КОМПЬЮТЕРНОЕ КРИМИНАЛИСТИЧЕСКОЕ ИССЛЕДОВАНИЕ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Белей А.В.[1], Томская С.И.[2]

Ключевые слова: *вредоносное программное обеспечение, специализированное программное обеспечение, эксперт, криминалистическое исследование, судебная компьютерно-техническая экспертиза, активность, эффективность, виртуальная среда, виртуальная машина, способ, эксперимент, криминалистически значимая информация.*

Аннотация

Цель: анализ эффективности компьютерного криминалистического исследования вредоносного программного обеспечения (ВПО) при помощи средств виртуализации и специализированного программного обеспечения.

Методы: моделирование среды операционной системы путем современных средств виртуализации, проведение имитационного эксперимента посредством помещения ВПО в виртуальную среду, индукция полученных в ходе эксперимента знаний для обобщения выводов и оценки эффективности компьютерного криминалистического исследования.

Результаты: разработаны и экспериментально обоснованы методические рекомендации по эффективному практическому применению специализированных программных средств при криминалистическом исследовании потенциально вредоносных исполняемых файлов; установлены недочеты в правоприменительной практике относительно подобного рода исследований, которые заключаются в невозможности использования рассмотренного способа компьютерного анализа в случае применения разработчиком ВПО техник обхода исполнения вредоносного кода в виртуальной среде.

Использование методических рекомендаций способствует повышению оперативности работы судебного эксперта над установлением криминалистически значимой информации и вопросами, поставленными перед ним относительно функционала и назначения ВПО.

DOI: 10.21681/1994-1404-2023-2-102-112

В настоящее время существует огромное количество видов вредоносного программного обеспечения (ВПО) [6]. По данным[3] компании SonicWall Capture Labs, каждую неделю более 18 млн веб-сайтов заражаются вредоносным кодом, 34% предприятий подвергаются вредоносным атакам, а 80% финансовых учреждений ежегодно становятся мишенью для вредоносных программ. По данным компании «КРОК», общее число кибератак за последний год увеличилось на 23%. При этом отмечается рост целенаправленных кибератак, бóльшая часть которых происходит с использованием ВПО[4]. Пагубное воздействие ВПО проявляется в том, что оно подрывает принцип надёжности устройства, нарушает неприкосновенность и конфиденциальность личной жизни, разрывает отношения между защищёнными механизмами работы компьютера посредством комбинаций несанкционированных действий и др.

Почти во всех странах приняты законы, которые запрещают создание и распространение компьютерных вирусов и прочих типов вредоносных программ. Но для того, чтобы законоисполнителю дать уголовно-правовую оценку и подтвердить принадлежность программы к классу вредоносных, необходимо назначить судебную компьютерно-техническую экспертизу, которая требует специальных навыков и знаний[5]. Для достижения данных целей перед судебным компьютерно-техническим экспертом должен быть поставлен вопрос о функциях ВПО, в частности, предназначенных для уничтожения, блокирования, модификации, копи-

[3] URL: http://www.sonicwall.com/2023-cyber-threat-report

[4] Ежегодные отчеты МВД РФ. URL: https://мвд.рф /reports/item/22678184 (дата обращения: 20.03.23).

[5] Банк данных угроз безопасности информации ФСТЭК России. URL: http://www.bdu.fstec.ru (дата обращения: 20.03.23).

[1] ***Белей Артём Вячеславович,*** *ассистент кафедры безопасности в цифровом мире Московского государственного университета имени Н.Э. Баумана, аспирант кафедры судебных экспертиз и криминалистики Российского государственного университета правосудия, г. Москва, Российская Федерация.*

E-mail: belyyskrat@yandex.ru

[2] ***Томская Стана Игоревна,*** *студентка кафедры безопасности в цифровом мире Московского государственного университета имени Н. Э. Баумана, г. Москва, Российская Федерация.*

E-mail: miss.stana@mail.ru

рования компьютерной информации или нейтрализации средств защиты компьютерной информации [3].

В рамках экспертизы с помощью специализированного программного обеспечения (СПО) осуществляется обнаружение, изъятие и фиксирование действий вредоносной программы. Обычно анализу подлежит не только загрузочный файл, но и метаданные файла, его содержание, поведение и сетевое взаимодействие [5].

В случае с исследованием ВПО эксперты часто сталкиваются с методическими сложностями, отсутствием полного объёма информации или недостаточным количеством теоретических знаний в сфере программирования или обратной разработки. От специализации эксперта во многом зависит, какой способ будет рациональным при анализе ВПО и насколько применение этого способа будет эффективным [2, 7, 8].

Несмотря на то, что многие исследователи акцентируют внимание на обнаружении и последующем анализе ВПО посредством анализа дампа оперативной памяти, такой способ не всегда эффективен [1, 4]. Возможность снять дамп оперативной памяти представляется крайне редко, к тому же существуют и другие, более эффективные способы анализа ВПО. У эксперта есть возможность исследовать исполняемый файл или библиотеку посредством использования средств обратной разработки, однако в современных реалиях не все эксперты обладают должным объёмом знаний для осуществления подобного рода анализа [12]. Такой способ требует опыта в программировании и использовании специфичных программных инструментов.

Помимо указанного, есть способ, который позволяет сделать оперативный вывод о функциональном назначении ВПО, его можно сравнить со способом эксперимента. Суть этого способа заключается в использовании средств виртуализации для моделирования программного окружения в виде операционной системы, настройке мониторинга посредством СПО и последующего запуска потенциально вредоносного файла в виртуальной среде. Данный способ позволяет эксперту исследовать потенциально вредоносный файл в изолированной среде и сделать оперативные выводы о функциях, выполняемых исполняемым файлом или библиотекой [12]. Такой способ аналогичен применению средств защиты компьютерных сетей или почтовых серверов, когда файл сначала запускается в изолированной среде — «песочнице», а уже потом становится доступным для взаимодействия у пользователя, если проверка прошла успешно [10].

Вместе с тем следует отметить несколько условий, которые обязательно должны быть соблюдены. *Во-первых*, образ операционной системы (ОС), устанавливаемый на виртуальную машину, должен быть идентичным ОС, которая стояла на персональном компьютере жертвы вредоносной активности. Связано это с тем, что некоторое ВПО использует либо уязвимости в определенной версии ОС, либо написано под определенные системы и выполняется только в их среде. *Во-вторых*, некоторое грамотно скомпилированное ВПО может обходить выполнение кода посредством анализа среды, в которой пытаются осуществить запуск, что бывает редко, но не исключено. Это может быть мониторинг пользовательской активности, исследование названий определенных директорий, попытки запросов к информации о системе или сети. Часто достаточно просто переименовать ряд директорий, используя другие названия, отличные от сервиса, и использовать виртуальную машину, где уже проводилась пользовательская активность. Если это не помогает и программа не запускается, то путем обратной разработки можно найти участок кода, который отвечает за анализ окружения и удалить его, после чего программа запустится. В данном случае вносится изменение в код программы, однако такой способ, хоть и разрушающий, используется с копией файла с образа диска, и он просто способствует ускорению и повышению эффективности анализа.

