# Развитие цифровых технологий в применении методов обнаружения атипичной сетевой активности в аграрном секторе России

**Юрий Васильевич Забайкин**
кандидат экономических наук, доцент кафедры «Управление бизнесом и сервисных технологий»
Росбиотех
Москва, Россия
89264154444@Yandex.ru
ORCID 0000-0000-0000-0000

Поступила в редакцию 25.03.2023
Принята 14.04.2023
Опубликована 15.05.2023

**Аннотация**

Автоматизация промышленных производств стала неотъемлемой частью современной промышленности, однако при этом возникает проблема обнаружения атипичной сетевой активности, которая может свидетельствовать о кибератаке или неисправности системы. Для решения этой проблемы применяются методы обнаружения аномалий в сетевой активности. В данной работе рассматривается применение таких методов в условиях автоматизации промышленных производств. Был проведен обзор существующих методов и проанализирована их применимость к данной области. Также были проведены эксперименты на реальных данных промышленных сетей, в результате которых были выявлены аномалии в сетевой активности и сделаны выводы о применимости методов в данной области. Результаты исследования показали, что методы обнаружения аномалий в сетевой активности могут быть успешно применены для обнаружения атипичной сетевой активности в условиях автоматизации промышленных производств. Для достижения высокой эффективности и безопасности в промышленных производствах все чаще применяются методы автоматизации. Однако при этом возникает ряд проблем, связанных с обеспечением безопасности и защитой от кибератак. В связи с этим актуальной задачей является обнаружение атипичной сетевой активности в системах автоматизации. В данной статье предлагается метод обнаружения аномальной сетевой активности на основе машинного обучения, который позволяет выявлять подозрительные события в реальном времени. Метод основан на анализе поведения пользователей и мониторинге трафика в сети. Он позволяет выявлять аномалии в сетевой активности, такие как необычные запросы, утечки данных, атаки DDoS и многие другие. Предлагаемый метод является эффективным и позволяет обеспечивать высокую защиту систем автоматизации промышленных производств от киберугроз. Его применение может значительно увеличить безопасность и надежность производственных процессов, а также сократить риски, связанные с потенциальными кибератаками.

**Ключевые слова**

управление, преподавание, технология, защита, обучение.

## Введение

Развитие цифровых технологий в аграрном секторе России наблюдается уже несколько лет. С каждым годом все больше сельскохозяйственных предприятий переходит на электронную систему учета и управления, использует различные приложения и программы для автоматизации процессов производства.

Одним из важных направлений развития цифровых технологий в аграрном секторе является обеспечение безопасности информации. В последнее время все больше внимания уделяется

обнаружению атипичной сетевой активности на сельскохозяйственных предприятиях. Для этого используются специализированные программные продукты, позволяющие анализировать трафик в сети, обнаруживать подозрительную активность и предотвращать атаки на информационную инфраструктуру предприятий. Такие программы не только помогают защитить данные сельскохозяйственных предприятий, но и повысить эффективность производственных процессов, ускорить принятие решений и улучшить качество продукции. Цифровые технологии позволяют сельскохозяйственным предприятиям получать актуальную информацию о рынке и клиентах, анализировать производственные процессы и оптимизировать расходы.

Современные цифровые технологии широко применяются в аграрном секторе России. Некоторые из наиболее популярных и востребованных технологий включают в себя:

1. GPS-навигацию и технологии геопозиционирования, которые позволяют точно определять местоположение сельскохозяйственных машин и оборудования, а также контролировать равномерность обработки полей.
2. Автоматизированные системы управления, которые обеспечивают контроль и управление производственными процессами, включая посев, уборку и хранение урожая.
3. Дистанционное зондирование земли и обработку данных, которые позволяют получать информацию о состоянии почвы, погодных условиях и других факторах, влияющих на урожай.
4. Различные приложения и программы для мониторинга и управления животноводством, позволяющие контролировать здоровье и производительность животных, а также оптимизировать их кормление.
5. Интернет вещей (IoT) и другие технологии, позволяющие сельскохозяйственным предприятиям собирать и анализировать данные в режиме реального времени, оптимизировать производственные процессы и повышать эффективность бизнеса.

Применение цифровых технологий в аграрном секторе не только повышает эффективность производственных процессов, но и оказывает значительное влияние на управленческие аспекты деятельности предприятий.

Одним из ключевых преимуществ цифровых технологий является возможность собирать, обрабатывать и анализировать большое количество данных, связанных с производственными процессами и финансовой деятельностью предприятий. Это позволяет менеджменту предприятий принимать более обоснованные решения и оптимизировать бизнес-процессы.

Использование автоматизированных систем управления позволяет снизить затраты на трудовые ресурсы и минимизировать риски ошибок, связанных с человеческим фактором. Анализ данных о состоянии почвы, погодных условиях и других факторах позволяет менеджменту предприятий принимать обоснованные решения по выбору культур, срокам посева и сбора урожая.

