Методы интеллектуального анализа в задаче обнаружения программ-вымогателей

М.И. Стародубов, А.Е. Боршевников, И.Л. Артемьева

Дальневосточный Федеральный Университет, Владивосток

Аннотация: Цель данной работы – проанализировать понятие угрозы программ-вымогателей, методы их обнаружения, а также рассмотреть методы интеллектуального анализа в решении задачи обнаружения, которые являются популярным средством среди исследователей программ-вымогателей и вредоносного программного обеспечения (ВПО) в целом. Интеллектуальный анализ данных помогает повысить точность и ускорить процесс обнаружения ВПО, обрабатывая большие объёмы информации. Благодаря этому специалисты могут выявлять новые, прежде неизвестные вредоносные программы. А с помощью генеративно-состязательных сетей можно обнаруживать вредоносное программное обеспечение нулевого дня. Несмотря на то, что прямое и объективное сравнение всех приведённых в работе исследований невозможно, в связи с разными наборами данных, можно предположить, что использование архитектуры генеративно-состязательных сетей является наиболее перспективным путём решения задачи обнаружения.

Ключевые слова: вредоносное программное обеспечение, программа-вымогатель интеллектуальный анализ, машинное обучение, нейронная сеть, генеративно-состязательная сеть

Введение

Компьютерная техника является неотъемлемой частью нашей жизни. Персональный компьютер или мобильное устройство с доступом в интернет есть почти у 65% (5,16 миллиарда) жителей земли (согласно статистике DataReportal, datareportal.com/global-digital-overview). При этом, на этих устройствах обрабатывается личная информация, выполняются платёжные транзакции и происходят другие операции как с пользовательскими и коммерческими данными, так и с государственной информацией. Поэтому, такие устройства представляют особый интерес для злоумышленников. Согласно статистике компании «Positive Technologies» доля атак с использованием вредоносного программного обеспечения (далее ВПО, вредоносного ПО) на такие устройства превышает 55 % и постоянно растёт (ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2022-q4/). При этом увеличивается не только количество атак, но и их сложность

(ptsecurity.com/ru-ru/research/analytics/ogo-kakaya-ib/). Злоумышленники становятся более квалифицированными, а методы для проникновения более изощрёнными. По этой причине возрастает и сложность вредоносных программ, а также их количество. Согласно статистике компании «Лаборатория Касперского» (securelist.ru/ksb-2022-statistics/106227/), в течение 2022 года атакам подверглось 15% от общего количества устройств, подключённых к сети Интернет.

Программы-вымогатели

Программа-вымогатель – тип вредоносного программного обеспечения, осуществляющий блокирование доступа к данным и/или компьютерной системы с целью получения выкупа за его возобновление (kaspersky.ru/resource-center/threats/ransomware).

Программы-вымогатели известны с конца 1989 года, однако они стали серьёзной угрозой сравнительно недавно, вследствие развития технологий шифрования и анонимных платёжных методов. В период с 2010 по настоящее время число семейств программ-вымогателей увеличилось в десятки, а число их образцов в сотни раз (securelist.com/state-of-ransomware-2023/112590/). В России первое использование программ-вымогателей было зарегистрировано в 2005 году (kaspersky.ru/resource-center/threats/ransomware) и с тех пор они плотно закрепились в нашей стране.

Точные экономические потери от атак с использованием вредоносных программ этого типа оценить достаточно трудно из-за неполных сведений о заражениях и изменения курса криптовалют. Однако, по минимальным оценкам, общая сумма выкупов постоянно растёт и уже превысила 1,1 миллиарда долларов (рис.1) (chainalysis.com/blog/ransomware-2024/). При этом основной урон наносится не за счёт выкупов, а за счёт потери данных, технологий и других последствий блокировок компьютерных систем

(cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/).

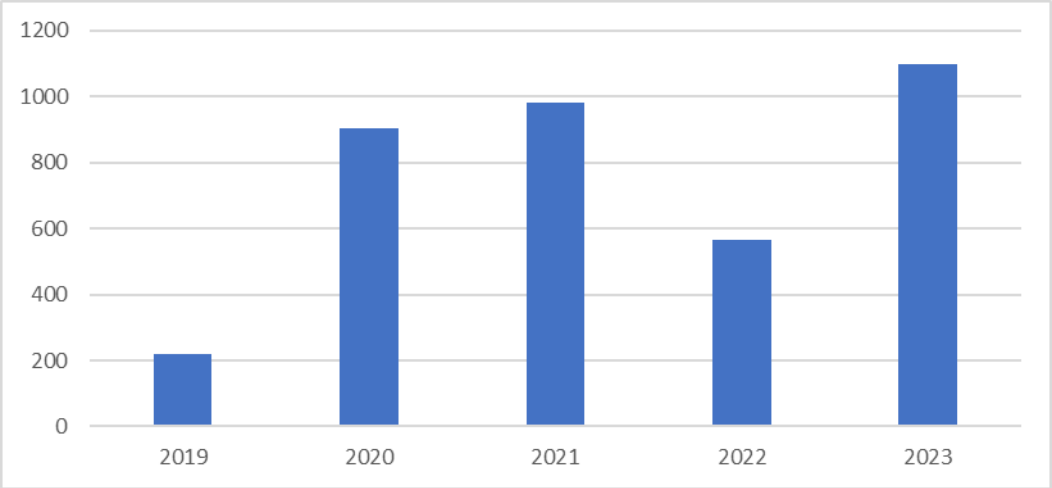

Рис. 1. – Общая сумма выкупов, полученных злоумышленниками в период с 2019 по 2023 годы, млн долларов

Резкое увеличение общей суммы выкупов в 2020 году также во многом связано с пандемией COVID-19 [1]. По мере того, как компании переходили на удаленный формат работы, их сотрудники становились всё более и более восприимчивыми к фишинговым атакам, тем самым создавая бреши в системе защиты.

Атаки программ-вымогателей обычно проходят в шесть основных этапов [2]:

- Распространение – начальный этап атаки, задача которого состоит в доставке вредоносной программы на целевое устройство.
- Заражение – этап установки программы-вымогателя на устройстве.
- Закрепление – этап подготовки вредоносной программы, на котором она закрепляется в целевой системе и начинает общение с «внешним миром» (другими компьютерными системами).
- Сканирование – вредоносная программа сканирует целевое устройство и все связанные с ним системы с целью поиска информации, которая может быть зашифрована.

- Шифрование – этап инициализации ключей шифрования и шифрование пользовательских данных.
- Вымогательство – последний этап работы программы-вымогателя, при котором отображается сообщение о выкупе и ожидается получение выкупа.

Распространение программ-вымогателей происходит различными путями [3 – 4]:

- Фишинг [5].
- Протокол удалённого рабочего стола (RDP) [6].
- Уязвимости программного обеспечения [7].
- Веб-страницы [8].

Существует несколько видов программ-вымогателей [9]:

1. Крипто-вымогатели, использующие стойкие алгоритмы шифрования, такие, как RSA, AES.
2. Блокировщики, блокирующие доступ жертвы к его компьютерной системе.
3. «Пугатели» (Scareware), показывающие пугающие сообщения, не нанося при этом реального вреда системе. С точки зрения оценки воздействия на систему не являются опасными.
4. «Воры» (Leakware), требующие выкуп не за восстановление доступа к файлам, а за недопустимость их появления в публичном поле.

Наиболее опасными являются программы-вымогатели первого вида, так как при правильной реализации восстановить зашифрованные данные без ключа за полиномиальное время практически невозможно.

Тенденции развития программ-вымогателей

В настоящий момент, когда технологии обнаружения и искусственного интеллекта постоянно совершенствуются, программы-вымогатели

развиваются в соответствии с общими тенденциями эволюции вредоносного кода и становятся всё более и более сложными. Перечислим наиболее важные методы избегания обнаружения:

1. Шифрование, сжатие и упаковка программного кода [9].
2. Полиморфизм и метаморфизм [10].
3. Мутация программного кода [11].
4. Технологии скрытия присутствия [12].
5. Технологии снижения эффективности антивирусного средства [13].
6. Распределение программного кода [14].
7. Атаки на антивирусные компании [15].

В последние годы также набирает обороты модель распространения «вымогатель, как услуга» (Raas) [3 – 4], аналогичная модели «программное обеспечение, как услуга» (SaaS) [16], при которой злоумышленники «сдают в аренду» программы-вымогатели другим злоумышленникам. Услуги такого типа позволяют быстро и легко проводить серьёзные атаки даже тем преступникам, у которых нет хороших навыков и времени для разработки собственных семейств программ-вымогателей. Широкое внедрение этой модели также способствует устойчивому росту количества атак с использованием программ-вымогателей [4].

В общих чертах, можно выделить следующие тенденции развития вредоносных программ:

1. Увеличение числа форм вредоносных программ и их сложности;
2. Интеллектуализация вредоносных программ;
3. Индустриализация разработки вредоносного кода.

Все представленные выше методы расположены в порядке увеличения их сложности. Они значительно повышают время обработки вредоносных программ и значительно уменьшает вероятность их обнаружения.

Методы анализа вредоносных программ

Анализ программы-вымогателя – процесс изучения образца вредоносного ПО с целью определения его характеристик, происхождения, поведения, назначения и потенциального воздействия. Так как программы-вымогатели является одним из типов вредоносных программ, то для них характерны все методы анализа вредоносных программ (рис.2), а также все методы обнаружения.

Рис. 2. – Подходы к анализу ВПО и выявление признаков

Метод статического анализа

Метод статического анализа позволяет изучить файл вредоносной программы без его выполнения, что может помочь выявить важную информацию об инфраструктуре вредоносного ПО, его целях и механизмах закрепления.

Данный подход имеет ряд преимуществ, главное из которых то, что анализ можно провести быстро, изучив особенности исполняемого объекта и сопоставив его с ранее обнаруженными вредоносными фрагментами кодов. Другим важным достоинством является возможность автоматизации, что упрощает и ускоряет процесс проверки больших объемов программного кода

на предмет наличия вредоносной активности. Однако исследователи также отмечают и недостатки, свойственные статическому анализу [17]:

– недостаточная информация о динамике выполнения;

– ложные срабатывания;

– ограниченность обнаружения сложных вредоносных программ.

Статический подход к анализу вредоносного ПО полезен для выявления некоторых видов угроз, но требует дополнительной поддержки других методов анализа для полного обеспечения безопасности системы.

Метод динамического анализа

Совершенно другим подходом является метод динамического анализа, для проведения которого используются различные инструменты, позволяющие отслеживать работу вредоносной программы, например:

– инструменты трассировки системных вызовов;

– дизассемблеры и отладчики;

– инструменты для мониторинга сетевой активности;

– системы виртуализации.

Целью динамического анализа является выявление функциональности вредоносного программного обеспечения.

В настоящее время используются две основные технологии для проведения динамического анализа: «песочница» (Sandbox) и виртуальная машина (VM), и они обе используют принципы виртуализации.

