для заражения [2]:

- рассылка спама и фишинговых писем;
- эксплуатация уязвимостей программных компонентов целевого компьютера пользователя, например веб-браузера;
- распространение вредоносного ПО через веб-сайты.

Наиболее популярным вектором заражения является распространение вредоносного ПО через веб-сайты.

II) Этап связи с управляющим сервером. После этапа заражения программа-шифровальщик обычно связывается через сеть Интернет с командным сервером, называемым Command and Control, C&C, C2 или кластером серверов для передачи и хранения ключей шифрования, а также информации о пользователе и заражённой системе.

Связь с сервером может быть организована с использованием следующих способов:

- статически настроенных IP-адресов;
- статических DNS-адресов (имён);
- динамически создаваемых DNS-адресов (имён).

Некоторые группы программ-шифровальщиков осуществляют работу без связи с сервером, поскольку открытый ключ из ассиметричной пары включен в двоичный файл программы-шифровальщика. Наиболее популярным способом связи является использованием статических DNS-адресов

III) Управление ключами шифрования. На данном этапе происходит генерация криптографических ключей для шифрования файлов пользователя. Криптографические ключи могут быть сгенерированы и сохранены на стороне управляющего сервера, так и локально на стороне программы-шифровальщика.

Для шифрования данных могут использоваться следующие сценарии:

- симметричного шифрования;
- асимметричного шифрования;
- гибридного шифрования, при котором содержимое файлов шифруется с использованием симметричного ключа, затем этот ключ шифруется открытым ключом из ассиметричной пары.

Программы-шифровальщики наиболее часто используют асимметричное и гибридное шифрование.

IV) Шифрование данных. Данный этап является основным для всех программ-шифровальщиков и заключается в блокировке оригинальных документов путём перезаписи их содержимого зашифрованными фрагментами.

Для ускорения этого этапа и сокращения использования ресурсов центрального процессора отдельные примеры программ-шифровальщиков изменяют фиксированное число байт в каждом файле, без изменения оставшейся части файла.

Рабочие документы пользователей, созданные с использованием пакета программ Microsoft Office уязвимы к таким атакам. Соответствующие файловые форматы docx и pptx включают в себя так называемые заголовки, обычно занимающие первые байты файла и хранящие информацию о бинарной структуре файла. Изменение такого заголовка блокирует доступ ко всему файлу, поскольку соответствующие программы-редакторы пакета программ Microsoft Office не могут работать с повреждёнными файлами. В результате пользователь теряет возможность читать и вносить изменения в рабочие файлы. Программы-шифровальщики также могут менять имя файла и изменять его расширение.

V) Этап вымогательства.

На финальном этапе программа-шифровальщик сообщает о выполненных вредоносных манипуляциях в системе и заявляет о требованиях, которые необходимо выполнить пользователю для восстановления изменённых файлов. Как правило, в требованиях указан перевод денежных средств на счета нарушителей с использованием децентрализованных платёжных систем.

Требования нарушителей могут быть предоставлены одним из следующих способов:

- текстовые файлы;
- HTML-документы;
- графические файлы различных форматов.

В ряде случаев программы-шифровальщики создают файлы с требованиями о выкупе в каждом каталоге с зашифрованными файлами [2].

1.1.1 Результаты анализа недавних атак программ-шифровальщиков

В ходе анализа популярных программ-шифровальщиков [3–7], таких как WannaCry, Locky, Ryuk, GandCrab, Coronavirus, Hive были выявлены их основные особенности приведенные в табл. 1.