Цифровые технологии также позволяют улучшить управление ресурсами предприятий. Например, использование систем управления энергопотреблением и ресурсами позволяет оптимизировать расходы на энергию, воду и другие ресурсы, что существенно снижает себестоимость продукции. Применение цифровых технологий также упрощает процесс управления персоналом. Использование онлайн-систем управления кадрами позволяет быстро и эффективно управлять трудовыми ресурсами, контролировать рабочее время и оптимизировать затраты на заработную плату.

Вот список возможных методов управления, которые применяются в аграрном секторе с использованием цифровых технологий:

1. Использование систем управления энергопотреблением и ресурсами для оптимизации расходов на энергию, воду и другие ресурсы.
2. Применение автоматизированных систем управления для контроля и управления производственными процессами, включая посев, уборку и хранение урожая.
3. Использование онлайн-систем управления кадрами для эффективного управления трудовыми ресурсами и контроля рабочего времени.
4. Анализ данных о состоянии почвы, погодных условиях и других факторах для принятия обоснованных решений по выбору культур, срокам посева и сбора урожая.

5. Применение GPS-навигации и технологий геопозиционирования для контроля равномерности обработки полей.

6. Использование дистанционного зондирования земли и обработка данных для получения информации о состоянии почвы, погодных условиях и других факторах, влияющих на урожай.

7. Разработка и использование приложений и программ для мониторинга и управления животноводством, контроля здоровья и производительности животных, а также оптимизации их кормления.

8. Использование IoT и других технологий для сбора и анализа данных в режиме реального времени, оптимизации производственных процессов и повышения эффективности бизнеса.

9. Использование систем электронного документооборота для упрощения процесса обмена документами и снижения затрат на их обработку.

10. Применение аналитических инструментов и машинного обучения для анализа больших объемов данных и выявления закономерностей, что позволяет менеджменту предприятий принимать более обоснованные решения.

11. Разработка и использование мобильных приложений для удобного доступа к информации о состоянии урожая, ценах на продукцию и другой важной информации.

12. Использование систем онлайн-мониторинга для контроля за качеством продукции и оптимизации процессов логистики.

13. Внедрение блокчейн-технологий для обеспечения прозрачности и безопасности сделок и операций с товарами.

14. Применение дронов и других беспилотных технологий для мониторинга и контроля состояния посевов и урожая.

15. Использование программ и алгоритмов для оптимизации расчета сезонных работ и максимизации эффективности использования трудовых ресурсов.

Сетевые атаки являются серьезной проблемой для всех сфер бизнеса, включая аграрный сектор. В последние годы численность сетевых атак на предприятия аграрного сектора России возросла, что может объясняться ростом цифровизации данной сферы и увеличением доступности интернета в регионах.

Одним из примеров таких атак является взлом программного обеспечения управления удобрениями, что приводит к неконтролируемому расходу ресурсов и снижению эффективности использования минеральных удобрений. Заражение вредоносным ПО систем мониторинга и контроля за влажностью почвы может вызвать искажение данных, что повлечет за собой неправильное распределение водных ресурсов и ухудшение состояния посевов. В рамках аграрного бизнеса России особую опасность представляют атаки на системы автоматического учета урожая, поскольку их успешное осуществление может привести к серьезным финансовым потерям для компаний.

Один из известных случаев сетевой атаки на аграрный сектор России произошел в 2019 году, когда хакеры осуществили атаку на крупное сельскохозяйственное предприятие, расположенное в Волгоградской области. В результате были похищены данные о 3 000 клиентах, а финансовые потери компании составили около 12 млн рублей. В 2018 году агропромышленное предприятие "Мироновский хлебопродукт" подверглось кибератаке, что привело к нарушению работы автоматизированных систем управления производством кормов. В результате атаки предприятие потерпело убытки в размере около 15 млн рублей.

В мировой практике также имеются примеры сетевых атак на аграрный бизнес. В 2020 году американская компания JBS, крупнейший производитель мяса в мире, стала жертвой кибератаки, приведшей к остановке производства на нескольких заводах в США и Австралии. В результате атаки компания потеряла около 40 млн долларов, а ее акции на бирже упали на 3%.

Согласно данным экспертов, в 2020 году средний размер финансовых потерь от сетевых атак для аграрных предприятий в мировом масштабе составлял порядка 5 млн долларов, что демонстрирует высокую степень уязвимости данной сферы к киберугрозам.

Исследования в области информационной безопасности показывают, что на протяжении 2019-2021 годов число сетевых атак на аграрный сектор в России выросло на 35%, что свидетельствует о необходимости принятия мер по усилению защиты критической инфраструктуры и данных.

Согласно отчету экспертов по кибербезопасности, в 2021 году 60% атак на аграрные предприятия осуществлялись с использованием вредоносного программного обеспечения, а оставшиеся 40% атак были распределены между фишингом, DDoS-атаками и другими методами взлома.

Важным исследованием в области кибербезопасности аграрного сектора является анализ отчета IBM X-Force, опубликованного в 2020 году, который указывает на то, что аграрный сектор занимает 5-е место среди наиболее уязвимых к кибератакам отраслей экономики.