Виртуализация – это процесс создания виртуальной версии чего-либо, включая, аппаратные платформы компьютеров, устройства хранения данных и ресурсы компьютерной сети [18]. В контексте изолированной среды виртуализация позволяет создать полностью изолированную операционную среду, которая может запускать приложения как автономная система.

В основе технологии виртуализации лежат гипервизоры, или мониторы виртуальных машин (VMM), которые представляют собой программное обеспечение, встроенное ПО или аппаратное обеспечение, создающее и запускающее виртуальные машины (VM). Гипервизоры располагаются между оборудованием и виртуальной средой, распределяя физические ресурсы, такие как процессор, память и хранилище, для каждой виртуальной машины. Существуют два основных типа гипервизоров:

- Тип 1: запускаются непосредственно на оборудовании хоста для управления оборудованием и гостевыми операционными системами.

- Тип 2: работают в обычной операционной системе так же, как и другие компьютерные программы.

Виртуальная машина – это строго изолированный программный контейнер, который может запускать свои собственные операционные системы и приложения, как если бы это был физический компьютер. В изолированной среде часто используются виртуальные машины для репликации нескольких различных пользовательских сред.

При анализе потенциально вредоносного объекта с помощью виртуальной машины и соответствующих инструментов вирусный аналитик выполняет работу вручную, для автоматизации которой и агрегирования собранной информации применяются инструменты технологии sandbox.

В своей основе они используют принципы виртуализации и изоляции для создания безопасной среды, в которой потенциально вредоносный код может выполняться без риска для целостности хост-системы. Образцы вредоносных программ автоматически запускаются, регистрируется их поведение и генерируются подробные отчеты, которые включают в себя индикаторы компрометации (IoC), различные сигнатуры и правила обнаружения.

Архитектура популярного открытого решения Cuckoo Sandbox изображена на рис.3.

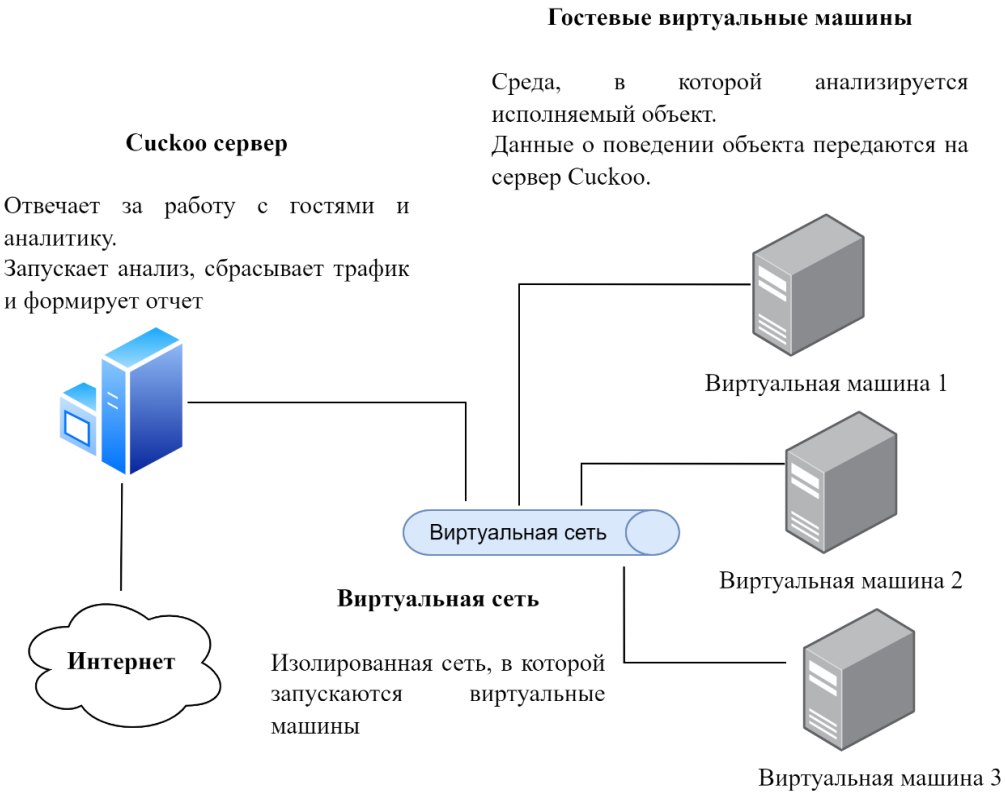

Рис. 3. – Основная архитектура Cuckoo Sandbox

При динамическом анализе, в сравнении со статическим, нет необходимости вручную разбирать потенциально опасный файл для его анализа. Методы обнаружения ВПО, основанные на динамическом анализе способны обнаруживать известные и неизвестные вредоносные программы, чего лишены методы, основанные на статическом анализе. Кроме того, обфусцированные и полиморфные вредоносные программы не могут избежать динамического обнаружения. Однако динамический анализ требует больше времени и ресурсов, по сравнению со статическим анализом [19].

Метод гибридного анализа

Метод гибридного анализа является сочетанием статического и динамического анализа, что позволяет получить более полное представление

о вредоносном ПО и его возможностях. Гибридный анализ – это мощный метод, который может быть использован для анализа широкого спектра вредоносных программ, включая программ-вымогателей. Однако, несмотря на все плюсы данного метода, у него есть один существенный недостаток – скорость выполнения анализа гибридного метода значительно ниже метода статического анализа и метода динамического анализа.

Методы обнаружения вредоносного программного обеспечения

Обнаружение вредоносного ПО заключается главным образом в обнаружении характерного вредоносного кода или опасного поведения. Методы обнаружения вредоносных программ подразделяются на несколько категорий. На рис.4 показаны основные методы обнаружения вредоносных программ.

Рис. 4. – Методы обнаружения вредоносного программного обеспечения

Рассмотрим основные концепции классических методов обнаружения вредоносных программ.

Сигнатурный анализ

Сигнатурный анализ в своей основе использует обнаружение сигнатур (определённых последовательностей), хранимых в базе данных анализатора, для идентификации вредоносного ПО и его типа. База данных содержит широкий спектр сигнатур уже известных вредоносных объектов. В случае совпадения сигнатуры объекта с любой сигнатурой из базы данных, анализатор определяет объект как вредоносный. В качестве сигнатур для

исполняемых файлов могут выступать строки, последовательности байт, последовательности программных кодов и другие данные [20 – 21]. Схема системы обнаружения представлена на рис.5.

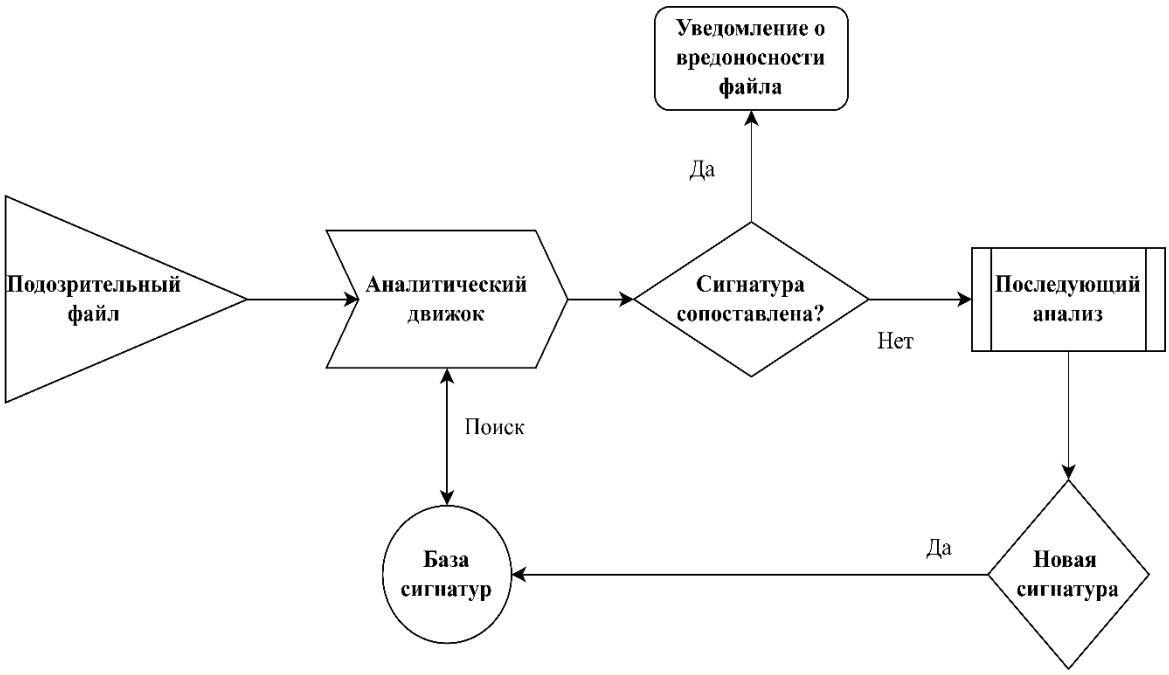

Рис. 5. – Схема обнаружения ВПО на основе сигнатурных методов

Сигнатурный анализ имеет ряд положительных черт: широкая применимость, простота работы, быстрая скорость работы и поиск исчерпывающей информации о вредоносном ПО. Несмотря на это, метод не лишён недостатков: неспособность обнаруживать полиморфное ВПО, зависимость размера базы сигнатур от числа обнаруживаемых объектов, необходимость постоянного обновления баз детектирующих логик и невозможность обнаружения ранее неизвестных вредоносных объектов [21 – 24].

Эвристический анализ

Другим известным методом обнаружения является эвристический анализ, включающий в себя использование не конкретизированных методов обнаружения для поиска новых и неизвестных вредоносных программ. Например, код файла (или другого объекта) проверяется на наличие

подозрительных инструкций. Если вес этих правил превышает заданный предел, предпринимаются превентивные действия, например, помещение файла в карантин [24 – 26]. Схема системы обнаружения представлена на рис.6.