Таблица 1. Результаты сравнения популярных программ-шифровальщиков

Название, год	**Операционная система**	**Тип заражения**	**Используемый алгоритм шифрования**
WannaCry, 2017	Windows	С помощью уязвимости EternalBlue	AES и RSA
Locky, 2017	Windows	Фишинг	AES
Ryuk, 2018	Кроссплатформенный	С помощью протокола удалённого доступа	AES и RSA
GandCrab, 2018	Кроссплатформенный	Фишинг	AES
CoronaVirus, 2020	Windows	Фишинг	AES и RSA
Hive, 2021	Кроссплатформенный	Фишинг	RSA

1.1.2 Особенность работы программ-шифровальщиков

В ходе анализа популярных программ-шифровальщиков была выявлена следующая особенность их работы. На этапе шифрования программа-шифровальщик совершает значительно большее число файловых операций, чем остальные работающие процессы в ОС за одинаковый промежуток времени. На рис. 2 продемонстрировано, что программа-шифровальщик под условным номером 1 совершает 2954 файловых операций, в то время как остальные легитимные программы генерируют не более 1300 файловых операций. Отмеченный рост файловых операций может быть использован для обнаружения программ-шифровальщиков.

Рис. 2. Количество файловых операций, генерируемых работающими процессами
Fig. 2. The number of files actions generated by running processes

1.2 Индикаторы обнаружения программ-шифровальщиков

Помимо отмеченного роста файловых операций, программы-шифровальщики также могут быть обнаружены с учётом их следующих признаков:

- увеличение числа вновь созданных файлов;
- увеличение числа удаленных файлов;
- увеличение числа переименованных файлов, а также файлов с изменённым расширением;
- увеличение операций чтения и записи для файлов, находящихся в одной папке;
- значительное замедление работы ОС;
- изменение обоев рабочего стола устройства;
- появление на экране явного уведомления о выкупе.

В данной работе использованы индикаторы, связанные с взаимодействием с файловой системой. Также можно выделить четыре общих особенности, которые характеризуют поведение большинства семейств программ-шифровальщиков, табл. 2.

Таблица 2. Примеры взаимодействия с файлами для различных программ-шифровальщиков

Название программ-шифровальщиков	**Примеры взаимодействия с файлами для программ-шифровальщиков**
Cerber, Keypass, Telsacrypt, Gandcrab	Преобразование файла с последующим изменением его имени: файл сначала перезаписывается зашифрованными данными пользователя, затем изменяется имя файла.
Locky	Преобразование файла с предварительным изменением его имени: эта особенность аналогична предыдущей, за исключением того, что сначала файл переименовывается, а затем перезаписывается.
InfinityCrypt, Dharma, Malevich, Sage, Syrk, WannaCry	Копирование исходного файла в новый файл с удалением старого: сначала создаётся новый файл и в него копируются зашифрованные данные исходного файла, после исходный файл удаляется.

1.3 Сравнительный анализ существующих способов обнаружения программ-шифровальщиков

На рис. 3 представлена систематизация существующих способов обнаружения программ-шифровальщиков.

Рис. 3. Систематизация способов обнаружения программ-шифровальщиков
Fig. 3. Systematization of the processes of detecting encryption programs

Способы обнаружения могут быть условно разделены на две группы: с использованием статического анализа и динамического анализа. С помощью статического анализа можно обнаружить программы-шифровальщиков с использованием файловых сигнатур. Динамический анализ позволяет извлекать и обрабатывать информацию о работающей программе [8].

Программы-шифровальщики могут использовать различные приёмы для противодействия своему обнаружению. Например, с помощью техник обфускации программа-шифровальщик может скрыться от сигнатурных средств обнаружения [9].

Обнаружение с помощью динамического анализа лишены таких недостатков и является более перспективным. Способы обнаружения вредоносных программ на основе машинного обучения принимают решение о наличии программ-шифровальщиков с помощью моделей, которые построены с использованием набора аналитических функций. Системы обнаружения программ-шифровальщиков на основе машинного обучения используют структурные особенности программ и их поведенческие особенности.

В данной работе был выбран способ обнаружения на основе машинного обучения с использованием поведенческих особенностей, связанных с взаимодействием программ с файловой системой. Были выбраны следующие индикаторы обнаружения, основанные на увеличении таких файловых манипуляций, как удаление файла, переименование файла, создание нового файла, запись в файл.