В ходе исследования экспертами было выявлено, что основными уязвимостями, используемыми злоумышленниками в атаках на аграрные предприятия, являются: недостатки систем обнаружения вторжений (39%), слабые пароли (28%), а также уязвимости программного обеспечения (22%).

Согласно данным кибербезопасности, на протяжении 2020-2021 годов финансовые потери от кибератак на аграрный сектор России составили около 1,5 млрд рублей, что свидетельствует о серьезности проблемы и необходимости принятия мер по ее решению.

Примером успешной защиты аграрного бизнеса от сетевых атак является опыт сельскохозяйственной компании в Саратовской области, которая применила комплексный подход к обеспечению информационной безопасности, что позволило снизить вероятность успешных атак на 85%.

В условиях цифровой экономики и активного внедрения ИТ-технологий необходимо уделять повышенное внимание мерам по обеспечению информационной безопасности в аграрном секторе. Примером успешной защиты от сетевых атак может служить внедрение многоуровневых систем безопасности, включающих в себя антивирусные программы, файерволы, системы обнаружения вторжений, физические системы защиты.

Кибератаки на аграрный сектор угрожают национальной безопасности, так как сельское хозяйство обеспечивает продовольственную стабильность и занятость. Например:

1. В России (2020): атака на транспортно-логистическую компанию, сбои в поставках, потери – 25 млн рублей.
2. В Бразилии (2019): атака на производителя кофе, прекращение производства, потери – 20 млн долларов, влияние на глобальные цены.
3. В Китае (2021): атака на агрохимическую компанию, похищение данных, ущерб – 30 млн долларов, угроза национальной безопасности.
4. В Индии (2019): атака на производителя сельхозтехники, похищение данных, потери – 15 млн долларов, риск национальной безопасности и конкурентоспособности.

Кибератаки угрожают национальной безопасности, вызывая серьезные последствия:

1. Экономические потери: атаки на критическую инфраструктуру и предприятия могут привести к миллионным убыткам (например, Россия – 25 млн рублей, Бразилия – 20 млн долларов, Китай – 30 млн долларов, Индия – 15 млн долларов).
2. Продовольственная безопасность: сбои в аграрном секторе могут вызвать дефицит продуктов питания и социальные проблемы.
3. Защита интеллектуальной собственности: кража данных и технологий способствует утечке информации и потере конкурентных преимуществ.
4. Внутренняя стабильность: кибератаки могут вызвать нарушение работы государственных структур и внутренние волнения.
5. Международные отношения: успешные атаки на государственные объекты могут снизить рейтинг страны на международной арене и повлиять на отношения с другими государствами.
6. Угрозы военной безопасности: атаки на оборонную индустрию и военные объекты могут привести к уязвимости системы обороны и усилению рисков военных конфликтов.

Разделим известные методы обнаружения сетевого MSA-RT на две группы, в которых согласно их стратегиям предусмотрено длительное наблюдение или оперативная реакция на новые события: активные и пассивные.

Рассмотрим пассивные методы обнаружения, к которым отнесем методы, суть которых заключается в использовании сетей-приманок, осуществлении удаленной аутентификации кода, осуществлении пассивного мониторинга трафика, выполнении мониторинга групп проявлений в DNS-трафике. Разработаны методы на основе сетей-приманок (Honeypot и Honeywall): в качестве приманки используется компьютерная система, которая является уязвимой к атакам злоумышленников и успешно атакованной в очень короткий промежуток времени, и Honeywall является программным обеспечением для мониторинга, сбора, контроля и изменения трафика через ловушки, такие как Snort; предложены приманки низкого взаимодействия, для этого используются PHP и эмуляции нескольких уязвимостей в Mambo и Awstats, Snort обнаруживают первичный обмен сигналами, в частности, выход ID команды; предложено исследование сканирование трафика бот-сетей, базирующееся на Honeynet. На основе исследований было сделано предположение, что большинство бот-сетей действительно используют случайные стратегии сканирования, но такая стратегия является неэффективной для выявления P2P бот-сетей; предложено использовать приманки и P2P бот-сети, при этом приманки используются в синхронизации с P2P бот-сетью, но есть ограничение на количество таких приманок (в среднем от 50 до 100).

Благодаря удаленной аутентификации кода, осуществляется помощь в сборе информации о различных бот-сетях с приманкой бота. Исследования показывают, что worm-вирусы и бот-сети можно контролировать, но остановить атаки, даже после обнаружения угрозы, сложно.

Метод пассивного мониторинга трафика предусматривает осуществление выявления бот-сетей на базе из семи основных компонентов: фильтрации, применения классификатора, мониторинга трафика, детектора злонамеренной активности, анализатора, мониторинга и кластеризации, и текущего анализатора обнаружения бот-сетей. От других методов он отличается тем, что при егго использовании отсутствует необходимость иметь предварительные знания о бот-сети, но для бот-сетей, которые используют компоненты полиморфного кода, данный метод имеет низкую достоверность результата обнаружения.