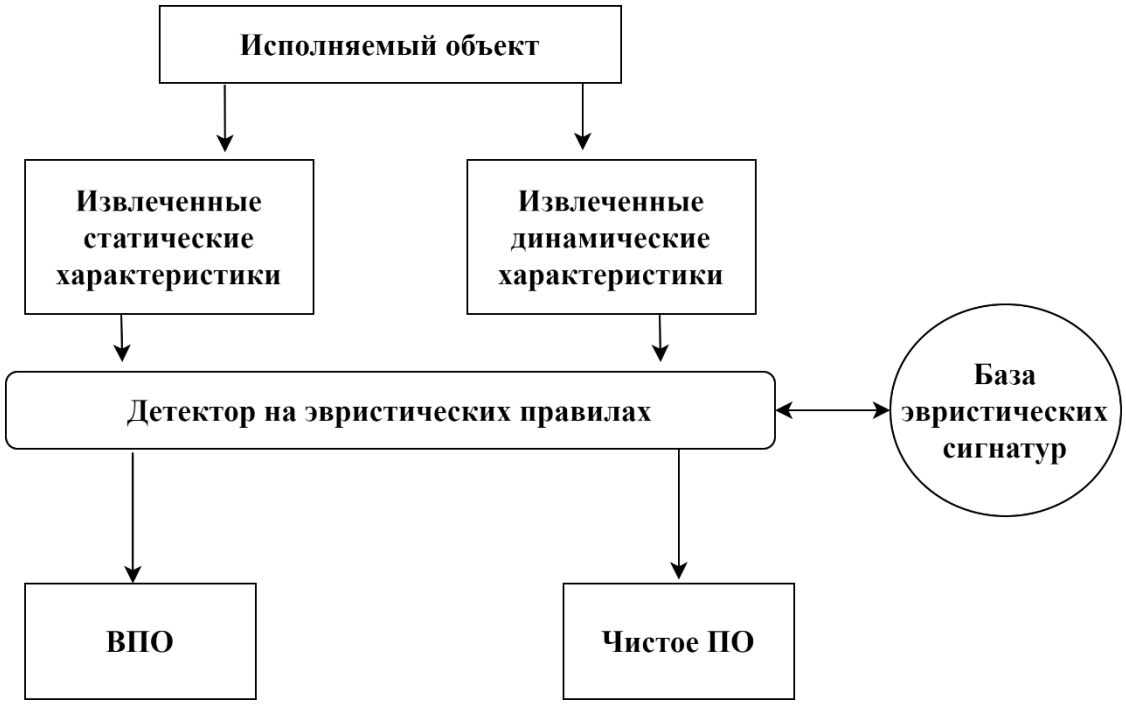

Рис. 6. – Схема обнаружения ВПО на основе эвристических методов

Метод эвристического анализа является более сложным, по сравнению с методом сигнатурного анализа, и имеет ещё ряд дополнительных положительных черт, главные из которых это возможность обнаружения полиморфного вредоносного ПО и возможность обнаруживать новые, не похожие на уже известные, вредоносные объекты.

Однако методы эвристического сканирования не могут гарантировать защиту от новых вредоносных программ, сигнатуры которых отсутствуют в базе сигнатур. Это связано с тем, что данный метод анализирует только известные сигнатуры и использует знания о механизме их полиморфизма. В то же время, поскольку этот метод поиска базируется на эмпирических предположениях, полностью исключить ложные срабатывания нельзя.

Чрезмерная подозрительность эвристического анализатора может вызывать ложные срабатывания при наличии в программе фрагментов кода, выполняющего действия, в том числе, свойственные и некоторым вирусам. В частности, распаковщик в файлах, запакованных PE-упаковщиком, вызывает ложные срабатывания целого ряда антивирусных средств, де-факто не признающих такой проблемы [10].

Метод анализа аномалий

Совершенно другим подходом является метод обнаружения, основанный на аномалиях, который заключается в проверке вредоносности программы путем обнаружения разницы между поведением аномальной и нормальной программы. Как правило, траектория поведения вредоносного ПО отличается от траектории поведения обычного программного обеспечения. После полного понимания поведения обычной программы будет сформирован набор стандартов и спецификаций. Если траектория поведения обнаруживаемой программы является ненормальной и нарушает этот набор спецификаций, она может быть определена как вредоносная [27].

На абстрактном уровне, аномалия определяется как паттерн, который не соответствует ожидаемому нормальному поведению. Таким образом, простой подход в её обнаружении заключается в определении области, представляющей нормальное поведение, и объявления аномалией любого наблюдения, не соответствующего области нормального поведения. Этот подход кажется достаточно простым, однако, существует несколько факторов, значительно его усложняющих [27]:

1. Определить нормальную область, которая охватывает все возможные варианты нормального поведения, очень сложно. Кроме того, граница между нормальным и аномальным поведением часто не является точной. Таким образом, аномальное наблюдение, лежащее близко к границе, на самом деле может быть нормальным, и наоборот.

2. В случае, когда аномалии являются результатом вредоносных действий, злоумышленники часто приспосабливаются, чтобы аномальные наблюдения выглядели нормальными, тем самым усложняя задачу определения нормального поведения.

3. Во многих областях нормальное поведение объекта постоянно изменяется и развивается. В связи с этим, нынешнее представление о нормальном поведении может оказаться недостаточно репрезентативным в будущем.

4. Доступность помеченных данных для обучения/валидации моделей, используемых методами обнаружения аномалий, обычно является серьезной проблемой.

5. Часто данные содержат шум, который, как правило, похож на фактические аномалии и, следовательно, его трудно различить и удалить.

Из-за этих проблем задачу обнаружения аномалий в ее самом общем виде решить непросто. На самом деле, большинство существующих методов обнаружения аномалий решают конкретную постановку проблемы. Формулировка определяется различными факторами, такими, как характер данных, доступность помеченных данных, тип аномалий, которые необходимо обнаружить, и так далее. Часто эти факторы определяются областью применения, в которой необходимо обнаружить аномалии. Исследователи взяли на вооружение концепции из различных дисциплин, таких как статистика, машинное обучение, интеллектуальный анализ данных, теория информации, спектральная теория, и применили их к конкретным постановкам задач. На рис.7 показаны ключевые компоненты, связанные с любым методом обнаружения аномалий.

Рис. 7. – Ключевые компоненты, связанные с методом обнаружения аномалий

Представленные выше подходы широко используются в задачах обнаружения вредоносных объектов, в таблице 1 представлен обобщённый сравнительный анализ наиболее важных исследовательских работ. На основе приведённых в ней исследований можно утверждать, что точность обнаружения достаточно сильно варьируется и не сильно зависит от метода обнаружения. Все представленные в исследованиях результаты были получены на основе анализа различных наборов данных, описанных недостаточно для воспроизведения точного результата. Несмотря на это, точность во всех работах оказалось высокой (86-100%), а приведённое выше предположение о возможности и невозможности обнаружения ранее неизвестного вредоносного программного обеспечения подтвердилось.

Несмотря на то, что методы обнаружения, представленные выше, играют очень важную роль, разработчики вредоносного кода часто используют различные средства для их обхода. Также они создают некоторые новые типы вредоносного кода или варианты вредоносного кода, в связи с чем точность традиционной модели обнаружения в этих случаях

будет значительно снижена. В связи с этим, необходимо постоянно развивать уже существующие методы обнаружения вредоносного ПО и создавать новые.

Таблица 1.

Методы обнаружения вредоносного программного обеспечения

Исследование	**Метод обнаружения**	**Тип данных**	**Возможность обнаружения ранее неизвестного ВПО**	**Точность (Accuracy), %**
[28]	Сигнатурный анализ	Исполняемые файлы формата APK	Отсутствует	86,56
[29]		Неизвестно		99,6
[30]		Сетевой траффик, системные журналы		Неизвестно
[31]	Эвристический анализ	Исполняемые файлы формата PE	Частичная	99,24
[32]				87-99
[33]				91.7 – 100
[34]				95,3
[35]				99,96
[36]		Исполняемые файлы формата APK		96
[37]		IoT приложения		99,93
[38]	Анализ, на основе обнаружения аномалий	Исполняемые файлы разных форматов	Присутствует	94,1-98,2
[39]				Неизвестно
[40]				98,43
[41]		Исполняемые файлы формата PE		93
[42]				87-92
[43]				93,75

Методы интеллектуального анализа

Одним из способов борьбы с вредоносным программным обеспечением является применение технологий интеллектуального анализа. Существует множество методов и алгоритмов, которые используются для обнаружения вредоносного программного обеспечения. Рассмотрим основные из них:

1. Методы анализа поведения – основаны на изучении поведения программы во время работы, для которых анализируются все действия, выполняемые программой, и сравниваются с известными алгоритмами вредоносных программ.

2. Методы статического анализа – основаны на анализе исходного кода программы, использующие алгоритмы статического анализа, выявляющие уязвимости, которые могут быть использованы злоумышленниками для внедрения вредоносного ПО.

3. Очень близким к методу анализа поведения является метод динамического анализа – метод, основанный на анализе программы во время выполнения в контролируемой среде, который позволяет выявлять новые виды вредоносных программ и обходить защиту от статического анализа. Данный метод отличается от метода анализа поведения средой исполнения: при динамическом анализе поведение образца изучается в контролируемой изолированной среде, в то время при методе анализа поведения образец изучается в реальных условиях на реальных системах.

4. Методы машинного обучения (далее МО, ML) и нейронные сети – методы, основанные на обучении алгоритмов распознавания вредоносных программ. Для этого используются нейронные сети, алгоритмы классификации и другие методы машинного обучения. Алгоритмы обучаются на множестве известных вредоносных объектов и затем применяются для распознавания новых видов ВПО.

Все представленные выше методы не являются взаимоисключающими и могут использоваться совместно. В таком случае часто говорят о применение «Гибридного метода обнаружения» – метода, включающего в себя несколько других методов обнаружения.

Классические методы машинного обучения

Согласно Тому Митчеллу [44], компьютерная программа обучается на опыте относительно некоторого класса задач и меры качества, если качество решения задач, измеренное мерой качества, возрастает с ростом опыта. Данный подход показал свою эффективность в большом количестве задач, в том числе, и в задаче обнаружения вредоносного программного обеспечения.

Таблица 2.

Методы машинного обучения в решении задачи обнаружения ВПО

Исследование	**Методы машинного обучения**	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Эффективность**
[45]	NB	-	-	1001 чистых, 3265 вредоносных	Accuracy: NB – 97,76 % MNBayes – 97,76 %
[46]	ANN, DT, NB и SVM	-	-	22735 чистых, 7688 вредоносных	Accuracy: ANN – 94,1 % DT – 94,3 % NB – 69,7 % BDT – 94,9 % BNB – 69,7 % SVM-lin – 92,1 % SVM-poly – 85,2 % SVM-rbf – 93,9 %
[47]	NB, ANN, DT, Boosted DT(BDT)	-	-	22735 чистых, 7688 вредоносных	Accuracy: ANN – 92,13 % DT – 93 % BDT – 94,43 % NB – 84,53 % BNB – 84,53 %
[48]	KNN, NB, J48 DT, SVM и MLP	-	-	250 чистых, 220 вредоносных	Accuracy: KNN – 92,9 % NB – 92,3 % SVM – 97,7 % J48 –96,8 % MLP – 94,2 %
[49]	Контролируемое обучение, метод «Informaition gain», DT, KNN, BN, SVM	-	-	1000 чистых, 1000 вредоносных	Accuracy: ~87 %
[50]	Автоматический анализ поведения вредоносных программ с использованием машинного обучения	-	-	3133 вредоносных, число чистых неизвестно	F1-мера: 95 %
[51]	DT, SVM, MLP, GMDH, PNN, RF	-	-	100 чистых, 117 троянских программ, 165 вирусных файлов, 134 червя.	Accuracy: DT – 98,46 % SVM – 98,63 % MLP – 90,8 % GMDH – 94,49 % PNN – 89,67 % RF – 90,3 %

Таблица 3.