2. Обнаружение программ-шифровальщиков с помощью механизма логирования событий ETW

Для сбора информации о файловых операциях в режиме реального времени в ОС Windows можно использовать ряд инструментов:

- Драйвер-фильтр файловой системы.
- Гипервизор на базе аппаратной виртуализации.
- Механизм трассировки событий Windows Event Tracing for Windows (ETW).

Драйвер-фильтр файловой системы добавляется в стек драйверов и получает уведомления о каждом файловом событии. В недавних ОС Windows для установки сторонних драйверов необходима цифровая подпись, что не всегда удобно.

Использование возможностей гипервизоров требует наличия поддержки аппаратной виртуализации, что доступно только в современных компьютерных системах.

Механизм трассировки событий Windows Event Tracing for Windows (ETW) исключает указанные недостатки: не требует установки драйверов, переносим на различные ОС семейства Windows и поддерживает широкий спектр регистрируемых событий ОС [10]. В работе был выбран механизм ETW для сбора информации о файловых событиях в режиме реального времени.

2.1 Анализ возможностей ETW

Механизм трассировки событий Windows Event Tracing for Windows (ETW) был добавлен во все версии ОС Windows, начиная в Windows 2000. ETW был изначально реализован для выявления ошибок и «узких мест» в программе. В настоящее время ETW используется различными средствами защиты информации для получения информации о событиях ОС.

Механизм ETW может регистрировать различные события ОС, такие как: выполненные системные вызовы, операции доступа к реестру Windows и файловой системе, данные о сетевом взаимодействии.

ETW состоит из четырёх компонентов:

- ETW-сессия – это объект ОС создаваемый ETW-контроллером для сбора заданных событий ОС – создание, настройку, запуск и остановку ETW-сессии;
- поставщики событий – это прикладные программы и драйверы, которые создают ETW-события и пересылают информацию потребителям событий;
- потребители событий – это прикладные программы и драйверы, которые получают сгенерированные события.

Например, программа Process Monitor из пакета программ Windows Sysinternals получает события о взаимодействии программ с сетью с использованием ETW. В работе события регистрируются в режиме реального времени [11].

3. Разработка системы обнаружения программ-шифровальщиков

Разработанная система обнаружения использует информацию о новых событиях ОС от механизма ETW и с помощью обработки собранной информации методами машинного обучения принимает решение о присутствии программ-шифровальщиков. Алгоритм работы программы представлен на рис. 4.