Методы на основе мониторинга группы проявлений в DNS трафике: предложено обнаружение бот-сетей с помощью различных функций бот-сетей и легитимных DNS, где требуется исследование трех составляющих частей, в частности, вставка-DNS-запрос, удаление-DNS-запрос, выявление-BotDNS-запрос, при этом используется база данных для хранения данных DNS-запросов, которые включают IP-адрес источника в запросе, доменное имя запроса и метку времени полученного запроса, все сгруппированные данные DNS-запросов с именем домена и временной меткой. Проблемой является время обработки данных.

Известные методы обнаружения для идентификации различного вида MSA-RT используют наборы структурных особенностей исполняемых файлов, сетевого трафика, содержимое оперативной памяти, значение ключей системного реестра и тому подобное (Rehman Javed, 2020). Фактически данный критерий определяет особенности (или признаки), которые выбираются для формирования сигнатуры или эвристических правил (Abdelraoof, 2020).

Важной характеристикой методов обнаружения MSA-RT является способ получения данных об исследуемом объекте (Ngo, 2020). Все методы анализа можно разделить на две группы: статические и динамические (Bhardwaj, 2020). Статические методы анализа исследуют исходный вредоносный код без его выполнения в реальной или виртуальной системе и включают в себя поиск и выделение поведенческих свойств исполняемого файла (Планирование, 2017). Статический анализ предполагает использование следующих подходов: исследование структуры исполняемых файлов (например информация о компиляции исполняемого файла, экспортированные и импортированные библиотеки), удаление строк и сообщений, которые могут идентифицировать MSA-RT, исследования отпечатка MSA-RT (fingerprinting), что включает формирование хеша, определение признаков окружающего исполнения, строк реестра, и тому подобное (van Oorschot, 2020). Также одним из самых распространенных подходов

статического анализа является дизассемблирование исполняемого файла (Basu, 2020). Процесс дизассемблирования предусматривает выполнение процедуры реверс-инжиниринга и преобразование исполняемого файла в низкоуровневый набор инструкций для поиска закономерностей и связей (Subairu, 2020). Преимуществами использования данного подхода является высокая скорость и низкое ресурсопотребление (по сравнению с динамическими методами анализа) (Jurgensen, 2020). Однако использование статического анализа не позволяет в полной мере осуществлять выявление MSA-RT, что меняет свою структуру в процессе выполнения (Hacc, 2015).

В противовес статическим динамические методы анализа предусматривают выполнение изучаемого исполняемого файла с целью получения знаний о поведении MSA-RT (Harini, 2021). Преимущественно, при использовании динамических методов анализа, привлекают виртуальные машины или ресурсы, представляющие собой приманку для MSA-RT (honeypot) (Deyannis, 2020). В процессе исследования исполняемого файла могут быть обнаружены атрибуты, проявляющиеся при выполнении, в частности, создания файлов, ключей реестра, открытия/закрытия системных портов, создания мостов (Zhang, 2021).

После получения данных или признаков о MSA-RT следующим этапом является обработка этих данных с использованием алгоритмов принятия решений (Dong, 2020). Вообще алгоритмы принятия решений можно разделить на две категории: методы, основанные на экспертных знаниях и методы машинного обучения (Kuzminykh, 2019).

Методы, основанные на экспертных знаниях, используются для формализации знаний об MSA-RT и знаний специалистов в области кибербезопасности (Khan, 2020). Система, построенная на основе методов экспертной оценки, помимо выполнения вычислительных операций, формирует выводы, основанные на базе знаний и продукционных правилах (Sudhakar, 2020). Знания могут быть связаны, например, с тем, какие действия являются преступными (поведенческий анализ) или какие особенности структуры свидетельствуют о злоумышлении исполняемого файла (структурный анализ) (Li, 2020). Примерами методов на основе экспертной оценки являются: метод продукционных правил, в основе которого заложены причинно-следственные связи в виде конструкций «если-то»; нейросетевые методы, то есть методы, в основу которых заложены продукционные правила и нечеткий логический вывод, позволяющий определять комплексные признаки, в то время как элементы искусственных нейронных сетей позволяют адаптировать правила под известное MSA-RT (Zhan, 2019). Методы на основе экспертной оценки характеризуются высокой достоверностью обнаружения известных образцов MSA-RT, однако они являются недостаточно эффективными для MSA-RT, применяют новые техники противодействия антивирусному программному обеспечению (Singh, 2020).

## Материалы и методы исследования

Информационное пространство содержит следующие основные компоненты: информационные ресурсы, средства информационного взаимодействия, информационную инфраструктуру (Nass, 2017). Другими словами, необходимо формализовать компоненты информационного пространства с целью представления вредоносного программного обеспечения и среды их функционирования для дальнейшего построения моделей вредоносного программного обеспечения и разработки методов и средств их обнаружения (Li, 2019).

Информационная инфраструктура включает программно-технические средства компьютерных сетей, которые обеспечивают организацию взаимодействия информационных потоков, функционирование и развитие средств информационного взаимодействия и информационного пространства организации (Yang, 2021). В качестве среды функционирования вредоносного программного обеспечения и его выявления будем рассматривать корпоративную сеть, которая состоит из группы локальных сетей организации. Средой обитания вредоносного программного обеспечения в корпоративной сети будут средства памяти и процессоры. Поэтому, необходимо при формализации информационного пространства учитывать эти аппаратные средства.