Основной задачей данной работы является изучение универсального способа исследования ВПО, который по причине *доступности, экономичности* и *оперативности* анализа позволит успешно сформулировать первичные выводы о функционале ВПО, тем самым ускорив анализ артефактов и повысив эффективность работы эксперта.

Эффективность данного способа анализа и поиска цифровых следов работы ВПО была рассмотрена на примере применения средства виртуализации *VirtualBox* с ОС *Windows* 10 и *Kali-Linux*.

Перед исследованием была произведена установка виртуальной машины в *VirtualBox*, созданы образы ОС *Kali-Linux* и *Windows* 10. На виртуальной машине *Kali-Linux* работали различные сетевые службы, а на виртуальной машине *Windows* было проведено исследование ВПО. После настройки виртуальных машин была создана изолированная сеть, которая связала две виртуальные машины. Для этого в настройках сетевого интерфейса виртуальной машины *Windows* 10 в разделе *Internet Protocol* был настроен шлюз и *DNS*-сервер, чтобы они соответствовали *IP*-адресу ОС *Kali-Linux*.

Настройка интерфейса для *Windows* 10: *IP*-адрес 10.0.2.15, маска (*subnet mask*) 255.255.255.0, основной шлюз (*default gateway*) 10.0.2.14, предпочитаемый *DNS*-сервер (*alternate DNS server*) 10.0.2.14.

После была открыта виртуальная машина *Kali-Linux*. Осуществлена проверка соответствия основного шлюза и *DNS*-сервера с *IP*-адресом ОС *Kali-Linux* с помощью команды ifconfig. Данная команда используется для настройки сети в операционных системах *Linux*. После чего был открыт файл конфигурации *Inetsim*, который находится по адресу: /etc/inetsim/inetsim.conf. По умолчанию *Inetsim* прослушивает только локальный хост, поэтому его необходимо сделать доступным для всех машин виртуальной сети. Для проверки была запущена программа *Inetsim* с помощью команды sudo inetsim и проведено обращение к адресу *yandex.ru*.

В виртуальной среде ОС *Windows* 10 были установлены программы, необходимые для анализа ВПО, такие как *PeStudio, Filealyzer, AutoRuns, Process Monitor*, *Reg-*

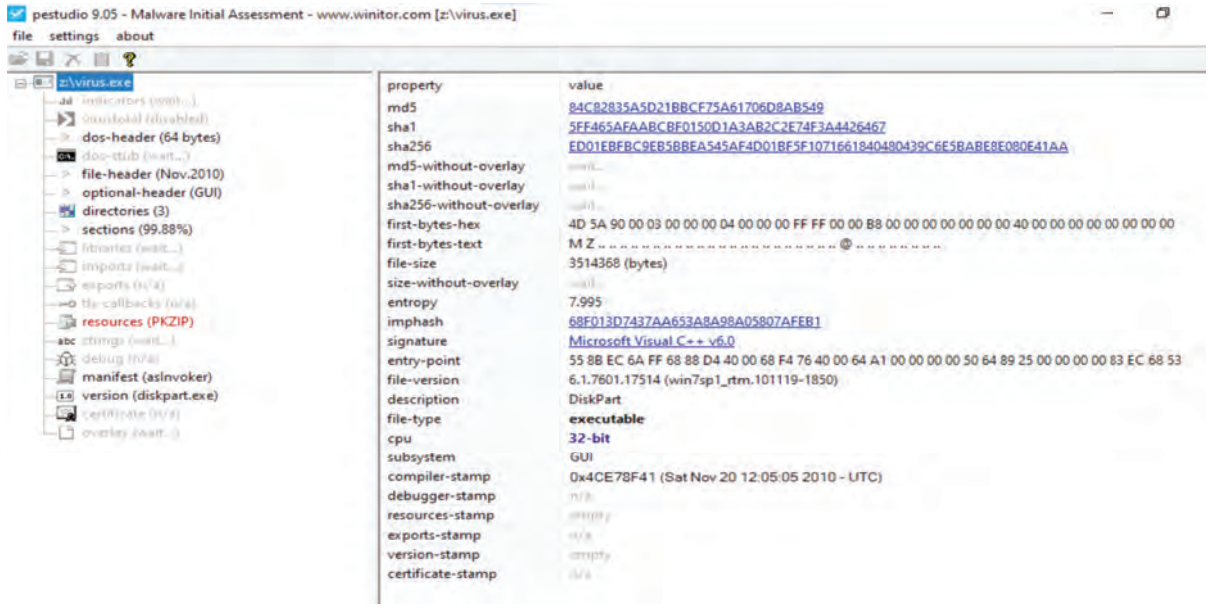

Рис. 1. Окно программы PeStudio, содержащее информацию о метаданных файла VIRUS.exe

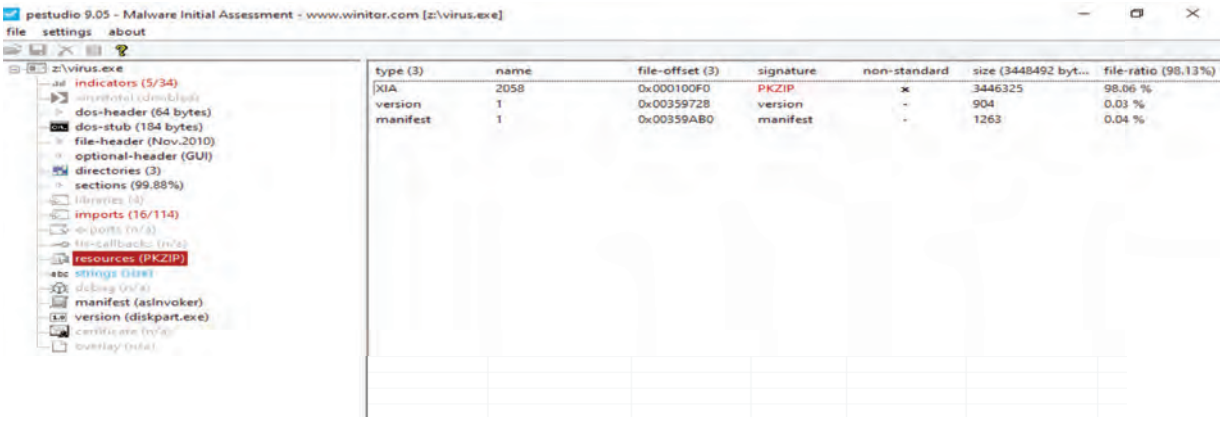

Рис. 2. Окно программы PeStudio, содержащее информацию о списке внутренних ресурсов исследуемого файла VIRUS.exe

Shot. На виртуальную ОС *Kali-Linux* была установлена программа *WireShark*. Далее был загружен файл с сайта *GitHub* под названием *VIRUS.exe*. Данный файл был перемещен в папку виртуальной машины *Windows* 10.