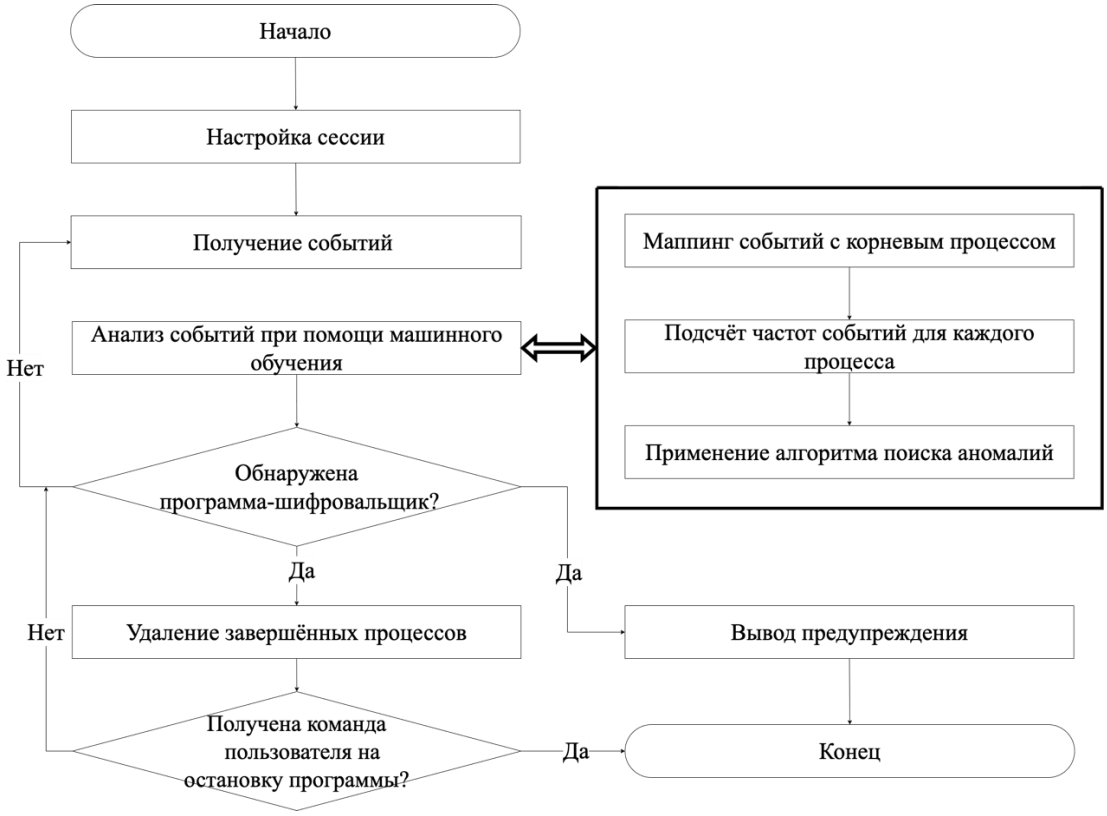

Рис. 4. Алгоритм работы программы
Fig. 4. The algorithm of the program

Для получения информации о файловых событиях были использованы возможности встроенного поставщика событий Microsoft-Windows-Kernel-File. Соответствующая сессия была запущена с использованием библиотеки PyWinTrace, которая предоставляет интерфейс для настройки и управления ETW-сессиями. Отдельные регистрируемые события сохраняются в формате json в файле журнала на жёстком диске, который можно преобразовать в формат csv для удобства дальнейшей обработки массивов данных (выборкой).

3.1 Разработка ПО

Программная реализация разработанного алгоритма была выполнена в среде разработки PyCharm для Windows 10 на языке Python 3 [12].

Для получения обучающей выборки из событий ПО необходимо собрать файловые события, генерируемые программами-шифровальщиками. Для того, чтобы безопасно получить такие события было разработано ПО, реализующее алгоритмы действий с файловой системой, основанное на популярных программах-шифровальщиках. Разработанное демонстрационное ПО осуществляло шифрование файлов только в одной из заданных папок, например, «D:\test». Были реализованы следующие наборы событий, наиболее часто встречающиеся в программах-шифровальщиках:

- переименование исходного файла, запись зашифрованных данных в исходный файл;
- запись зашифрованных данных в исходный файл (без переименования исходного файла);
- создание нового файла, запись зашифрованных данных в новый файл, удаление исходного файла.

3.2 Создание набора данных для классификатора

Для создания набора данных (датасета) последовательно запускались популярные программы-шифровальщики и сборщик событий, в результате работы был получен журнал событий в формате .json. Вручную были помечены ID процессов (pid) для программ-шифровальщиков единицей, для остальных легитимных процессов – нулём. Эти действия повторяются для всех наборов программ-шифровальщиков. Каждая программа-шифровальщик запускалась 16 раз. Фрагмент обучающей выборки представлен на рис. 5.