Корпоративная сеть является частью открытого информационного пространства и, как правило, является частью Internet пространства. Вредоносное программное обеспечение может функционировать

и распространяться в одной компьютерной системе, при расширении вычислительных ресурсов к локальной компьютерной сети – в компьютерных системах локальной сети, и при расширении вычислительных ресурсов к глобальным компьютерным сетям – в компьютерных системах глобальных сетей. Таким образом, информационныое пространство, в котором может функционировать сеть, имеет подпространства на уровне корпоративных сетей и на уровне отдельных компьютерных систем, где хранятся те же возможности для развития и распространения сетей, которые есть на уровне глобальной сети. Учитывая возможность сравнения результатов функционирования на распространение руткитов в разных компьютерных системах в локальной сети в отличие от рассмотрения только одной компьютерной системы, наименьшим подпространством будем считать корпоративную сеть. Результаты, полученные в ней, можно распространить на фрагменты всего пространства, то есть глобальной сети, рассматривая ее как составленную из локальных сетей.

**Результаты и обсуждение**

Применение цифровых технологий в аграрном секторе повышает эффективность производства и упрощает управление, однако также увеличивает уязвимость системы к кибератакам и вредоносному программному обеспечению.

Локальные сети, используемые в аграрном секторе, могут быть подвержены атакам хакеров, что может привести к утечке конфиденциальной информации или нарушению работы системы. Кроме того, некоторые типы аграрного оборудования, такие как дроны и автономные тракторы, также могут быть подвержены атакам злоумышленников, что может привести к краже данных или даже управлению устройством со стороны злоумышленника.

Вредоносное программное обеспечение также может проникнуть в систему и вызвать серьезные проблемы. Например, вредоносное ПО может зашифровать данные и потребовать выкуп для их расшифровки, что может привести к серьезным финансовым потерям для предприятия. Также вредоносное ПО может уничтожить или повредить данные, что также приведет к значительным проблемам для предприятия.

Для предотвращения подобных угроз в агробизнесе необходимо принимать меры по защите системы. Например, важно использовать современные программы и аппаратные средства для защиты локальных сетей и оборудования. Также необходимо проводить регулярную обновление системы и программного обеспечения и обучать персонал правилам безопасности в сети.

Обозначим множество всего вредоносного программного обеспечения $V$, которое находится в компьютерных системах локальных сетей. То есть будем рассматривать то MSA-RT (Malicious software – Rootkit), которое при определенных обстоятельствах и в течение определенного времени эксплуатации локальных компьютерных сетей проникло в компьютерные системы, смогло пройти определенные системы защиты и функционирует там. Представим MSA-RT в локальных компьютерных сетях, особенностью которого является задача имплементации в исполняемые файлы, загрузочный сектор жесткого диска, оперативное запоминающее устройство и сеть и распространение своих копий, алгебраической системой типа $\tau = (\alpha, \beta)$ по формуле (1):

$$\mathcal{U}_V = V, \Omega_F, \Omega_P \quad (1)$$

где $\Omega_F = \{F_0, F_1, F_2, ..., F_{\alpha_1}, ...\}$ – множество операций, заданных на множестве $V$ для каждого $\alpha_1 = 0,1,2,\ldots; \Omega_P = \{P_0, P_1, P_2, ...P_{\beta_1}, ...\}$ – множество предикатов заданных на множестве $V$ для каждого $\beta_1 = 0,1,2,...; \alpha = 1, \beta = 1$ – частности операций, поэтому тип системы $\tau = (1,1)$.

Элементами множества $v_j \in V\,(j = 1,2,...)$ будем считать все объекты файловой системы, загрузочного сектора диска, оперативной памяти, сетевых пакетов, которые относятся к

рассматриваемому MSA-RT. Элементы $v_0 \in V_0 \subseteq V$ являются единичными элементами, то есть такими, которые содержат единственный функционал, содержание которого заключается в необходимости осуществления самокопирования с целью распространения, но без конкретного функционального наполнения для выполнения технически этих действий. Остальные операции представлены другими функциями. Эти элементы, формирующие множественное число $V_0$ являются порождающими для остальных различных элементов множества $V$. Функции из множества $\Omega_F$ выполняются на элементах $v_0$, что формирует другие объекты, которые будут принадлежать множеству $V$, а также могут выполняться на других элементах множества $V$, которые не принадлежат множеству $V_0$. Функции из множества $\Omega_F$ не всегда успешно будут выполняться по отношению к элементам из множества $V$, поэтому для представления MSA-RT в локальных сетях выбрано также множество предикатов, отражающих результат успешного/ неуспешного выполнения функций.

Функции $F_{\alpha_1}\left(\alpha_1 = 0,1,2,...\right)$ из множества $\Omega_F$ определим, как будут осуществлять отображение элементов из множества $V$ на нее. Их конкретное определение будет зависеть от деления множества $\Omega_F$ на подмножества по различным характеристическим свойствам. Предикаты $P_{\beta_1}\left(\beta_1 = 0,1,2,...\right)$ из множества $\Omega_P$ определим, как они будут истинными при успешном выполнении операций и ложными – в противном случае.