При исследовании в первую очередь необходимо провести анализ метаданных файла, не запуская его. Чтобы определить, упакован файл или нет, а также найти все возможные артефакты, используем программу *PeStudio*.

Среди артефактов следует обратить внимание на временны́е метки компиляции файла, загружаемые библиотеки, используемые ресурсы, информацию о версии исполняемого файла, характерные строки, а также отладочную информацию и файл сборки (*Manifest*). С помощью программы *PeStudio* были определены метаданные файла *VIRUS.exe* (*рис. 1—3*).

При анализе метаданных файла были определены: временны́е метки компиляции файла, загружаемые библиотеки, используемые ресурсы, информация о версии исполняемого файла (6.1.7601.17514 win7sp1_rtm.101119-1850), тип файла, а также отладочная информация и файл сборки (*Manifest*) [11]. При анализе содержания файла в нем был найден еще один упакованный и зашифрованный файл с сигнатурой *PKZIP*.

Далее для анализа структуры файла следует воспользоваться программой *Filealyzer*. При анализе метаданных файла были определены: размер файла, его расположение, информация о версии исполняемого файла (6.1.7601.17514), внутреннее имя файла (*diskpart.exe*), время создания и последнего запуска файла, *CRC*-32 значение, содержимое *PE*-секций, таблицы импорта/экспорта и список ресурсов, среди которых тип ресурса *XIA* — это данные, зашифрованные *Chiasmus*. По заголовку в 16-ричном дампе исследуемого файла можно определить, что файл имеет расширение *.exe,* так как сигнатура файла "4D 5A" относится к исполняемым файлам формата *DOS MZ executable* (*рис. 4, 5*). С помощью данной программы была также определена контроль-

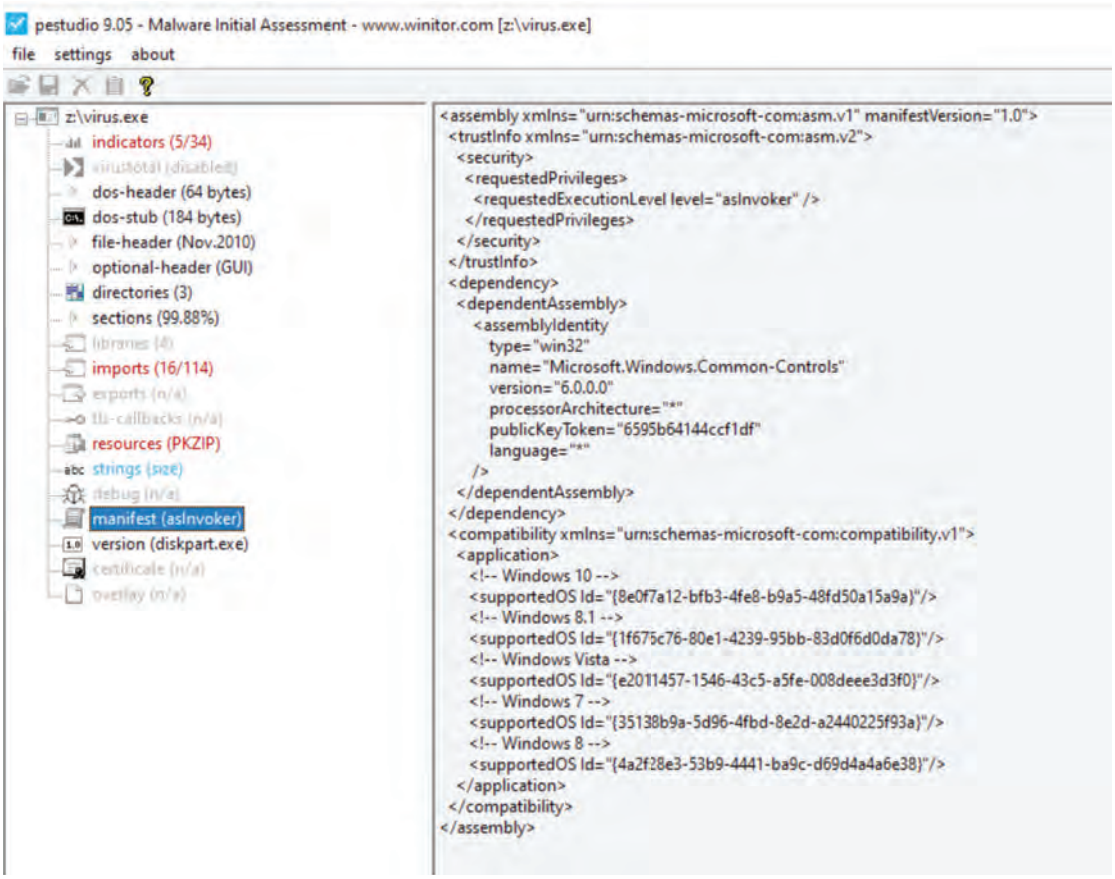

Рис. 3. Окно программы PeStudio, содержащее информацию о манифесте исследуемого файла VIRUS.exe

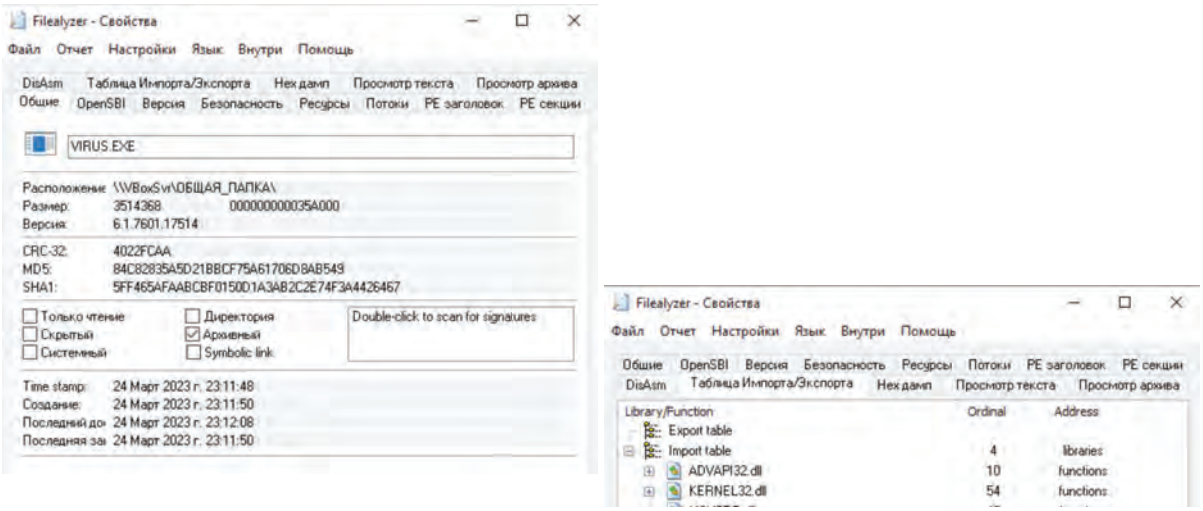

Рис. 4. Фрагменты окон программы Filealyzer, содержащих информацию о хэш-сумме исследуемого файла VIRUS.exe и содержимое таблицы импорта/экспорта

ная сумма (хэш-сумма) файла, что позволяет проверить факт того, что файл остался неизменным.