	pid	label	OPERATIONEND	24	CLOSE	14	CREATE	12	QUERYSECURITY	32	FSCTL	23	QUERYINFORMATION
0	6848 - 0	1.0	1551.0	1551.0	174.0	174.0	154.0	154.0	76.0	76.0	152.0	152.0	343.0
1	4796 - 0	0.0	462.0	462.0	67.0	67.0	67.0	67.0	0.0	0.0	105.0	105.0	150.0
2	27068 - 0	0.0	7.0	7.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0	1.0
3	22156 - 0	0.0	107.0	107.0	0.0	0.0	8.0	8.0	4.0	4.0	0.0	0.0	40.0
4	20196 - 0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	4 - 0	0.0	550.0	550.0	527.0	527.0	1.0	1.0	0.0	0.0	30.0	30.0	0.0
6	3620 - 0	0.0	249.0	249.0	39.0	39.0	51.0	51.0	8.0	8.0	4.0	4.0	79.0
7	9712 - 0	0.0	243.0	243.0	37.0	37.0	43.0	43.0	8.0	8.0	0.0	0.0	34.0
8	23576 - 0	0.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
9	16392 - 0	0.0	86.0	86.0	17.0	17.0	17.0	17.0	0.0	0.0	0.0	0.0	35.0
10	2180 - 0	0.0	6.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Рис. 5. Фрагмент обучающей выборки
Fig. 5. Fragment of the training sample

Параметры, необходимые для составления обучающей выборки:

- pid – id процесса;
- 0, 1, 2, … ,15 – номер журнала событий в формате .json;
- label – метка (1– программа-шифровальщик, 0 – обычный процесс);
- данные о файловой активности, полученные от ETW: operationend, close, create, querysecurity, queryinformation, cleanup, read, dirnotify, createnewfile, write, namecreate, setinformation, rename, renamepath, namedelet, setdelet, deletpath, direnum, queryea, flush, setsecurity.

3.3 Обучение классификатора

Для обучения и тестирования использовалось следующее разделение: обучающая выборка (30%), тестовая выборка (70%).

Для обучения модели, было принято решение использовать алгоритмы обнаружения аномалий, что позволило решить задачу классификации на выборках с крайне неравномерным распределением классов. Для обучения классификатора были тестированы следующие алгоритмы:

- OneclassSVM (Метод опорных векторов для одного класса) [13];
- IsolationForest (Изолирующий лес) [14];
- LocalOutlierFactor (Локальный фактор выбросов) [15].

Каждый из алгоритмов запускался по 10 раз, в табл. 3 представлены наилучшие результаты каждого из алгоритмов.

В ходе тестов лучше всего себя показал алгоритм Isolation Forest. Далее эта модель была использована в работе в качестве основной.

Таблица 3. Результаты тестирования алгоритмов машинного обучения

Алгоритмы машинного обучения	**Процессы программ-шифровальщиков (12 процессов)**		**Легитимные процессы (119 процессов)**	
	Precision	Recall	Precision	Recall
One Class Support Vector Machines (одноклассовый метод опорных векторов)	0,31	0,83	0,98	0,82
Isolation Forest (алгоритм изолирующего леса)	1	1	1	1
Local Outlier Factor (алгоритм локального уровня выброса)	0,11	0,17	0,91	0,86

3.4 Тестирование системы обнаружения программ-шифровальщиков

Тестирование происходило в два этапа:

- тестирование с целью проверки обнаружения существующих программ-шифровальщиков;
- тестирование с целью проверки отсутствия ложноположительных срабатываний на легитимных программах.

При обнаружении программы-шифровальщика выводится предупреждение, которое представлено на рис. 6.

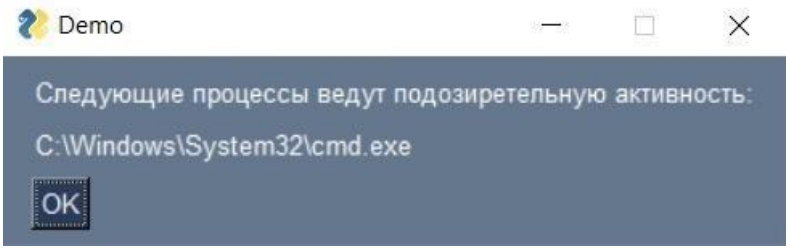

Рис. 6. Вывод предупреждения
Fig. 6. Warning output

В ходе тестирования были успешно обнаружены следующие известные программы-шифровальщики: Wannacry и TeslaCrypt.

Число файлов, зашифрованных программой-шифровальщиком WannaCry до момента обнаружения, составляет 15 файлов. TeslaCrypt до момента обнаружения успевает зашифровать 10 файлов пользователя. Эти результаты являются приемлемыми, так как уменьшение порогового значения числа зашифрованных файлов приведёт к увеличению ложноположительных срабатываний.