Множество $\Omega_F$ представим его подмножествами $\Omega_{F_{s,t}}$, которые будут отражать такие характерные для MSA-RT свойства и заложенные в его функционал особенности:

1) хранение знаний о механизме месторасположения своих следующих копий;
2) поиск места в памяти для размещения своей копии;
3) знание о механизмах воплощения в исполняемые программы;
4) механизмы записи в оперативную память;
5) сокрытие своего пребывания в компьютерных системах;
6) поиск других узлов сети для своего распространения;
7) механизмы для формирования и отправки сетевых пакетов;
8) преодоление механизмов защиты;
9) техники записи своих копий в главный загрузочный сектор;
10) выполнение деструктивных действий.

Эти характеристические свойства MSA-RT связаны с системными вызовами, которые относятся к работе с файлами, оперативной памятью и командами работы в сети: создание, открытие, закрытие, удаление, чтение, запись, добавление, нахождение, получения атрибутов и установление атрибутов, команды доступа к оперативной памяти, команды для работы в сети. Соответствие характеристических признаков MSA-RT системным вызовам представлено в табл. 1.

Таблица 1. Характеристические признаки MSA-RT

| Типичные характеристики | Создание | Открытие | Закрытие | Удаление | Чтение | Запись | Приложения | Геолокация |
|---|---|---|---|---|---|---|---|---|
| Хранение знаний о механизме месторасположения своих следующих копий | | | | | | | | + |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Поиск места в памяти для размещения своей копии | | | | | + | | | |
| Знание о механизмах воплощения в исполняемые программы | | + | + | + | + | + | | |
| Механизмы записи в оперативную память | | | | | | | | |
| Сокрытие своего пребывания в компьютерных системах | | | | | | | | |
| Поиск других узлов сети для своего распространения | | | | | | | | |
| Механизмы для формирования и отправки сетевых пакетов | | | | | | | | |
| Преодоление механизмов защиты | + | + | + | + | + | + | + | + |
| Техники записи своих копий в главный загрузочный сектор | + | + | + | + | + | + | | |
| Выполнение деструктивных действий | + | + | + | + | + | + | + | + |

Реализация характеристических свойств MSA-RT связана с системными вызовами, и команды для работы в сети будут определять наполнение функций из множества $\Omega_F$ и зависеть от них, что позволит идентифицировать такие действия.

Под новыми копиями MSA-RT будем считать совпадение MSA-RT по семантике, а не только по синтаксису. Поэтому, при распространении MSA-RT в случае изменения синтаксиса, важными являются особенности функционала независимо от синтаксиса кода.

Местом возможного размещения MSA-RT в компьютерных системах локальных компьютерных сетей может быть оперативная память, внешняя память и сетевые пакеты. Такое выделение трех основных составляющих необходимо для построения моделей MSA-RT, которые бы стали основой для разработки новых методов их обнаружения. Наличие MSA-RT во внешней памяти является характерным для всех его разновидностей. Нахождения MSA-RT в сетевых пакетах для части MSA-RT является обязательным, поскольку характеризует механизмы его распространения. Для остальной части MSA-RT

пребывание в сетевых пакетах не является обязательным, то есть их распространение может происходить другими путями, в том числе и через носители внешней памяти. Использование MSA-RT для своего пребывания возможно только при включенной работающей корпоративной сети. Для определенной части MSA-RT использование оперативной памяти является обязательным местом размещения и функционирования, а для другой – только местом на время выполнения. Учет в моделях MSA-RT места их нахождения является важным элементом, который может быть использован при их выявлении, поскольку разработчики MSA-RT закладывают в него знание об их местонахождении в корпоративной сети. То есть MSA-RT владеет техниками проверки своего местонахождения, что является важным при разработке его моделей. Элементы мест размещения MSA-RT требуют детализации в зависимости от их функционального назначения и технических характеристик, что будет влиять на модели MSA-RT и потребует их детализации и уточнения.

Рассматривая MSA-RT с точки зрения его места размещения и поиска им его для хранения себя и своих копий при распространении, представим модель MSA-RT по этому характерному свойству. Выделим в множестве всех программ набор вредоносного программного обеспечения, для которого характерное свойство заключается в сохранении заложенной в MSA-RT информации о механизмах распространения в части пребывания их во внешней памяти, оперативной памяти и сетевых пакетах. Осуществим ее формализованное представление для использования в процессе поиска MSA-RT.

Выделим в множестве $\Omega_F$ подмножества $\Omega_{F_\rho}$ таким образом, чтобы $\Omega_F = \cup_{p=1}^{k} \Omega_{F_p}$, где k – количество характеристических свойств MSA-RT. Тогда алгебраическую систему для первого свойства, характеризующую знание о местонахождении MSA-RT при $p = 1$ для всей локальной сети, зададим по формуле (2):

$$\mathcal{U}_{V,1} = V, \Omega_{F_1}, \Omega_{P_1} \tag{2}$$

где $\Omega_{F_1}$ – множество операций, заданных на множестве $V$; $\Omega_{P_1}$ – множество предикатов, заданных на множестве $V$.