Далее следует обратить внимание на реестр *Windows,* с которым очень часто взаимодействует ВПО. Наиболее известной программой для отслеживания изменений в реестре является *Regshot*. Она позволяет сделать снимки до запуска ВПО и после сравнить две версии системного реестра. Надо заметить, что *Regshot* не позволяет сканировать определенные разделы и ключи, из-за чего в файл отчета записываются изменения, сделанные самой *Windows*.

После запуска утилиты *Regshot* был просканирован и создан снимок реестра до заражения. Была также запущена программа *Autoruns*. После выключения компьютера или перезагрузки ВПО требуется определенный механизм для продолжения работы на устройстве, иными словами, необходимо «закрепиться» в ОС. Для этого могут использоваться встроенные функции ОС *Windows*,

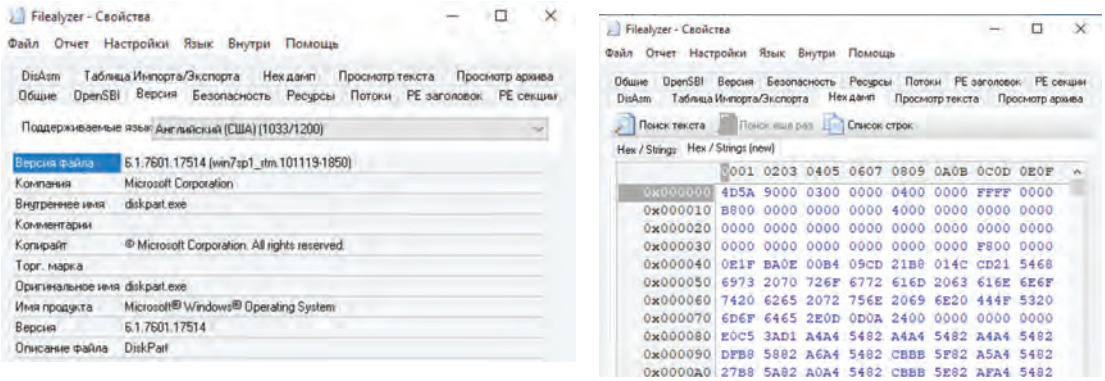

Рис. 5. Фрагменты окна программы Filealyzer, содержащего информацию о версии исследуемого файла VIRUS.exe и 16-ричный дамп файла

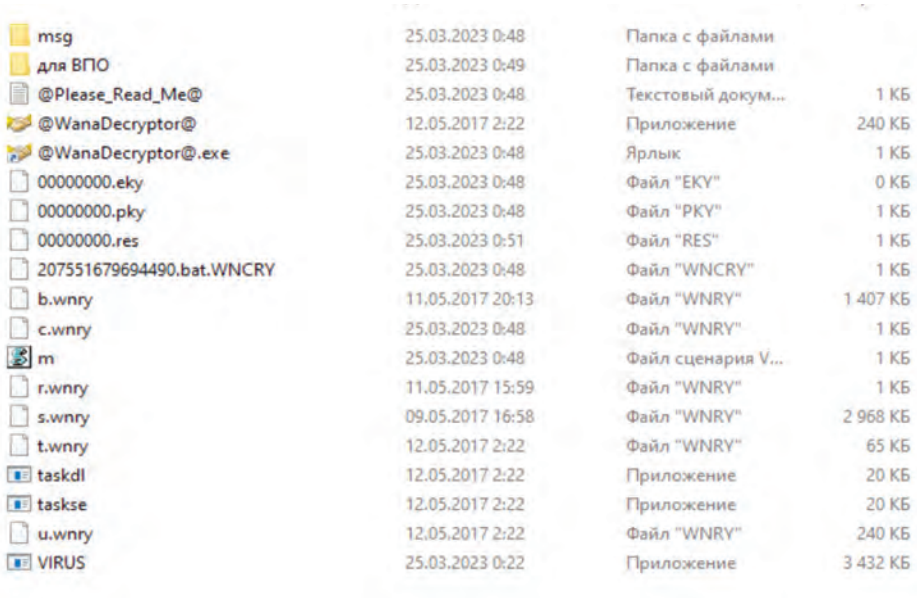

Рис. 6. Фрагмент окна программы «Проводник», содержащего распакованный файл VIRUS.exe

позволяющие запускать программы при загрузке. Утилита *Autoruns* способна управлять автозагрузкой программ, сервисов, модулей, драйверов и других компонентов системы и содержит информацию о необходимых компонентах. В частности, в данной работе была востребована *функция* данной утилиты — определение стандартных мест загрузки и места запуска программ. Был создан отчет о проверке системы данной утилитой и сохранен на рабочий стол виртуальной ОС.

Перед тем как запускать ВПО в виртуальной среде, необходимо начать записывать сетевой трафик. Любые передаваемые ВПО по сети данные обеспечивают его «корректную» работу и могут влиять на одну из характеристик информации, которая хранится и обрабатывается в системе, а именно — *целостность, доступность, конфиденциальность* [6]. В большинстве случаев ВПО генерирует в сети: отчет об инфицировании системы, собранные в системе учетные данные, принимаемые команды от управляющего сервера, загружаемые модули обновления «вредоноса», сетевой трафик, который используется для атак *DDoS*.

Для обнаружения подобных артефактов в виртуальной среде запускается программа *Wireshark*. Это профессиональная программа для захвата, мониторинга и анализа сетевого трафика и сетевых пакетов в режиме реального времени. С её помощью можно изучать сетевую активность.

Чтобы следить за происходящими изменениями в файловой системе и реестре, используется утилита *Process Monitor*. Программа контролирует и следит за всей работой ОС и отображает все происходящие процессы, работающие библиотеки, различные драйвера устройств, а также все изменения, происходящие с файлами, и выводит сообщение об их удалении или открытии. Она также включает в себя инструмент для мониторинга системного реестра и показывает, какие программы обращаются к нему (какие ключи читают и пытаются в них что-либо записать). Программа работает в режиме реального времени и позволяет записывать эти изменения в отдельный файл для последующего анализа. После того как были запущены программы *Process Monitor* и *WireShark*, был запущен файл *VIRUS.exe*.