Для тестирования с целью проверки отсутствия ложноположительных срабатываний использовалось следующее ПО:

- программы для создания виртуальных дисков и шифрования на «лету»: TrueCrypt и VeraCrypt;
- программные пакеты Microsoft Office;
- система управления реляционными базами данных Oracle;
- файловый архиватор 7z;
- встроенное в ОС Windows средство для дефрагментации диска.

В результате тестирования ложноположительных срабатываний не выявлено.

Все тесты проводились, для операционной системы Windows 10 x64. Также возможна адаптация разработанного средства обнаружения программ-шифровальщиков на Windows 10 x32, Windows 11 x64/x32 и Windows 7 x64/x32.

Заключение

В работе был выполнен анализ популярных программ-шифровальщиков, проведён сравнительный анализ существующих способов обнаружения программ-шифровальщиков. Выявлена особенность работы программ-шифровальщиков, заключающая в резком росте файловых операций, таких как создание, переименование, перезапись и удаление, что позволило разработать систему обнаружения.

Для сбора информации о работающих процессах и их файловых операциях был использован механизм ETW. Собранные данные обрабатывались методами машинного обучения. Разработано ПО, реализующее алгоритмы взаимодействия с файловой системой, основанные на популярных программах-шифровальщиках.

При обучении классификатора решалась задача поиска аномалий. В ходе тестирования алгоритм Isolation Forest (Изолирующий лес) показал высокую точность обнаружения и малое количество ложноположительных срабатываний относительно остальных алгоритмов, и был взят за основу при разработке средства обнаружения программ-шифровальщиков.

В ходе тестирования разработанного средства были успешно обнаружены следующие образцы программ-шифровальщиков: Wannacry и TeslaCrypt. В ходе тестирования ложноположительных срабатываний выявлено не было.

СПИСОК ЛИТЕРАТУРЫ:

1. CYBER THREAT REPORT, 2022. URL: https://www.infopoint-security.de/media/2022-sonicwall-cyber-threat-report.pdf (дата обращения: 20.02.2022).
2. A Survey on Detection Techniques for Cryptographic Ransomware, 2019. URL: http://dataset.tlm.unavarra.es/ransomware/articles/IEEEAccess.pdf (дата обращения: 20.02.2022).
3. Analyzing WannaCry Ransomware Considering the Weapons and Exploits, 2022. URL: https://icact.org/upload/2018/0708/20180708_finalpaper.pdf (дата обращения: 20.02.2022).
4. Threat spotlight: the curious case of Ryuk ransomware, 2019. URL: https://blog.malwarebytes.com/threat-spotlight/2019/12/threat-spotlight-the-curious-case-of-ryuk-ransomware/ (дата обращения: 20.02.2022).
5. RYUK RANSOMWARE, 2021. URL: https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-006.pdf (дата обращения: 20.02.2022).
6. Digital CoronaVirus: Yet Another Ransomware Combined with Infostealer, 2020. URL: https://www.acronis.com/en-us/blog/posts/digital-coronavirus-yet-another-ransomware-combined-infostealer/ (дата обращения: 20.02.2022).
7. Hive Attacks. Analysis of the Human-Operated Ransomware Targeting Healthcare, 2021. URL: https://www.sentinelone.com/labs/hive-attacks-analysis-of-the-human-operated-ransomware-targeting-healthcare/ (дата обращения: 20.02.2022).
8. A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions, 2022. URL: https://arxiv.org/pdf/2102.06249.pdf (дата обращения: 20.02.2022).
9. Advances In Malware Detection-An Overview, 2021. URL: https://arxiv.org/pdf/2104.01835.pdf (дата обращения: 20.02.2022).
10. Peeler: Profiling Kernel-Level Events to Detect Ransomware, 2020. URL: https://ndss21-summer.hotcrp.com/doc/ndss21-summer-paper121.pdf?cap=0121aPNQ0EBTM4to (дата обращения: 12.12.2020).
11. A Comparison of Malware Detection Techniques Based on Hidden Markov Model, 2016. URL: https://www.scirp.org/pdf/JIS_2016042209291406.pdf (дата обращения: 20.02.2022)
12. Kalinkin A. Ransomware detection, 2022. URL: https://github.com/kalinkinartem1/Ransomware-detection (дата обращения: 20.02.2022).