Пусть $V = \cup_{s=1}^{n} V_s$, то есть выделим в каждой локальной сети подмножество MSA-RT $V_s$, где $s = 1, 2, \ldots n$. Для выделенных подмножеств из множества $V$ в момент времени $t = 0$ будет справедливо утверждение $\cap_{s=1}^{n} V_s = \varnothing$. Действительно, с самого начала первоначальной установки программного обеспечения во все сети в них нет MSA-RT. В процессе увеличения времени их работы, то есть при $t > 0$, вероятность появления MSA-RT определенного на различных участках сети растет, поэтому справедливым может быть утверждение $\cap_{s=1}^{n} V_s \neq \varnothing$. Зададим алгебру для первого свойства, характеризующего знание о местонахождении MSA-RT при $p = 1$ для одной из форм сети по формуле (3):

$$\mathrm{B}_{V_{s,1}} = V_s, \Omega_{F_1} \tag{3}$$

где $s$ – количество узлов; $\Omega_{F_1}$ – множество функций, заданных на множестве $V$, которое влияет на место размещения следующих копий MSA-RT.

Это множество функций осуществляет отображение копии MSA-RT в определенный объект внешней памяти, оперативного запоминающего устройства и сетевого пакета. Если $\nu_{s,j,l} \in V_s$, где $j$ – это номер элемента MSA-RT, $l$ – номер версии $j$ элемента, тогда $F_{1,k} \in \Omega_{F_1}$, где k – количество функций в множестве $\Omega_{F_1}, k \in N$. Эти функции $F_{1,k}$ осуществляют отображение элемента $\nu_{s,j,l}$ в множество $V_s$, то есть $F_{1,k}\left(\nu_{s,j,l}\right) = \nu_{s,j,l+1}$, где $\nu_{s,j,l+1} \in V_s$, если следующая копия MSA-RT будет

создаваться в той же сети. Но следующая копия может создаваться и в другой сети, поэтому функцию $F_{1,k}$ можно задать по формуле (4):

$$F_{1,k}\left(v_{s,j,l}\right)=\begin{cases} v_{s,j,l+1}, v_{s,j,l+1} \in V_s \\ v_{s',j,l'}, v_{s',j,l'} \notin V_s \end{cases} \quad (4)$$

где $s'$ – номер в сети отличный от $s$; $l'$ – номер копии MSA-RT отличный от $l'$.

Для функции $F_{1,k}$ существует также обратная функция $F_{1,k}^{-1}$, которая устанавливает соответствие между элементами множества $V$ по формуле (5):

$$F_{1,k}^{-1}\left(v_{s,j,l+1}\right)=v_{s,j,l}$$
$$F_{1,k}^{-1}\left(v_{s',j,l'}\right)=v_{s,j,l} \quad (5)$$

Для сетей с номером $s$ в случае появления $j$-го элемента множества $V$ через время $t$ от начала ее функционирования в сети могут быть созданы копии непосредственно в этой же $s$-й или других сегментах локальной сети. Поэтому, результат распространения одного элемента MSA-RT можно отобразить последовательностями в табл. 2. Данное представление зависит от времени и узла, поэтому его можно интерпретировать как временную модель распространения MSA-RT.

Таблица 2. Элементы MSA-RT во временном представлении

| Номер сети | Время, t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | … | $t_1$ | … | $t_z$ | … |
| 1 | | | | $v_{1,j,0}$ $\left(v_{s',j,0'}\right)$ | | … | | | $v_{1,j,1}$ | … |
| … | … | … | … | … | … | … | … | … | … | … |
| s-2 | | | | | $v_{s-2,j,0}$ $\left(v_{s',j,0'}\right)$ | … | $v_{s-2,j,1}$ | | | … |
| s-1 | | | | $v_{s-1,j,0}$ $\left(v_{s',j,0'}\right)$ | | … | $v_{s-1,j,1}$ $\left(v_{s+1',j,0'}\right)$ | | | … |
| s | | $v_{s,j,0}$ | | | $v_{s,j,1}$ | … | $v_{s,j,l}$ | … | $v_{s,j,l+1}$ | … |
| s+1 | | | $v_{s+1,j,0}$ $\left(v_{s',j,0'}\right)$ | | | … | | | | … |
| s+2 | | | $v_{s+2,j,0}$ $\left(v_{s',j,0'}\right)$ | | | … | | | $v_{s+2,j,1}$ | … |
| … | … | … | … | … | … | … | … | … | … | … |
| n | | | | | | … | $v_{n,j,0}$ $\left(v_{s',j,1'}\right)$ | | | … |

Переходы, которые будут осуществляться между элементами, между различными узлами будут содержать промежуточные уровни, отвечающие за формирование, пересылку и обработку сетевых пакетов. Также в дугах есть еще один уровень переходов, который отображает выполнение операции

распространения копий с использованием оперативного запоминающего устройства. Временную диаграмму для графа изобразим на рис. 1.