Методы машинного обучения в решении задачи обнаружения ВПО

Исследование	**Методы машинного обучения**	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Эффективность**
[52]	Улучшенная последовательность API	-	-	1226 - чистых 1316 вредоносных	Accuracy: 95,42 %
[53]	KNN	-	-	1160 чистых, 1050 вредоносных	Accuracy: 97.66 %
[54]	LightGBM и DT	-	-	1447 чистых, 15925 вредоносных	ROC AUC: 99,821 %
[55]	Ансамбль из пяти алгоритмов классификации машинного обучения	-	-	7212 чистых, 21653 вредоносных	Accuracy: RF – 100 % LMT – 100 % NBT – 100 % J48 – 100 % REPTREE – 99,96 %
[56]	Анализ использования разрешений - Sigpic.	-	-	310926 чистых, 5494 вредоносных	Accuracy: 95,63 %
[57]	Ансамбль из 6 алгоритмов: KNN, SVM, Random Forest, AdaBoost, GBDT, and LightGBM	-	-	163946 чистых, 18168 вредоносных	Accuracy: 94.5%
[58]	RA, DT, KNN, AdaBoost, SGD, Extra Trees, GNB	-	-	72 чистых, 301 вредоносных	Accuracy: RF – 99 % DT – 99 % KNN – 99 % AdaBoost – 99 % SGD – 100 % Extra Tree – 100 % Gaussian NB – 100 %

Особенно хорошо себя показали широко распространённые методы машинного обучения, такие, как искусственные нейронные сети (ANN) [45 – 47], байесовская сеть (BN) [45, 49], дерево решений (DT) [45 – 49, 51, 55], случайный лес (RF) [55, 57], классификатор наивного Байеса (NB) [46, 48,

55], машина опорных векторов (SVM) [46, 48, 49, 57], метод k-ближайших соседей (KNN) [48, 49, 53, 57], LightGBM [54, 57] и другие.

Активно и успешно для отбора признаков применяются коэффициент усиления (GR) [46, 47], оценка Фишера (FS) [45, 46, 47] и метод «Information gain» [49]. В качестве признаков используются как статические параметры объектов [45 – 49], так и динамические, такие как поведение [50] и последовательности вызовов API [51 – 53]

Как видно из представленных в таблице 2 и таблице 3 исследований методы машинного обучения имеют достаточно высокие показатели точности независимо от объёма и сбалансированности данных. Некоторые исследования [55, 58] позволяют достичь 100% точности. Однако такой идеальный результат выглядит подозрительно, и, вероятнее всего, связан не с хорошим результатом, а с недостаточным объёмом данных. Также, использование указанных подходов не позволяет обнаруживать ранее неизвестные вредоносные программ, в том числе и ВПО нулевого дня.

Нейронные сети

Помимо классических методов машинного обучения, таких, как алгоритмы кластеризации и классификации, в анализе вредоносного программного обеспечения также активно применяются нейронные сети различных архитектур и методы глубокого обучения. Часто утверждается, что главное преимущество таких методов возникает тогда, когда объем обучающих данных велик. Например, в руководстве [59] приведены данные, которые призваны показать, что глубокое обучение будет продолжать достигать улучшенных результатов по мере увеличения размера набора данных, в то время как другие методы машинного обучения выйдут на плато на каком-то относительно раннем этапе. То есть модели, созданные с помощью методов, не связанных с глубоким обучением, будут “насыщаться” относительно быстро, и как только эта точка насыщения будет достигнута,

большее количество данных не даст улучшения. Напротив, предполагается, что глубокое обучение будет продолжать обучение без ограничений, по мере увеличения объема данных, или, по крайней мере, оно достигнет плато на гораздо более высоком уровне. Конечно, даже если это полностью верно, существуют практические вычислительные ограничения, поскольку для получения большего количества данных требуется больше вычислительной мощности для обучения.

Многослойный персептрон (MLP)

Архитектура многослойного персептрона (MLP) чрезвычайно популярна, и в большинстве областей является одним из первых рассмотренных методов обучения. Информационная безопасность не является исключением, поскольку MLP был применен практически ко всем проблемам безопасности, где применимы методы глубокого обучения. Неудивительно, что в большом количестве исследовательских работ по вредоносным программам используются MLP [60, 61].

Таблица 4.

Модели многослойного персептрона (MLP) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Точность (Accuracy), %**
[60]	+	-	1000 - чистых, 20000 вредоносных	93,97
[61]	+	-	40 - чистых, 900 вредоносных	Неизвестно

Как видно из таблицы 4, архитектура многослойного персептрона позволяет определять новые вредоносные объекты, похожие на уже известные, но не вредоносные программы нулевого дня. Однако,

эффективность такой архитектуры ниже методов машинного обучения даже на небольшом объёме данных.

Свёрточные нейронные сети (CNN)

Свёрточные нейронные сети (CNN) доказали свою состоятельность в широком спектре приложений, связанных с безопасностью. Некоторые из этих приложений, такие, как обнаружение спама с изображениями [62 – 64], являются очевидными и относительно простыми вариантами использования CNN. Однако другие проблемы безопасности, в которых нет какого-либо очевидного компонента, основанного на изображениях, также добились успеха с использованием CNN. Рассматривая исполняемые файлы как изображения, исследователи смогли использовать сильные стороны CNN для обнаружения, классификации и анализа вредоносных программ. Например, в ряде исследований [65, 66] исполняемые файлы рассматриваются как изображения. Также CNN активно используются в задаче обнаружения ВПО для IoT устройств [67, 68] и детектирования вредоносных android приложений [69].

Таблица 5.

Модели свёрточных нейронных сетей (CNN) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Точность (Accuracy), %**
[65]	+	-	Malimg dataset	99,60
[66]	+	-	Malimg dataset	100
[67]	+	-	2486 - чистых, 1473 вредоносных	97,18
[68]	+	-	12000, соотношение неизвестно	99,14
[69]	+	-	3200 - чистых, 10200 вредоносных	99,94
[70]	+	-	9000 вредоносных файлов	99

Как видно из таблицы 5, точность обнаружения при использовании свёрточных нейронных сетей значительно выше, чем при использовании многослойного персептрона, и достигает уровня методов машинного обучения. При этом возможность обнаружения новых объектов сохраняется.

Машины экстремального обучения (ELM)

Новым этапом развития свёрточных нейронных сетей является использование машин экстремального машинного обучения (ELM), которые тоже использовались в задаче обнаружения вредоносного ПО.

Например, в работах [71 – 73] приводится сравнение CNN с ELM на одном и том же наборе данных на разных платформах. При этом, что впечатляет, время обучения ELM значительно снижается по сравнению с CNN. В работе [74] авторы рассматривают эффективность метода высокопроизводительных машин экстремального обучения (HP-ELM). Также в работе [76] ELM используется для анализа обфусцированных вредоносных файлов, а в работе [77] ELM эффективно используется для анализа полиморфного и метаморфного вредоносного программного обеспечения.

Таблица 6.

Модели машин экстремального обучения (ELM) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Точность (Accuracy), %**
[71]	+	-	Malimg dataset	97,7
[72]	+	-	Неизвестно	96,8
[73]	+	-	1100 - чистых, соотношение неизвестно	97,52
[74]	+	-	CTU-13	90,17
[75]	+	-	10825, соотношение неизвестно	99.03
[77]	+	-	Malimg dataset	Неизвестно

Исходя из сравнительного анализа, приведённого в таблице 6, видно, что точность обнаружения, по сравнению с использованием CNN без ELM, сохраняется.

Рекуррентные нейронные сети (RNN)

Модели рекуррентных нейронных сетей (RNN) тоже активно применяются в решении задачи обнаружения вредоносного программного обеспечения. В коммерческом смысле модели долгой краткосрочной памяти (LSTM), безусловно, являются наиболее успешной технологией глубокого обучения, из когда-либо разработанных, поэтому неудивительно, что LSTM также были успешно применены для решения проблемы обнаружения вредоносных программ для ОС Windows [78 – 80], Linux [81, 82], Android [83, 84] и IoT [85, 86]. Как LSTM, так и модель управляемого рекуррентного блока (GRU) рассматриваются в работе [87]. Причем авторы заявляют о значительном увеличении показателей по сравнению с соответствующей предыдущей работой. В статье [88] рассматривается состязательная атака, при которой злоумышленник может победить систему, использующую RNN на основе вызовов API. В работе [89] представлена система раннего обнаружение программ-вымогателей в инфраструктуре IoT, а в работе [90] представлена похожая система для обнаружения разных классов ВПО для ОС Android .

Как видно из таблицы 7 точность обнаружения не уступает представленным ранее работам с использованием CNN, MLP и классическим методам машинного обучения.

Существует множество применений RNN в областях информационной безопасности за пределами области вредоносных программ: обнаружение событий кибербезопасности на основе сообщений в социальных сетях [91], генерация онтологий безопасности [92], взлом капч [93], обнаружение вторжений на основе хоста [94] и обнаружение сетевых аномалий [95].

Таблица 7.

Модели рекуррентных нейронных сетей (RNN) в решении задачи обнаружения ВПО

Исследование	**Методы RNN**	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Эффективность**
[78]	LSTM	+	-	123 - чистых, 969 вредоносных	Accuracy: 97,87 %
[79]		+	-	Mal-API-2019, CIC Maldroid 2020	Accuracy: 93,16 %
[81]		+	-	50000 - чистых, 50000 вредоносных	Accuracy: 97 %
[82]		+	-	Неизвестно	Accuracy: > 95 %
[83]		+	-	179000 - чистых, 5560 вредоносных	F1-мера: 94,07 %
[86]		+	-	Неизвестно	Accuracy: 95 %
[84]		+	-	Неизвестно	Неизвестно
[85]	CNN + LSTM	+	-	Microsoft Malware Dataset, IoT Malware Datase	F1-мера: 100 %
[80]	CNN + GRU	+	-	1285 - чистых, 1285 вредоносных	Accuracy: 99,07 %
[87]	LSTM + GRU	+	-	75000	Отсутствуют в явном виде
[88]	Bi-GRU	+	-	5850 - чистых, 5585 программ-вымогателей	MSE: 0.5314
[90]	GNN + LSTM	+	-	5727 чистых, 4350 вредоносных	Accuracy: 91,41 %

Остаточные свёрточные нейронные сети (ResNet)

На момент написания работы архитектура остаточных свёрточных нейронных сетей (ResNet) в задаче обнаружения ВПО не применяется активно. Тем не менее, архитектура ResNet показала многообещающие возможности в задачах анализа вредоносных программ [96 – 99] и обнаружения вторжений [100 – 103].

Таблица 8.