13. Introduction to One-class Support Vector Machines, 2013. URL: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/ (дата обращения: 20.02.2022).
14. Anomaly detection using Isolation Forest – A Complete Guide, 2021. URL: https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/ (дата обращения: 20.02.2022).
15. Anomaly Detection Techniques in Python, 2019. URL: https://medium.com/learningdatascience/anomaly-detection-techniques-in-python-50f650c75aaf (дата обращения: 20.02.2022).

REFERENCES:

[1] CYBER THREAT REPORT, 2022. URL: https://www.infopoint-security.de/media/2022-sonicwall-cyber-threat-report.pdf (accessed: 20.02.2022).
[2] A Survey on Detection Techniques for Cryptographic Ransomware, 2019. URL: http://dataset.tlm.unavarra.es/ransomware/articles/IEEEAccess.pdf (accessed: 20.02.2022).
[3] Analyzing WannaCry Ransomware Considering the Weapons and Exploits, 2022. URL: https://icact.org/upload/2018/0708/20180708_finalpaper.pdf (accessed: 20.02.2022).
[4] Threat spotlight: the curious case of Ryuk ransomware, 2019. URL: https://blog.malwarebytes.com/threat-spotlight/2019/12/threat-spotlight-the-curious-case-of-ryuk-ransomware/ (accessed: 20.02.2022).
[5] RYUK RANSOMWARE, 2021. URL: https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-006.pdf (accessed: 20.02.2022).
[6] Digital CoronaVirus: Yet Another Ransomware Combined with Infostealer, 2020. URL: https://www.acronis.com/en-us/blog/posts/digital-coronavirus-yet-another-ransomware-combined-infostealer/ (accessed: 20.02.2022).
[7] Hive Attacks. Analysis of the Human-Operated Ransomware Targeting Healthcare, 2021. URL: https://www.sentinelone.com/labs/hive-attacks-analysis-of-the-human-operated-ransomware-targeting-healthcare/ (accessed: 20.02.2022).
[8] A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions, 2022. URL: https://arxiv.org/pdf/2102.06249.pdf (accessed: 20.02.2022).
[9] Advances In Malware Detection-An Overview, 2021. URL: https://arxiv.org/pdf/2104.01835.pdf (accessed: 20.02.2022).
[10] Peeler: Profiling Kernel-Level Events to Detect Ransomware, 2020. URL: Режим доступа: https://ndss21-summer.hotcrp.com/doc/ndss21-summer-paper121.pdf?cap=0121aPNQ0EBTM4to (accessed: 12.12.2020).
[11] A Comparison of Malware Detection Techniques Based on Hidden Markov Model, 2016. URL: https://www.scirp.org/pdf/JIS_2016042209291406.pdf (accessed: 20.02.2022).
[12] Kalinkin A. Ransomware detection, 2022. URL: https://github.com/kalinkinartem1/Ransomware-detection (accessed: 20.02.2022).
[13] Introduction to One-class Support Vector Machines, 2013. URL: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/ (accessed: 20.02.2022).
[14] Anomaly detection using Isolation Forest – A Complete Guide, 2021. URL: https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/ (accessed: 20.02.2022).
[15] Anomaly Detection Techniques in Python, 2019. URL: https://medium.com/learningdatascience/anomaly-detection-techniques-in-python-50f650c75aaf (accessed: 20.02.2022).

Поступила в редакцию – 17 июля 2022 г. Окончательный вариант – 01 сентября 2022 г.
Received – July 17, 2022. The final version – September 01, 2022.