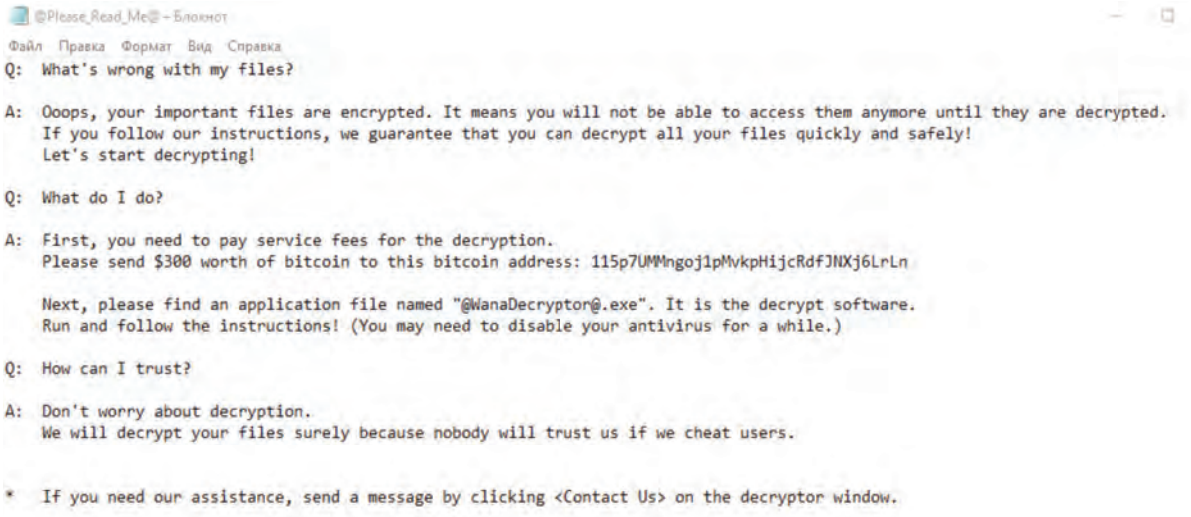

Рис. 7. Фрагмент окна программы «Блокнот», содержащего текст файла @Please_Read_Me@.txt

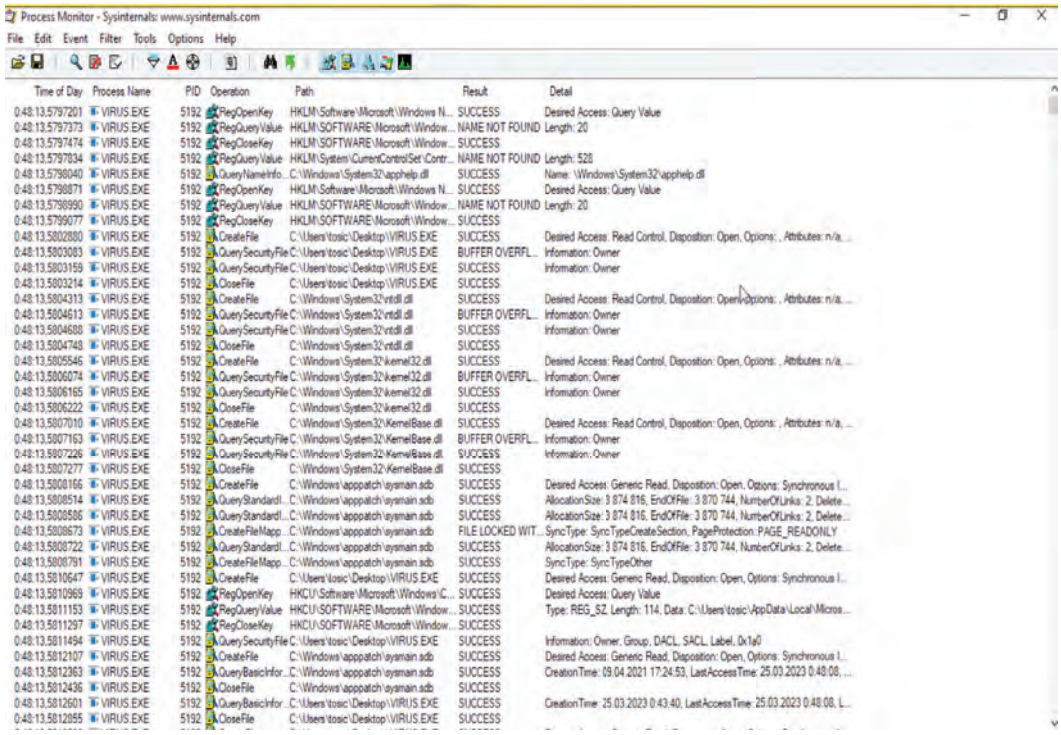

Рис. 8. Фрагмент окна программы Process Monitor, содержащего информацию о процессах, запущенных в ходе работы VIRUS.exe

После запуска файл был распакован (*рис. 6*). Появилось 2 новые папки и 16 новых файлов, 7 из которых с расширением *.wnry*. На рабочем столе компьютера появился текстовый документ с уведомлением о заражении, а также изменились обои рабочего стола (*рис. 7*).

Далее была открыта вкладка с программой *Process Monitor*. С помощью нее можно проследить все происходящие процессы, работающие библиотеки, различные драйвера устройств, а также все изменения, происходящие с файлами (их удаление, создание или изменение).

В реестре были найдены следы работы ВПО (*рис. 8—9*).

При помощи программы *RegShot* был создан снимок реестра после заражения и сравнение его с предыдущим снимком. Результат сохранен в отчет, который можно просмотреть в обычном текстовом или *HTML*-файле (*рис. 10*).