Модели остаточных свёрточных нейронных сетей (ResNet) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Эффективность**
[96]	+	-	DREBIN	F1-мера: 97,8 %
[97]	+	-	Malimg, MaleVis	Accuracy: 99,46 %
[98]	+	-	CICMalDroid2020	Accuracy: 99,20 %
[99]	+	-	5727 чистых, 4350 вредоносных	Accuracy: 96,48 %

Как видно из таблицы 8, эффективность обнаружения достаточно высокая, а на одних и тех же данных практически полностью совпадает со свёрточными нейронными сетями и машинами экстремального обучения.

Графовые нейронные сети (GNN)

Другой мало используемой архитектурой в задаче обнаружения вредоносного ПО является архитектура графовой нейронной сети (GNN). Несмотря на это, они показали свою продуктивность в задаче классификации вредоносных объектов [104] и показали многообещающие возможности в задаче обнаружения вредоносных объектов [105]. Эта архитектура не является самостоятельной, это адаптация других архитектур для работы с графами.

Из таблицы 9 видно, что показатели графовых нейронных сетей достаточно высокие, и подход кажется перспективным.

Таблица 9.

Модели графовых нейронных сетей (RNN) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Точность (Accuracy), %**
[104]	+	-	MalNet-Tiny dataset, 5000 файлов	97,69
[105]	+	-	Неизвестно	Неизвестно
[106]	+	-	Drebin dataset – 5560 вредоносных объектов	99.22

Генеративно-состязательные сети (GAN)

Генеративно-состязательные сети (GAN), по-видимому, обещают решить некоторые из самых сложных проблем в области информационной безопасности. Например, GAN с некоторым успехом применялись для обнаружения вредоносных программ нулевого дня [107 – 110]. Кроме того, генеративный аспект GAN может быть использован для создания сложных проблем безопасности в “лаборатории”, что позволяет исследователям рассмотреть средства защиты от потенциальных угроз до того, как эти угрозы возникнут в реальных условиях [111].

Таблица 10.

Модели генеративно-состязательных сетей (GAN) в решении задачи обнаружения ВПО

Исследование	**Возможность определения новых вредоносных объектов**	**Возможность определения вредоносных программ нулевого дня**	**Набор данных**	**Точность (Accuracy), %**
[107]	+	+	180000 объектов, из которых 30% - вредоносные	100 %
[108]	+	+	Kaggle: Microsoft Malware Classification Challenge 2015 dataset	95,74 %
[109]	+	+	75503 чистых, 140849 вредоносных	96,42 %

Как видно из описанных в таблице 10 исследований, эффективность обнаружения достаточно высокая. К тому же, данный подход позволяет обнаруживать новое вредоносное ПО, в том числе и вредоносное программное обеспечение нулевого дня.

Сравнительный анализ

В результате анализа указанных выше работ, было определено, что средние показатели точности (accuracy) методов обнаружения близки друг к другу (таблица 11).

Таблица 11.

Методы обнаружения

Методы обнаружения	Возможность определения новых образцов ВПО	Возможность определения ВПО нулевого дня	Средний показатель точности (Accuracy), %
Сигнатурный анализ	-	-	93,08
Эвристический анализ	+	-	96,46
Обнаружение аномалий	+	+	96,12

Все представленные в исследованиях результаты были получены на основе анализа различных наборов данных, описанных недостаточно для воспроизведения точного результата. Несмотря на это, точность во всех работах оказалось высокой.

Ключевым моментом сравнения является предположение о возможности обнаружения новых вредоносных программ. Метод сигнатурного анализа не позволяет обнаруживать новые вредоносные объекты, а метод эвристического анализа дает возможность обнаруживать только новые вредоносные объекты, похожие на ранее известные. В то же время, метод обнаружения аномалий кажется наиболее перспективным, так как только при его использовании возможно определение не только новых вредоносных объектов, похожих на предыдущие, но и вредоносных программ нулевого дня.

Таблица 12.

Методы интеллектуального анализа в задаче обнаружения ВПО

Архитектуры	**Возможность определения новых образцов ВПО**	**Возможность определения ВПО нулевого дня**	**Средний показатель точности (Accuracy), %**
ML	-	-	93,96
MLP	+	-	93,97
CNN	+	-	97,92
ELM	+	-	95,57
RNN	+	-	95,84
ResNet	+	-	98,23
GNN	+	-	98,20
GAN	+	+	97,39

В дополнение к классическим методам обнаружения, методы интеллектуального анализа активно используются в задаче обнаружения ВПО. Как видно из таблицы 12, методы интеллектуального анализа позволяют достичь высоких показателей точности (accuracy). Важно отметить, что показатели всех исследований достаточно высокие и зависят больше от набора данных, чем от используемого подхода. Несмотря на это, архитектуры нейронных сетей в среднем достигают более высоких показателей, чем методы машинного обучения. При этом самыми перспективными кажутся нейронные сети архитектуры GAN, которые позволяют обнаруживать новые угрозы до момента их появления в реальных системах, в том числе, и ВПО нулевого дня, чего другие архитектуры не позволяют.

Заключение

Задача обнаружения вредоносного программного обеспечения представляется сложной и нетривиальной. Программы-вымогатели являются одним из самых опасных классов вредоносного программного обеспечения, методы обнаружения которого не отличаются от методов обнаружения любого другого класса ВПО. При этом, в связи с ростом количества и сложности вредоносных программ этого класса, классических методов обнаружения: сигнатурного и эвристического, – уже недостаточно для

эффективного обнаружения новых, ранее неизвестных вредоносных программ. На данный момент только метод обнаружения аномалий имеет такую возможность. Однако в общем случае решить эту задачу не просто, так как существует ряд факторов, значительно её усложняющих.

В этом случае на помощь приходят методы интеллектуального анализа, которые позволяют:

1. Повысить точность обнаружения. Методы интеллектуального анализа позволяют выявлять скрытые связи и закономерности в данных, что может помочь повысить точность обнаружения ВПО.
2. Ускорить процесс обнаружения. Методы интеллектуального анализа могут автоматизировать процесс обнаружения ВПО, что позволяет ускорить процесс и снизить количество ложных срабатываний.
3. Анализировать большие объемы данных. Обнаружение ВПО требует анализа больших объемов данных, включая файлы, сетевой трафик и действия пользователей. Методы интеллектуального анализа позволяют автоматически анализировать такие данные и выявлять угрозы.
4. Обнаруживать новые угрозы. ВПО постоянно эволюционирует и появляются новые виды угроз. Методы интеллектуального анализа могут помочь выявить новые угрозы, которые не были обнаружены ранее.

Использование методов интеллектуального анализа в задаче обнаружения ВПО повышает эффективность обнаружения, сокращает время реакции на угрозы, снижает количество ложных срабатываний, а также позволяет не только обнаруживать вредоносные объекты, похожие на уже известные, но и, используя модели генеративно-состязательных сетей, обнаруживать ранее неизвестные вредоносные программы, в том числе, и ВПО нулевого дня.

Несмотря на то, что прямое и объективное сравнение всех приведённых в работе исследований невозможно, в связи с разными наборами данных,

можно предположить, что использование архитектуры генеративно-состязательных сетей является наиболее перспективным путём решения задачи обнаружения.

Исследование проведено при финансовой поддержке Минобрнауки России («Грант ИБ МТУСИ») № 40469-25-23-К.

Литература

1. Lallie H.S. et al. Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic // Computers & security. 2021. V. 105. URL: sciencedirect.com/science/article/pii/S0167404821000729.

2. Mirza Q.K.A. et al. Ransomware analysis using cyber kill chain // 8th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2021. P. 58-65.

3. Аникин И.В., Исяндавлетова Я.М. Реверсивный анализ вредоносного программного обеспечения Raccoon Stealer // Инженерный вестник Дона. 2023. №. 4. URL: ivdon.ru/ru/magazine/archive/n4y2023/8346.

4. Alwashali A.A. M.A., Abd Rahman N.A., Ismail N. A survey of ransomware as a service (RaaS) and methods to mitigate the attack // 14th International Conference on Developments in eSystems Engineering (DeSE). IEEE, 2021. pp. 92-96.

5. Alkhalil Z. et al. Phishing attacks: A recent comprehensive study and a new anatomy // Frontiers in Computer Science. 2021. V. 3. URL: frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2021.563060.

6. Ramirez R. Behavioral Characterization of Attacks on the Remote Desktop Protocol. 2022. URL: apps.dtic.mil/sti/citations/trecms/AD1201693.

7. Oz H. et al. {RøB}: Ransomware over Modern Web Browsers // 32nd USENIX Security Symposium (USENIX Security 23). 2023. pp. 7073-7090.

8. Razaulla S. et al. The age of ransomware: A survey on the evolution, taxonomy, and research directions // IEEE Access. 2023. V. 11. URL: ieeexplore.ieee.org/abstract/document/10105244.

9. Muralidharan T. et al. File packing from the malware perspective: techniques, analysis approaches, and directions for enhancements // ACM Computing Surveys. 2022. V. 55. №. 5. P. 1-45.

10. Badhwar R. Polymorphic and metamorphic malware // The CISO's Next Frontier: AI, Post-Quantum Cryptography and Advanced Security Paradigms. Cham : Springer International Publishing, 2021. P. 279-285.

11. Murali R., Thangavel P., Velayutham C.S. Evolving malware variants as antigens for antivirus systems // Expert Systems with Applications. 2023. V. 226. URL: sciencedirect.com/science/article/abs/pii/S0957417423005948.

12. Kara I. Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges // Expert Systems with Applications. 2023. V. 214. URL: sciencedirect.com/science/article/abs/pii/S0957417422021510.

13. Zhang X. et al. Android application forensics: A survey of obfuscation, obfuscation detection and deobfuscation techniques and their impact on investigations // Forensic Science International: Digital Investigation. 2021. V. 39. URL: sciencedirect.com/science/article/pii/S2666281721002031.

14. Damjanović D.Z. Malicious code in the cloud // Vojnotehnički glasnik/Military Technical Courier. 2022. V. 70. №. 3. pp. 734-755.

15. Beardwood J. There'sa Strong Wind a'Blowing re Security Disclosures: Lessons Learned from SolarWinds—An analysis of the significance of the SolarWinds case for companies with making security statements on websites and in public filings // Computer Law Review International. 2024. V. 25. №. 2. pp. 41-52.

16. Rahman A., Subriadi A.P. Software as a service (SaaS) adoption factors: individual and organizational perspective // 2nd International Conference on Information Technology and Education (ICIT&E). IEEE, 2022. pp. 31-36.

17. Pan Y. et al. A systematic literature review of android malware detection using static analysis // IEEE Access. 2020. V. 8. P. 116363-116379.

18. Debas E., Alhumam N., Riad K. Unveiling the Dynamic Landscape of Malware Sandboxing: A Comprehensive Review // Preprints. 2023. URL: preprints.org/manuscript/202312.1009/v1.

19. Lebbie M., Prabhu S.R., Agrawal A.K. Comparative Analysis of Dynamic Malware Analysis Tools // Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021. Singapore: Springer Singapore, 2022. pp. 359-368.

20. Демина Р.Ю., Ажмухамедов И.М. Методика формирования обучающего множества при использовании статических антивирусных методов эвристического анализа // Инженерный вестник Дона. 2015. №. 3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3265.

21. Dutta N. et al. Introduction to malware analysis // Cyber Security: Issues and Current Trends. 2022. pp. 129-141.

22. Moser A., Kruegel C., Kirda E. Limits of static analysis for malware detection // Twenty-third annual computer security applications conference (ACSAC 2007). IEEE, 2007. pp. 421-430.

23. Tahir R. A study on malware and malware detection techniques // International Journal of Education and Management Engineering. 2018. V. 8. №. 2. URL: mecs-press.org/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf.

24. Talukder S. Tools and techniques for malware detection and analysis // arXiv preprint arXiv: 2002.06819. 2020. URL: researchgate.net/publication/339301928_Tools_and_Techniques_for_Malware_Detection_and_Analysis.

25. Aslan Ö.A., Samet R. A comprehensive review on malware detection approaches // IEEE access. 2020. V. 8. pp. 6249-6271.

26. Li N. et al. A survey on feature extraction methods of heuristic malware detection // Journal of Physics: Conference Series. IOP Publishing, 2021. V. 1757. №. 1. URL: iopscience.iop.org/article/10.1088/1742-6596/1757/1/012071/meta.

27. Sharma P. et al. A comparative analysis of malware anomaly detection // Advances in Computer, Communication and Computational Sciences: Proceedings of IC4S 2019. Springer Singapore, 2021. pp. 35-44.

28. Ngamwitroj S., Limthanmaphon B. Adaptive Android malware signature detection // Proceedings of the 2018 International Conference on Communication Engineering and Technology. 2018. pp. 22-25.

29. Ojugo A., Eboka A.O. Signature-based malware detection using approximate Boyer Moore string matching algorithm // International Journal of Mathematical Sciences and Computing. 2019. V. 5. №. 3. pp. 49-62.

30. Punyasiri D.L.S. Signature & Behavior Based Malware Detection. 2023. URL: researchgate.net/profile/Sathishka_Punyasiri/publication/374386435_Signature_Behavior_Based_Malware_Detection/links/651b90dbb0df2f20a20ac28a/Signature-Behavior-Based-Malware-Detection.pdf.

31. Kozachok A.V., Kozachok V.I. Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis // Journal of computer virology and hacking techniques. 2018. V. 14. №. 3. pp. 225-231.

32. Alkhateeb E.M., Stamp M. A dynamic heuristic method for detecting packed malware using naive bayes // 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA). IEEE, 2019. pp. 1-6.

33. Khodamoradi P. et al. Heuristic metamorphic malware detection based on statistics of assembly instructions using classification algorithms // 2015 18th

CSI International Symposium on Computer Architecture and Digital Systems (CADS). IEEE, 2015. pp. 1-6.

34. Treadwell S., Zhou M. A heuristic approach for detection of obfuscated malware // 2009 IEEE International Conference on Intelligence and Security Informatics. IEEE, 2009. pp. 291-299.

35. Zakeri M., Faraji Daneshgar F., Abbaspour M. A static heuristic approach to detecting malware targets // Security and Communication Networks. 2015. V. 8. №. 17. pp. 3015-3027.

36. Yunmar R.A. et al. Hybrid Android Malware Detection: A Review of Heuristic-Based Approach // IEEE Access. 2024. V. 12. pp. 41255-41286.

37. Shah I.A. et al. HeuCrip: A malware detection approach for internet of battlefield things // Cluster Computing. 2023. V. 26. №. 2. pp. 977-992.

38. Gyunka B.A., Abikoye O.C., Adekunle A.S. Anomaly Android malware detection: A comparative analysis of six classifiers // Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24–27, 2020, Revised Selected Papers. Cham : Springer International Publishing, 2021. pp. 145-157.

39. Tang A., Sethumadhavan S., Stolfo S.J. Unsupervised anomaly-based malware detection using hardware features // Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17-19, 2014. Proceedings 17. Springer International Publishing, 2014. pp. 109-129.

40. Tajoddin A., Abadi M. RAMD: registry-based anomaly malware detection using one-class ensemble classifiers // Applied Intelligence. 2019. V. 49. pp. 2641-2658.

41. Garg V., Yadav R.K. Malware detection based on API calls frequency // 2019 4th International Conference on Information Systems and Computer Networks (ISCON). IEEE, 2019. pp. 400-404.

42. Mushtaq E., Zameer A., Nasir R. Knacks of a hybrid anomaly detection model using deep auto-encoder driven gated recurrent unit // Computer Networks. 2023. V. 226. URL: sciencedirect.com/science/article/abs/pii/S1389128623001263.

43. Antić J. et al. Runtime security monitoring by an interplay between rule matching and deep learning-based anomaly detection on logs // 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN). IEEE, 2023. pp. 1-5.

44. Mitchell T.M. Machine learning. McGraw-Hill, New York, 1997, V. 1. №. 9.

45. Schultz M.G. et al. Data mining methods for detection of new malicious executables // Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001. IEEE, 2000. pp. 38-49.

46. Moskovitch R. et al. Unknown malcode detection and the imbalance problem // Journal in computer virology. 2009. V. 5. pp. 295-308.

47. Moskovitch R. et al. Unknown malcode detection using opcode representation // Intelligence and Security Informatics: First European Conference, EuroISI 2008, Esbjerg, Denmark, December 3-5, 2008. Proceedings. Springer Berlin Heidelberg, 2008. pp. 204-215.

48. Firdausi I. et al. Analysis of machine learning techniques used in behavior-based malware detection // 2010 second international conference on advances in computing, control, and telecommunication technologies. IEEE, 2010. pp. 201-203.

49. Santos I., Nieves J., Bringas P.G. Semi-supervised learning for unknown malware detection // International Symposium on Distributed Computing and Artificial Intelligence. Springer Berlin Heidelberg, 2011. pp. 415-422.

50. Rieck K. et al. Automatic analysis of malware behavior using machine learning // Journal of computer security. 2011. V. 19. №. 4. pp. 639-668.

51. Sundarkumar G.G. et al. Malware detection via API calls, topic models and machine learning // 2015 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, 2015. pp. 1212-1217.

52. Naval S. et al. Employing program semantics for malware detection // IEEE Transactions on Information Forensics and Security. 2015. V. 10. №. 12. pp. 2591-2604.

53. Wu S. et al. Effective detection of android malware based on the usage of data flow APIs and machine learning // Information and software technology. 2016. V. 75. pp. 17-25.

54. Anderson H.S., Roth P. Ember: an open dataset for training static pe malware machine learning models // arXiv preprint arXiv: 1804.04637. 2018. URL: arxiv.org/abs/1804.04637.

55. Sharma S., Rama Krishna C., Sahay S.K. Detection of advanced malware by machine learning techniques // Soft Computing: Theories and Applications: Proceedings of SoCTA 2017. Springer Singapore, 2019. pp. 333-342.

56. Li J. et al. Significant permission identification for machine-learning-based android malware detection // IEEE Transactions on Industrial Informatics. 2018. V. 14. №. 7. pp. 3216-3225.

57. Liu C. et al. MOBIPCR: Efficient, accurate, and strict ML-based mobile malware detection // Future Generation Computer Systems. 2023. V. 144. pp. 140-150.

58. Akhtar M.S., Feng T. Evaluation of machine learning algorithms for malware detection // Sensors. 2023. V. 23. №. 2. P. 946. URL: mdpi.com/1424-8220/23/2/946.

59. Zhang A. et al. Dive into deep learning. Cambridge University Press, 2023. 548 p.

60. Basole S., Di Troia F., Stamp M. Multifamily malware models // Journal of Computer Virology and Hacking Techniques. 2020. V. 16. pp. 79-92.

61. Singh T. et al. Support vector machines and malware detection // Journal of Computer Virology and Hacking Techniques. 2016. V. 12. pp. 203-212.

62. Annadatha A., Stamp M. Image spam analysis and detection // Journal of Computer Virology and Hacking Techniques. 2018. V. 14. pp. 39-52.

63. Chavda A. et al. Support vector machines for image spam analysis // 15th International Joint Conference on e-Business and Telecommunications. 2018. V. 1. pp. 431-441.

64. Sharmin T. et al. Convolutional neural networks for image spam detection // Information Security Journal: A Global Perspective. 2020. V. 29. №. 3. pp. 103-117.

65. Bhodia N. et al. Transfer learning for image-based malware classification // arXiv preprint arXiv: 1903.11551. 2019. URL: arxiv.org/abs/1903.11551.

66. Yajamanam S. et al. Deep Learning versus Gist Descriptors for Image-based Malware Classification // Icissp. 2018. pp. 553-561.

67. Khan S.H. et al. A new deep boosted CNN and ensemble learning based IoT malware detection // Computers & Security. 2023. V. 133. P. 103385.

68. Dhanya K.A. et al. Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN // IEEE Systems Journal. 2023. V. 17. №. 2. pp. 2756-2766.

69. Ullah F. et al. NMal-Droid: network-based android malware detection system using transfer learning and CNN-BiGRU ensemble // Wireless Networks. 2023. pp. 1-22.

70. Tang M., Qian Q. Dynamic API call sequence visualisation for malware classification // IET Information Security. 2019. V. 13. №. 4. pp. 367-377.

71. Jain M., Andreopoulos W., Stamp M. Convolutional neural networks and extreme learning machines for malware classification // Journal of Computer Virology and Hacking Techniques. 2020. V. 16. pp. 229-244.

72. Dong S., Shu L., Nie S. Android Malware Detection Method Based on CNN and DNN Bybrid Mechanism // IEEE Transactions on Industrial Informatics. 2024. URL: ieeexplore.ieee.org/abstract/document/10444689.

73. Zhang W. et al. Exploring feature extraction and ELM in malware detection for android devices // Advances in Neural Networks–ISNN 2015: 12th International Symposium on Neural Networks, ISNN 2015, Jeju, South Korea, October 15-18, 2015, Proceedings 12. Springer International Publishing, 2015. pp. 489-498.

74. Shamshirband S., Chronopoulos A. T. A new malware detection system using a high performance-ELM method // Proceedings of the 23rd international database applications & engineering symposium. 2019. pp. 1-10.

75. Jahromi A.N. et al. An improved two-hidden-layer extreme learning machine for malware hunting // Computers & Security. 2020. V. 89. URL: sciencedirect.com/science/article/abs/pii/S0167404819301981.

76. Moraga L.I. et al. Detection of obfuscated malware by engineering memory functions applying ELM // 2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI). IEEE, 2023. pp. 1-6.

77. Reddy V.S.K. et al. MDC-Net: Intelligent Malware Detection and Classification using Extreme Learning Machine // 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2023. pp. 1590-1594.

78. Lu R. Malware detection with lstm using opcode language // arXiv preprint arXiv:1906.04593. 2019. URL: arxiv.org/abs/1906.04593.