В результате сравнения: были добавлены 286 ключей в реестр, многие из которых с расширением *.wnry*; добавлено 1615 новых значений в реестр; изменено 123 значения; удалено 2 папки; создано 12 новых папок; в папке *VIRUS* были изменены атрибуты (к папке

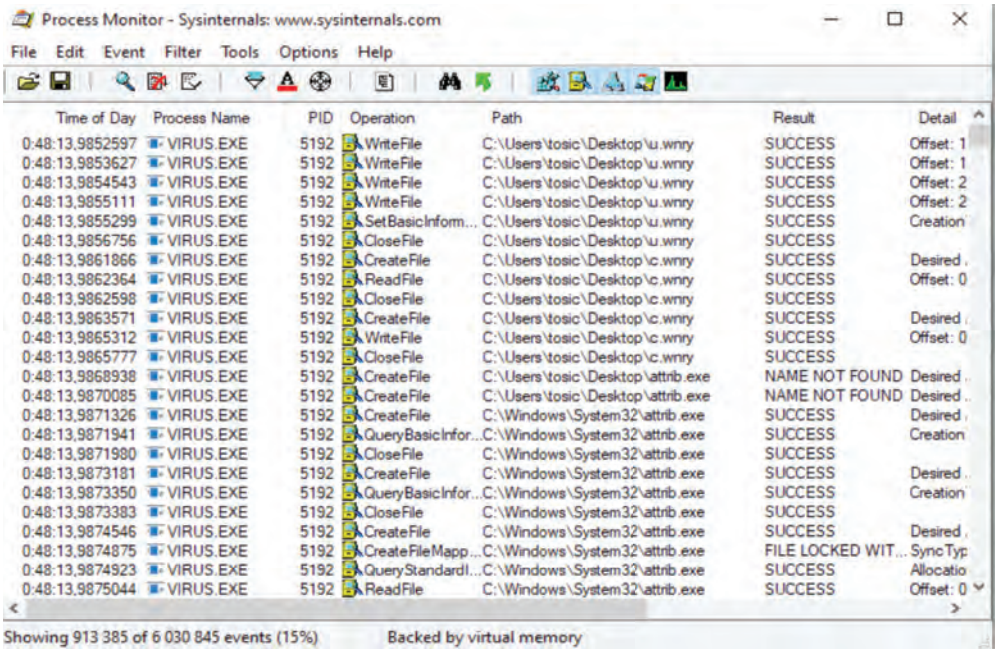

Рис. 9. Фрагмент окна программы Process Monitor, содержащего информацию о процессах VIRUS.exe

Regshot 2.1.0.17 unicode

Сводный отчёт

	Снимок A	Снимок B
Дата снимка	24.03.2023 23:55:21	25.03.2023 0:53:39
Компьютер	DESKTOP-0KFMCL0	DESKTOP-0KFMCL0
Пользователь	tosic	tosic
SID пользователя	S-1-5-21-2253714482-2605030331-1306191407-1001	S-1-5-21-2253714482-2605030331-1306191407-1001
Тип снимка	Реестр полностью	Реестр полностью
Время снимка	6.47	11.41
Ошибки	612	612
Ключи	197791	198077
Удалённые / Новые ключи	1	287
Изменённые ключи (разрешения)	67	67
Параметры	381354	382969
Удалённые / Новые параметры	6	1621
Изменённые параметры	0	123
Папки	0	0
Удалённые / Новые папки	2	12
Изменённые папки	0	0
Файлы	0	0
Удалённые / Новые файлы	0	0
Изменённые файлы	0	0
Альтернативные потоки	0	0
Удалённые / Новые альтернативные потоки	0	0
Изменённые альтернативные потоки	0	0
Всего изменений	197	22098

Рис. 10. Фрагмент окна браузера Google Chrome, содержащего отчет RegShot

предоставлен доступ для всех пользователей). Всего изменений: 21 901.

Далее была повторно запущена утилита *Autoruns*. С ее помощью было произведено сравнение с предыдущим отчетом программы о различных компонентах системы (*рис. 11*).

В результате дальнейшего глубокого анализа полученной информации можно прийти к *выводу*, что в реестр была добавлена запись для запуска исполняемого файла из папки *VIRUS*.

После открытия окна программы *WireShark* были исследованы сетевые протоколы: *DNS*, *HTTP* и *TCP*, которые содержат следы соединений и обмена данными

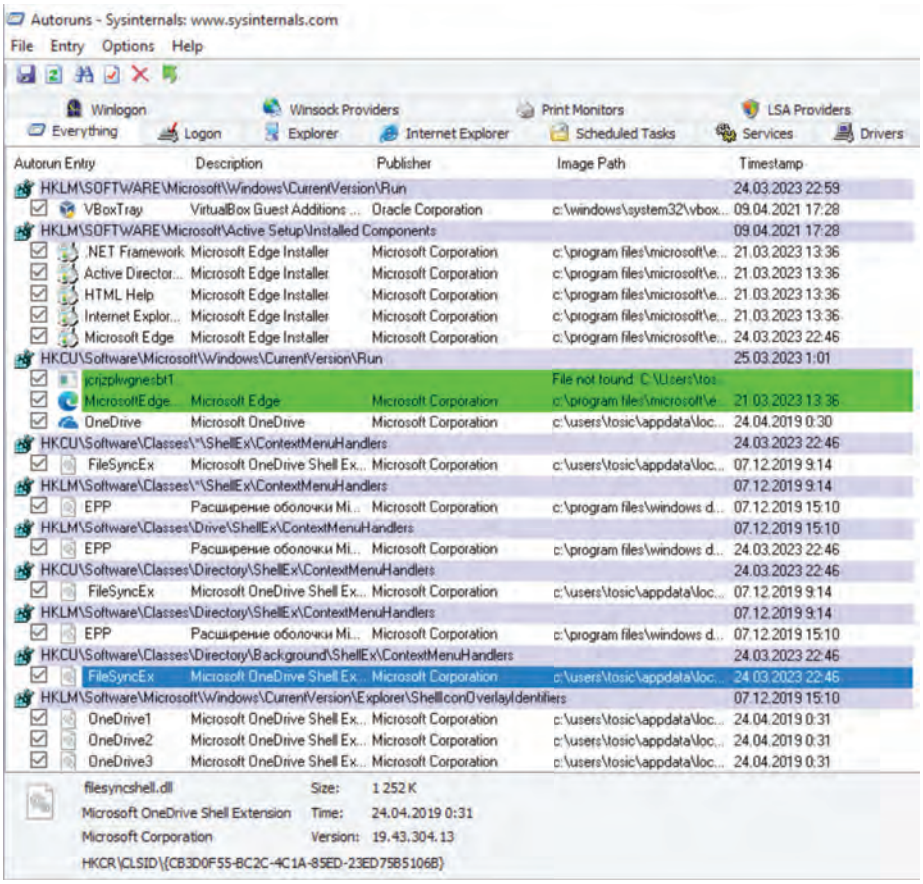

Рис. 11. Фрагмент окна программы Autoruns, содержащего информацию об изменениях в реестре