79. Avci C., Tekinerdogan B., Catal C. Analyzing the performance of long short- term memory architectures for malware detection models // Concurrency

and Computation: Practice and Experience. 2023. V. 35. №. 6. URL: onlinelibrary.wiley.com/doi/full/10.1002/cpe.7581.

80. Maniriho P., Mahmood A.N., Chowdhury M.J.M. API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques // Journal of Network and Computer Applications. 2023. V. 218. URL: sciencedirect.com/science/article/pii/S1084804523001236.

81. Bhardwaj S., Dave M. Integrating a Rule-Based Approach to Malware Detection with an LSTM-Based Feature Selection Technique // SN Computer Science. 2023. V. 4. №. 6. URL: link.springer.com/article/10.1007/s42979-023-02177-2.

82. Ramamoorthy J., Shashidhar N.K., Zhou B. Anomaly based malware threat detection on Linux Systems // 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2023. pp. 1744-1750.

83. Kumar M. et al. LSTM-based Approach for Android Malware Detection // Procedia Computer Science. 2023. V. 230. pp. 679-687.

84. Jebin Bose S., Kalaiselvi R. An optimal detection of android malware using dynamic attention-based LSTM classifier // Journal of Intelligent & Fuzzy Systems. 2023. V. 44. №. 1. pp. 1425-1438.

85. Abdullah M.A. et al. HCL-Classifier: CNN and LSTM based hybrid malware classifier for Internet of Things (IoT) // Future Generation Computer Systems. 2023. V. 142. pp. 41-58.

86. Devi R.A., Arunachalam A.R. Enhancement of IoT device security using an Improved Elliptic Curve Cryptography algorithm and malware detection utilizing deep LSTM // High-Confidence Computing. 2023. V. 3. №. 2. URL: sciencedirect.com/science/article/pii/S2667295223000156.

87. Athiwaratkun B., Stokes J.W. Malware classification with LSTM and GRU language models and a character-level CNN // 2017 IEEE international

conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017. pp. 2482-2486.

88. Hu W., Tan Y. Black-box attacks against RNN based malware detection algorithms // arXiv preprint arXiv: 1705.08131. 2017. URL: arxiv.org/abs/1705.08131.

89. Jeon J. et al. Early prediction of ransomware API calls behaviour based on GRU-TCN in healthcare IoT // Connection Science. 2023. V. 35. №. 1. URL: tandfonline.com/doi/full/10.1080/09540091.2023.2233716.

90. Wu Y. et al. DeepCatra: Learning flow- and graph- based behaviours for Android malware detection // IET Information Security. 2023. V. 17. №. 1. pp. 118-130.

91. Yagcioglu S. et al. Detecting cybersecurity events from noisy short text // arXiv preprint arXiv: 1904.05054. 2019. URL: arxiv.org/abs/1904.05054.

92. Zhao X. et al. A survey on cybersecurity knowledge graph construction // Computers & Security. 2023. V. 136. URL: sciencedirect.com/science/article/abs/pii/S0167404823004340.

93. Challagundla B.C., Gogireddy Y.R., Peddavenkatagari C.R. Efficient CAPTCHA Image Recognition Using Convolutional Neural Networks and Long Short-Term Memory Networks // International Journal of Scientific Research in Engineering and Management (IJSREM). 2024. V. 8. URL: academia.edu/download/113099762/Efficient_CAPTCHA_Image_Recognition_Using_Convolutional_Neural_Networks_and_Long_Short_Term_Memory.pdf.

94. Hnamte V. et al. A novel two-stage deep learning model for network intrusion detection: LSTM-AE // IEEE Access. 2023. URL: ieeexplore.ieee.org/document/10101759.

95. Shanmuganathan V., Suresh A. LSTM-Markov based efficient anomaly detection algorithm for IoT environment // Applied Soft Computing. 2023. V. 136. URL: sciencedirect.com/science/article/abs/pii/S1568494623000728.

96. Nahhas L. et al. Android Malware Detection Using ResNet-50 Stacking // Computers, Materials & Continua. 2023. V. 74. №. 2.

97. Al-Khater W., Al-Madeed S. Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware // Alexandria Engineering Journal. 2024. V. 89. pp. 39-52.

98. Fu X. et al. A hybrid approach for Android malware detection using improved multi-scale convolutional neural networks and residual networks // Expert Systems with Applications. 2024. V. 249. URL: sciencedirect.com/science/article/abs/pii/S0957417424005414.

99. Zhu H. et al. Android malware detection based on multi-head squeeze-and-excitation residual network // Expert Systems with Applications. 2023. V. 212. URL: sciencedirect.com/science/article/abs/pii/S095741742201733X.

100. Wu P., Guo H., Moustafa N. Pelican: A deep residual network for network intrusion detection // 2020 50th annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W). IEEE, 2020. pp. 55-62.

101. Kumar G.S.C. et al. Deep residual convolutional neural Network: An efficient technique for intrusion detection system // Expert Systems with Applications. 2024. V. 238. URL: sciencedirect.com/science/article/abs/pii/S0957417423024144.

102. Duan X., Fu Y., Wang K. Network traffic anomaly detection method based on multi-scale residual classifier // Computer Communications. 2023. V. 198. pp. 206-216.

103. Rao Y.N., Suresh Babu K. An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset // Sensors. 2023. V. 23. №. 1. URL: mdpi.com/1424-8220/23/1/550.

104. Malhotra V., Potika K., Stamp M. A Comparison of Graph Neural Networks for Malware Classification // arXiv preprint arXiv:2303.12812. 2023. URL: arxiv.org/abs/2303.12812.

105. Warmsley D. et al. A Survey of Explainable Graph Neural Networks for Cyber Malware Analysis // 2022 IEEE International Conference on Big Data (Big Data). IEEE, 2022. pp. 2932-2939.

106. Feng P. et al. Android malware detection via graph representation learning // Mobile Information Systems. 2021. V. 2021. pp. 1-14.

107. Hu W., Tan Y. Generating adversarial malware examples for black-box attacks based on GAN // Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore: Springer Nature Singapore, 2023. pp. 409-423.

108. Kim J.Y., Bu S.J., Cho S.B. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders // Information Sciences. 2018. V. 460. pp. 83-102.

109. Peppes N. et al. The effectiveness of zero-day attacks data samples generated via GANs on deep learning classifiers // Sensors. 2023. V. 23. №. 2. URL: mdpi.com/1424-8220/23/2/900.

110. Ahmad R. et al. Zero-day attack detection: a systematic literature review // Artificial Intelligence Review. 2023. V. 56. №. 10. pp. 10733-10811.

111. Rigaki M., Garcia S. Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection // 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018. pp. 70-75.

References

1. Lallie H.S. et al. Computers & security. 2021. V. 105. URL: sciencedirect.com/science/article/pii/ S0167404821000729.

2. Mirza Q.K.A. et al. 8th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2021. P. 58-65.

3. Anikin I.V., Isjandavletova Ja.M. Inzhenernyj vestnik Dona. 2023. №. 4 URL: ivdon.ru/ru/magazine/archive/n4y2023/8346.

4. Alwashali A.A. M.A., Abd Rahman N.A., Ismail N. 14th International Conference on Developments in eSystems Engineering (DeSE). IEEE, 2021. P. 92-96.

5. Alkhalil Z. et al. Frontiers in Computer Science. 2021. V. 3. URL: frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2021.563060.

6. Ramirez R. Behavioral Characterization of Attacks on the Remote Desktop Protocol. 2022. URL: apps.dtic.mil/sti/citations/trecms/AD1201693.

7. Oz H. et al. 32nd USENIX Security Symposium (USENIX Security 23). 2023. P. 7073-7090.

8. Razaulla S. et al. IEEE Access. 2023. V. 11. URL: ieeexplore.ieee.org/abstract/document/10105244.

9. Muralidharan T. et al. ACM Computing Surveys. 2022. V. 55. №. 5. P. 1-45.

10. Badhwar R. The CISO's Next Frontier: AI, Post-Quantum Cryptography and Advanced Security Paradigms. Cham: Springer International Publishing, 2021. P. 279-285.

11. Murali R., Thangavel P., Velayutham C.S. Expert Systems with Applications. 2023. V. 226. URL: sciencedirect.com/science/article/abs/pii/S0957417423005948.

12. Kara I. Expert Systems with Applications. 2023. V. 214. URL: sciencedirect.com/science/article/abs/pii/ S0957417422021510.

13. Zhang X. et al. Forensic Science International: Digital Investigation. 2021. V. 39. URL: sciencedirect.com/science/article/pii/S2666281721002031.

14. Damjanović D.Z. Vojnotehnički glasnik/Military Technical Courier. 2022. V. 70. №. 3. P. 734-755.

15. Beardwood J. Computer Law Review International. 2024. V. 25. №. 2. P. 41-52.

16. Rahman A., Subriadi A.P. 2nd International Conference on Information Technology and Education (ICIT&E). IEEE, 2022. P. 31-36.

17. Pan Y. et al. IEEE Access. 2020. V. 8. P. 116363-116379.

18. Debas E., Alhumam N., Riad K. Preprints. 2023. URL: preprints.org/manuscript/202312.1009/v1.

19. Lebbie M., Prabhu S.R., Agrawal A.K. Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021. Singapore: Springer Singapore, 2022. P. 359-368.

20. Demina R.Ju, Azhmuhamedov I.M. Inzenernyj vestnik Dona. 2015. №. 3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3265.

21. Dutta N. et al. Cyber Security: Issues and Current Trends. 2022. P. 129-141.

22. Moser A., Kruegel C., Kirda E. Twenty-third annual computer security applications conference (ACSAC 2007). IEEE, 2007. P. 421-430.

23. Tahir R. International Journal of Education and Management Engineering. 2018. V. 8. №. 2. URL: mecs-press.org/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf.

24. Talukder S. arXiv preprint arXiv:2002.06819. 2020. URL: researchgate.net/publication/339301928_Tools_and_Techniques_for_Malware_Detection_and_Analysis.

25. Aslan Ö.A., Samet R. IEEE access. 2020. V. 8. P. 6249-6271.

26. Li N. et al. Journal of Physics: Conference Series. IOP Publishing, 2021. V. 1757. №. 1. URL: iopscience.iop.org/article/10.1088/1742-6596/1757/1/012071/meta.

27. Sharma P. et al. Advances in Computer, Communication and Computational Sciences: Proceedings of IC4S 2019. Springer Singapore, 2021. P. 35-44.

28. Ngamwitroj S., Limthanmaphon B. Proceedings of the 2018 International Conference on Communication Engineering and Technology. 2018. P. 22-25.

29. Ojugo A., Eboka A.O. International Journal of Mathematical Sciences and Computing. 2019. V. 5. №. 3. P. 49-62.

30. Punyasiri D.L.S. ResearchGate. 2023. URL: researchgate.net/profile/Sathishka_Punyasiri/publication/374386435_Signature_Behavior_Based_Malware_Detection/links/651b90dbb0df2f20a20ac28a/Signature-Behavior-Based-Malware-Detection.pdf.

31. Kozachok A.V., Kozachok V.I. Journal of computer virology and hacking techniques. 2018. V. 14. №. 3. P. 225-231.

32. Alkhateeb E.M., Stamp M. 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA). IEEE, 2019. P. 1-6.

33. Khodamoradi P. et al. 2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS). IEEE, 2015. P. 1-6.

34. Treadwell S., Zhou M. 2009 IEEE International Conference on Intelligence and Security Informatics. IEEE, 2009. P. 291-299.

35. Zakeri M., Faraji Daneshgar F., Abbaspour M. Security and Communication Networks. 2015. V. 8. №. 17. P. 3015-3027.

36. Yunmar R.A. et al. IEEE Access. 2024. V. 12. P. 41255-41286.

37. Shah I.A. et al. Cluster Computing. 2023. V. 26. №. 2. P. 977-992.

38. Gyunka B.A., Abikoye O.C., Adekunle A.S. Information and Communication Technology and Applications: Third International Conference, ICTA 2020, Minna, Nigeria, November 24–27, 2020, Revised Selected Papers. Cham : Springer International Publishing, 2021. P. 145-157.

39. Tang A., Sethumadhavan S., Stolfo S.J. Research in Attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden,

September 17-19, 2014. Proceedings 17. Springer International Publishing, 2014. P. 109-129.

40. Tajoddin A., Abadi M. Applied Intelligence. 2019. V. 49. P. 2641-2658.

41. Garg V., Yadav R.K. 2019 4th International Conference on Information Systems and Computer Networks (ISCON). IEEE, 2019. P. 400-404.

42. Mushtaq E., Zameer A., Nasir R. Computer Networks. 2023. V. 226. URL: sciencedirect.com/science/article/abs/pii/S1389128623001263.

43. Antić J. et al. 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN). IEEE, 2023. P. 1-5.

44. Mitchell T.M. Machine learning. McGraw-Hill, New York, 1997, V. 1. №. 9.

45. Schultz M.G. et al. Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001. IEEE, 2000. P. 38-49.

46. Moskovitch R. et al. Journal in computer virology. 2009. V. 5. P. 295-308.

47. Moskovitch R. et al. Intelligence and Security Informatics: First European Conference, EuroISI 2008, Esbjerg, Denmark, December 3-5, 2008. Proceedings. Springer Berlin Heidelberg, 2008. P. 204-215.

48. Firdausi I. et al. 2010 second international conference on advances in computing, control, and telecommunication technologies. IEEE, 2010. P. 201-203.

49. Santos I., Nieves J., Bringas P.G. International Symposium on Distributed Computing and Artificial Intelligence. Springer Berlin Heidelberg, 2011. P. 415-422.

50. Rieck K. et al. Journal of computer security. 2011. V. 19. №. 4. P. 639-668.

51. Sundarkumar G.G. et al. 2015 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, 2015. P. 1212-1217.

52. Naval S. et al. IEEE Transactions on Information Forensics and Security. 2015. V. 10. №. 12. P. 2591-2604.

53. Wu S. et al. Information and software technology. 2016. V. 75. P. 17-25.

54. Anderson H.S., Roth P. arXiv preprint arXiv:1804.04637. 2018. URL: arxiv.org/abs/1804.04637.

55. Sharma S., Rama Krishna C., Sahay S.K. Soft Computing: Theories and Applications: Proceedings of SoCTA 2017. Springer Singapore, 2019. P. 333-342.

56. Li J. et al. IEEE Transactions on Industrial Informatics. 2018. V. 14. №. 7. P. 3216-3225.

57. Liu C. et al. Future Generation Computer Systems. 2023. V. 144. P. 140-150.

58. Akhtar M.S., Feng T. Sensors. 2023. V. 23. №. 2. P. 946. URL: mdpi.com/1424-8220/23/2/946.

59. Zhang A. et al. Dive into deep learning. Cambridge University Press, 2023. 548 p.

60. Basole S., Di Troia F., Stamp M. Journal of Computer Virology and Hacking Techniques. 2020. V. 16. P. 79-92.

61. Singh T. et al. Journal of Computer Virology and Hacking Techniques. 2016. V. 12. P. 203-212.

62. Annadatha A., Stamp M. Journal of Computer Virology and Hacking Techniques. 2018. V. 14. P. 39-52.

63. Chavda A. et al. 15th International Joint Conference on e-Business and Telecommunications. 2018. V. 1. P. 431-441.

64. Sharmin T. et al. Information Security Journal: A Global Perspective. 2020. V. 29. №. 3. P. 103-117.

65. Bhodia N. et al. arXiv preprint arXiv:1903.11551. 2019. URL: arxiv.org/abs/1903.11551.

66. Yajamanam S. et al. Icissp. 2018. P. 553-561.

67. Khan S.H. et al. Computers & Security. 2023. V. 133. P. 103385.

68. Dhanya K.A. et al. IEEE Systems Journal. 2023. V. 17. №. 2. P. 2756-2766.

69. Ullah F. et al. Wireless Networks. 2023. P. 1-22.

70. Tang M., Qian Q. IET Information Security. 2019. V. 13. №. 4. P. 367-377.

71. Jain M., Andreopoulos W., Stamp M. Journal of Computer Virology and Hacking Techniques. 2020. V. 16. P. 229-244.

72. Dong S., Shu L., Nie S. IEEE Transactions on Industrial Informatics. 2024. URL: ieeexplore.ieee.org/abstract/document/10444689.

73. Zhang W. et al. Advances in Neural Networks–ISNN 2015: 12th International Symposium on Neural Networks, ISNN 2015, Jeju, South Korea, October 15-18, 2015, Proceedings 12. Springer International Publishing, 2015. P. 489-498.

74. Shamshirband S., Chronopoulos A. T. Proceedings of the 23rd international database applications & engineering symposium. 2019. P. 1-10.

75. Jahromi A.N. et al. Computers & Security. 2020. V. 89. URL: sciencedirect.com/science/article/abs/pii/S0167404819301981.

76. Moraga L.I. et al. 2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI). IEEE, 2023. P. 1-6.

77. Reddy V.S.K. et al. 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2023. P. 1590-1594.

78. Lu R. arXiv preprint arXiv:1906.04593. 2019. URL: arxiv.org/abs/1906.04593.

79. Avci C., Tekinerdogan B., Catal C. Concurrency and Computation: Practice and Experience. 2023. V. 35. №. 6. URL: onlinelibrary.wiley.com/doi/full/10.1002/cpe.7581.

80. Maniriho P., Mahmood A.N., Chowdhury M.J.M. Journal of Network and Computer Applications. 2023. V. 218. URL: sciencedirect.com/science/article/pii/S1084804523001236.

81. Bhardwaj S., Dave M. SN Computer Science. 2023. V. 4. №. 6. URL: link.springer.com/article/10.1007/s42979-023-02177-2.

82. Ramamoorthy J., Shashidhar N.K., Zhou B. 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2023. P. 1744-1750.

83. Kumar M. et al. Procedia Computer Science. 2023. V. 230. P. 679-687.

84. Jebin Bose S., Kalaiselvi R. Journal of Intelligent & Fuzzy Systems. 2023. V. 44. №. 1. P. 1425-1438.

85. Abdullah M.A. et al. Future Generation Computer Systems. 2023. V. 142. P. 41-58.

86. Devi R.A., Arunachalam A.R. High-Confidence Computing. 2023. V. 3. №. 2. URL: sciencedirect.com/science/article/pii/S2667295223000156.

87. Athiwaratkun B., Stokes J.W. 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017. P. 2482-2486.

88. Hu W., Tan Y. arXiv preprint arXiv: 1705.08131. 2017. URL: arxiv.org/abs/1705.08131.

89. Jeon J. et al. Connection Science. 2023. V. 35. №. 1. URL: tandfonline.com/doi/full/10.1080/09540091.2023.2233716.

90. Wu Y. et al. IET Information Security. 2023. V. 17. №. 1. P. 118-130.

91. Yagcioglu S. et al. arXiv preprint arXiv: 1904.05054. 2019. URL: arxiv.org/abs/1904.05054.

92. Zhao X. et al. Computers & Security. 2023. V. 136. URL: sciencedirect.com/science/article/abs/pii/S0167404823004340.

93. Challagundla B.C., Gogireddy Y.R., Peddavenkatagari C.R. International Journal of Scientific Research in Engineering and Management (IJSREM). 2024. V. 8. URL: academia.edu/download/113099762/Efficient_CAPTCHA_Image_Recognition_Using_Convolutional_Neural_Networks_and_Long_Short_Term_Memory.pdf.

94. Hnamte V. et al. IEEE Access. 2023. URL: ieeexplore.ieee.org/document/10101759.

95. Shanmuganathan V., Suresh A. Applied Soft Computing. 2023. V. 136. URL: sciencedirect.com/science/article/abs/pii/S1568494623000728.

96. Nahhas L. et al. Computers, Materials & Continua. 2023. V. 74. №. 2.

97. Al-Khater W., Al-Madeed S. Alexandria Engineering Journal. 2024. V. 89. P. 39-52.

98. Fu X. et al. Expert Systems with Applications. 2024. V. 249. URL: sciencedirect.com/science/article/abs/pii/S0957417424005414.

99. Zhu H. et al. Expert Systems with Applications. 2023. V. 212. URL: sciencedirect.com/science/article/abs/pii/S095741742201733X.

100. Wu P., Guo H., Moustafa N. 2020 50th annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W). IEEE, 2020. P. 55-62.

101. Kumar G.S.C. et al. Expert Systems with Applications. 2024. V. 238. URL: sciencedirect.com/science/article/abs/pii/S0957417423024144.

102. Duan X., Fu Y., Wang K. Computer Communications. 2023. V. 198. P. 206-216.

103. Rao Y.N., Suresh Babu K. Sensors. 2023. V. 23. №. 1. URL: mdpi.com/1424-8220/23/1/550.

104. Malhotra V., Potika K., Stamp M. arXiv preprint arXiv:2303.12812. 2023. URL: arxiv.org/abs/2303.12812.

105. Warmsley D. et al. 2022 IEEE International Conference on Big Data (Big Data). IEEE, 2022. P. 2932-2939.

106. Feng P. et al. Mobile Information Systems. 2021. V. 2021. P. 1-14.

107. Hu W., Tan Y. Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore : Springer Nature Singapore, 2023. P. 409-423.

108. Kim J.Y., Bu S.J., Cho S.B. Information Sciences. 2018. V. 460. P. 83-102.

109. Peppes N. et al. Sensors. 2023. V. 23. №. 2. URL: mdpi.com/1424-8220/23/2/900.

110. Ahmad R. et al. Artificial Intelligence Review. 2023. V. 56. №. 10. P. 10733-10811.

111. Rigaki M., Garcia S. 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018. P. 70-75.

Дата поступления: 14.07.2024

Дата публикации: 29.08.2024