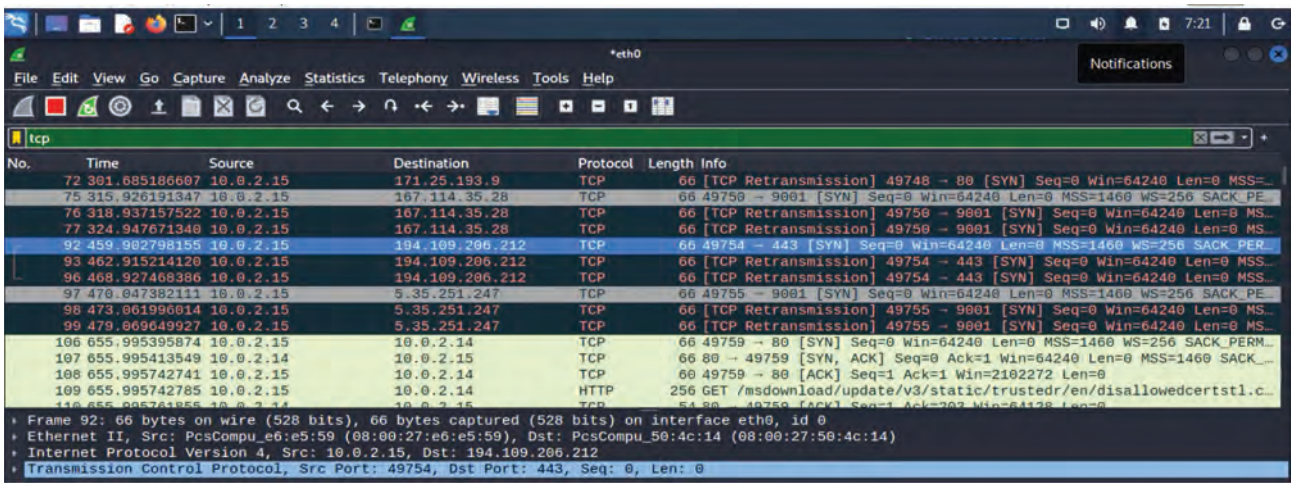

Рис. 12. Фрагмент окна программы «WireShark», содержащего информацию о сетевых протоколах и IP-адресах

в сети (*рис.* 12). В списке сетевых протоколов *TCP* (протокол управления передач, который обеспечивает доставку данных и гарантию сохранения порядка следования сообщений) найдены подозрительные *IP*-адреса.

Для проверки данных *IP*-адресов использовался *VirusTotal*, который осуществляет анализ подозрительных файлов и ссылок на предмет выявления ВПО. Система производит проверку 70 антивирусными сканерами и службами; анализирует *URL*-адреса, эвристические механизмы, подписи, метаданные[6].

После проверки *IP*-адресов установлено следующее:

– «194.109.206.212» — 5 антивирусных решений указали, что данный адрес вредоносный.

[6] Веб-сайт VirusTotal. URL: https://www.virustotal.com/gui/home/upload (дата обращения: 20.03.23).

– «167.114.35.28» — не оказался определен как вредоносный;
– «5.35.251.247» — 2 антивирусных решения классифицировали его как вредоносный.

То есть в ходе проведенного экспериментального компьютерного анализа потенциально вредоносного ПО в виртуальной среде удалось получить ряд криминалистически важных данных, которые свидетельствуют о способе закрепления ВПО в операционной системе; информацию о сетевой активности, обращениям к файловой системе и межпроцессном взаимодействии.

С помощью полученных сведений можно сделать *вывод*, что данная программа является вредоносной, определить функционал данной программы, а также представляется возможным классифицировать тип ВПО. Данный исполняемый файл относится к типу вредоносных программ, используемых киберпреступниками для шантажа и получения выкупа, а также саботажа или диверсии — программа-вымогатель или шифровальщик [9]. После заражения компьютера ВПО шифрует все (полнодисковое шифрование) или некоторые хранящиеся на компьютере файлы и предлагает заплатить денежный выкуп за их расшифровку.

Итак, были экспериментально проанализированы метаданные файла, его содержание, поведение при работе с процессами и операционной системой, сетевое взаимодействие и сформирован алгоритм поиска и обнаружения криминалистически значимой информации при практическом применении эмулятора ОС.

По итогам исследования можно выделить ряд *достоинств* реализованного способа исследования вредоносных программ. Они заключаются в эффективности противостояния полиморфизму вредоносных программ, что достигается за счет исследования действий, совершаемых программой, а не ее программного кода. Данный способ можно назвать наиболее безопасным, так как процесс анализа ВПО полностью изолирован от основной системы. В ходе использования данного способа эксперт может оперативно, без использования навыков отладки или обратной разработки, сделать *вывод* о функционале ВПО и впоследствии коррелировать полученные знания с данными, получаемыми при реализации других способов анализа.

Одним из существенных *недостатков* реализованного способа в настоящее время является отсутствие возможности его применения в юридической сфере при условии использования разработчиком ВПО техник обхода «песочниц» или запуска ВПО в виртуальной среде — в данном случае появляется необходимость глубокого анализа путем обратной разработки, так как в случае изменения кода программы суд может не рассмотреть заключение эксперта как доказательство.

Таким образом, в работе представлены методические рекомендации осуществления *эффективного* (оперативного, доступного, экономичного) криминалистического анализа на примере компьютерного исследования программы-вымогателя в виртуальной среде. Имитационное моделирование показало, что разработанные и экспериментально обоснованные методические рекомендации пригодны для исследования любых других типов ВПО и обеспечивают обнаружение криминалистически значимой информации. Поэтому представляется целесообразным активно использовать реализованный способ компьютерного криминалистического анализа в судебно-экспертной практике.

Рецензент: Моисеева Татьяна Федоровна, *доктор юридических наук, кандидат биологических наук, профессор, заслуженный деятель науки Российской Федерации, заведующий кафедрой судебных экспертиз и криминалистики имени Н. В. Радутной Российского государственного университета правосудия, Москва, Российская Федерация.*
E-mail: moiseevatf@mail.ru

Литература

1. Алексеев Д.С., Миндубаев В.В., Нефедов В.С. Методы выявления вредоносных программ в оперативной памяти на основе анализа аномалий // Chronos. 2022. Т. 7. № 11 (73). С. 144–146.
2. Бурый А.С. Тестирование качества программного обеспечения в процессе его сертификации // Правовая информатика. 2019. № 1. С. 46–55. DOI: 10.21681/1994-1404-2019-1-46-55 .
3. Гайнельзянова В. Р. Возможности судебной компьютерно-технической экспертизы при расследовании преступлений в сфере компьютерной информации // Вестник Уфимского юридического института. 2021. № 1 (91). С. 144–149.
4. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения: внедрение, управление и эксплуатация. М. : ЛОРИ, 2003. 567 с.
5. Казанцев А.О., Малюков В.О., Скакунов Р.С. Описание методов анализа и обнаружения вредоносных программ // Тр. XI Междунар. науч.-техн. и науч.-метод. конф. «Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2022)» (15–16 февраля 2022 г.). В 2-х тт. Т. 2. / СГУТ. СПб. : СПб. гос. ун-т телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2022. С. 253–256.
6. Ловцов Д.А. Теория защищенности информации в эргасистемах : монография. М. : РГУП, 2021. 276 с. ISBN 978-5-93916-896-0.

7. Льянов М.М. Вредоносные компьютерные программы как электронно-цифровые следы // Материалы криминалистических чтений (24 ноября 2022 г.) / Под ред. О. В. Кругликовой. Барнаул : Барнаульский юридический институт МВД РФ, 2022. С. 56–58.
8. Марков А.С. Модели оценки и планирования испытаний программных средств по требованиям безопасности информации // Вестник МГТУ им. Н.Э. Баумана. Сер.: Приборостроение. 2011. № 51. С. 90–103.
9. Россинская Е.Р., Рядовский И.А. Концепция вредоносных программ как способов совершения компьютерных преступлений: классификации и технологии противоправного использования // Всероссийский криминологический журнал. 2020. Т. 14. № 5. С. 699–709. DOI: 10.17150/2500-4255.2020.14(5).699-709 .
10. Умаров Д.А., Борисова С.Н. Исследование поведения вредоносных программ в защищенной среде исполнения // Инжиниринг и технологии. 2019. Т. 4. № 1. С. 47–50. DOI: 10.21685/2587-7704-2019-4-1-12 .
11. Фан Т.Х.Х. Анализ PE-заголовка для обнаружения вредоносных программ на основе машинного обучения // Наука настоящего и будущего. 2020. Т. 1. С. 277–280.
12. Юмашева Е.С. Анализ вредоносных программ // Тр. X Междунар. науч.-прак. конф. «Информационные управляющие системы и технологии (ИУСТ-ОДЕССА-2021)» (23–25 сентября 2021 г.) / ГУМРФ. Одесса : Гос. ун-т морс. и реч. флота им. адм. С.О. Макарова, 2021. С. 110–112.

COMPUTER CRIMINOLOGICAL STUDY OF MALWARE

Artem Belei, *Assistant Professor at the Department of Security in the Digital World of the Bauman Moscow State University, Ph.D. student at the Department of Forensic Science and Criminalistics of the Russian State University of Justice, Moscow, Russian Federation.*
E-mail: belyyskrat@yandex.ru

Stana Tomskaia, *student at the Department of Security in the Digital World of the Bauman Moscow State University, Moscow, Russian Federation.*
E-mail: miss.stana@mail.ru

Keywords: *malware, specialised software, expert, criminalistic study, forensic computer and technical assessment, activity, efficiency, virtual environment, virtual machine, method, experiment, criminalistically significant information.*

Abstract

Purpose of the work: analysing the efficiency of computer criminological study of malware using virtualisation tools and specialised software.

Methods used: simulating the operating system environment using modern virtualisation tools, carrying out an imitation experiment by means of placing malware into a virtual environment, inductive generalisation of conclusions and evaluation of the efficiency of computer criminological study using the knowledge obtained in the experiment.

Study findings: methodological recommendations for efficient practical implementation of specialised software in a criminalistic study of potentially malicious executable files are worked out and experimentally justified. Shortcomings in law enforcement practice related to such studies are identified which consist in the impossibility to use the studied method of computer analysis when the malware developer uses techniques for bypassing the execution of malicious code in a virtual environment.

Using the methodological recommendations contributes to speeding up the forensic expert's activities of establishing criminalistically significant information and studying questions concerning the malware's functionality and destination.

References

1. Alekseev D.S., Mindubaev V.V., Nefedov V.S. Metody vyiavleniia vredonosnykh programm v operativnoi pamiati na osnove analiza anomalii. Chronos, 2022, t. 7, No. 11 (73), pp. 144–146.
2. Buryi A.S. Testirovanie kachestva programmnogo obespecheniia v protsesse ego sertifikatsii. Pravovaia informatika, 2019, No. 1, pp. 46–55. DOI: 10.21681/1994-1404-2019-1-46-55 .
3. Gainel'zianova V. R. Vozmozhnosti sudebnoi komp'iuterno-tekhnicheskoi ekspertizy pri rassledovanii prestuplenii v sfere komp'iuternoi informatsii. Vestnik Ufimskogo iuridicheskogo instituta, 2021, No. 1 (91), pp. 144–149.
4. Dastin E., Reshka D., Pol D. Avtomatizirovannoe testirovanie programmnogo obespecheniia: vnedrenie, upravlenie i ekspluatatsiia. M. : LORI, 2003. 567 pp.

5. Kazantsev A.O., Maliukov V.O., Skakunov R.S. Opisanie metodov analiza i obnaruzheniia vredonosnykh programm. Tr. XI Mezhdunar. nauch.-tekhn. i nauch.-metod. konf. "Aktual'nye problemy infotelekommunikatsii v nauke i obrazovanii (APINO-2022)" (15–16 fevralia 2022 g.). V 2-kh tt, t. 2. SGUT. SPb. : SPb. gos. un-t telekommunikatsii im. prof. M. A. Bonch-Bruevicha, 2022, pp. 253–256.
6. Lovtsov D.A. Teoriia zashchishchennosti informatsii v ergasistemakh : monografiia. M. : RGUP, 2021. 276 pp. ISBN 978-5-93916-896-0.
7. L'ianov M.M. Vredonosnye komp'iuternye programmy kak elektronno-tsifrovye sledy. Materialy kriminalisticheskikh chtenii (24 noiabria 2022 g.). Pod red. O. V. Kruglikovoi. Barnaul : Barnaul'skii iuridicheskii institut MVD RF, 2022, pp. 56–58.
8. Markov A.S. Modeli otsenki i planirovaniia ispytanii programmnykh sredstv po trebovaniiam bezopasnosti informatsii. Vestnik MGTU im. N.E. Baumana, ser.: Priborostroenie, 2011, No. 51, pp. 90–103.
9. Rossinskaia E.R., Riadovskii I.A. Kontseptsiia vredonosnykh programm kak sposobov sovershenia komp'iuternykh prestuplenii: klassifikatsii i tekhnologii protivopravnogo ispol'zovaniia. Vserossiiskii kriminologicheskii zhurnal, 2020, t. 14, No. 5, pp. 699–709. DOI: 10.17150/2500-4255.2020.14(5).699-709 .
10. Umarov D.A., Borisova S.N. Issledovanie povedeniia vredonosnykh programm v zashchishchennoi srede ispolneniia. Inzhiniring i tekhnologii, 2019, t. 4, No. 1, pp. 47–50. DOI: 10.21685/2587-7704-2019-4-1-12 .
11. Fan T.Kh.Kh. Analiz PE-zagolovka dlia obnaruzheniia vredonosnykh programm na osnove mashinnogo obucheniia. Nauka nastoiashchego i budushchego, 2020, t. 1, pp. 277–280.
12. Iumasheva E.S. Analiz vredonosnykh programm. Tr. X Mezhdunar. nauch.-prak. konf. "Informatsionnye upravliaiushchie sistemy i tekhnologii (IUST-ODESSA-2021)" (23–25 sentiabria 2021 g.). GUMRF. Odessa : Gos. un-t mors. i rech. flota im. adm. S.O. Makarova, 2021, pp. 110–112.