| | 0 | 1 | 2 | 3 | 4 | … | $t_1$ | … | $t_z$ | … |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | + | … | | … | | … |
| … | … | … | … | … | … | … | … | … | … | … |
| s-2 | | | | | + | … | + | … | | … |
| s-1 | | | | + | | … | + | … | | … |
| s | | + | | | + | … | + | … | + | … |
| s+1 | | | + | | | … | | … | | … |
| s+2 | | | + | | | … | | … | + | … |
| … | … | … | … | … | … | … | … | … | … | … |
| n | | | | | | … | + | … | | … |

Рисунок 1. Временная интерпретация распространение элемента $vs,j,l$ из множества $V$

Сводка данных о распространении копий со всех сегментов локальной сети, например, изображена на рис. 2.

| Время, t | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ | $t_9$ | $t_{10}$ | $t_{11}$ | $t_{12}$ | $t_{13}$ | $t_{14}$ | $t_{15}$ | $t_{16}$ | $t_{17}$ | $t_{18}$ | $t_{19}$ | $t_{20}$ | $t_{21}$ | $t_{22}$ | … |
| | | + | | + | | | | + | | + | | | | | + | | | | | | + | … |

Рисунок 2. Сводная временная интерпретация распространение элемента $v_{s,j,l}$ из множества $V$

Следствием из такой интерпретации распространения элемента $v_{s,j,l}$ из множества $V$ является возможность вычисления среднего времени распространения для одного элемента и многих по формуле (6):

$$\Delta t = \frac{t_z - t_0}{\sum_{i=1}^{r} l_i} \tag{6}$$

где $l_i$ – количество копий элемента $v_{s,j,l}$ из множества $V$ в каждом узле; $i$ – количество сегментов в сети; $t_0$ – начальное время; $t_z$ – текущее время работы.

Также эту формулу можно обобщить для всех элементов из множества $V$, которые могут распространяться, тогда среднее время распространения для всех элементов определяется по формуле (7):

$$\Delta t = \frac{t_z - t_0}{\sum_{j=1}^{q}\sum_{i=1}^{r} l_{i,j}} \tag{7}$$

где $l_{i,j}$ – количество копий $j$ -того элемента $v_{i,j,l}$ из множества $V$ в каждом узле; $i$ – количество сегментов в сети; $j$ -тый элемент из множества $V$; $t_0$ – начальное время; $t_z$ – текущее время работы.

Кроме того, из формулы (6) можно выразить скорость распространения MSA-RT за определенное время по формуле (7):

$$w = \frac{\sum_{j=1}^{q}\sum_{i=1}^{r} l_{i,j}}{t_z - t_0} \quad (8)$$

Эту скорость $w$ можно использовать для оценки прогноза распространения MSA-RT в течение некоторого времени.

Важным, также, является исследование количества распространения элементов из множества $V$ в конкретных узлах сети в позиционировании их от места распространения и от определенной копии одного элемента, то есть определения количества распространенных копий в узлах одного элемента $v_{i,j,l}$ из множества $V$. Матрицей смежности представим в табл. 3 зависимость порожденных копий элементов в узлах от копий элементов из различных узлов.

Матрица является несимметричной, ибо граф распространения MSA-RT является ориентированным. Очевидно, что копии элементов, которые распространены в одном сегменте могут распространить свои копии тоже на эту же сеть. Поэтому, при условии распространения MSA-RT в течение длительного времени возможно наличие всех ненулевых элементов матрицы. Такие матрицы для одного или более элементов из множества $V$, полученные многократно на протяжении определенного времени, позволяют определять уровень безопасности распределенной многоуровневой системы выявления MSA-RT.

Таблица 3. Зависимость порожденных копий элементов в узлах

| Номер | 1 | 2 | … | n | Всего |
|---|---|---|---|---|---|
| 1 | $l_{1,1}$ | $l_{1,1}$ | … | $l_{1,n}$ | |
| 2 | $l_{1,1}$ | $l_{2,2}$ | … | $l_{2,n}$ | |
| … | … | … | … | … | … |
| n | $l_{n,1}$ | $l_{n,2}$ | … | $l_{n,n}$ | |
| Всего | $\sum_{i=1}^{n} l_{i,1}$ | $\sum_{i=1}^{n} l_{i,2}$ | … | $\sum_{i=1}^{n} l_{i,n}$ | $\sum_{j=1}^{n}\sum_{i=1}^{n} l_{i,j}$ |

Рассмотрим MSA-RT относительно места размещения без учета времени, в течение которого оно распространялось и будет распространяться. Особым признаком выделим количество объектов, которые могут быть размещены в памяти каждой сети. Это количество является конечным и зависит от объема памяти, причем в разных сегментах она может быть разной. Для сети с номером $s$ в случае появления $j$-того элемента множества $V$ через время $t$ от начала ее функционирования в сети могут быть созданы копии непосредственно в этой же $s$-й или других локальных сетях. Поэтому результат распространения одного элемента MSA-RT можно отобразить последовательностями в табл. 4, которые, в отличие от временной модели, могут быть размещенными не изначально, а в определенной части памяти и распространяться в объекты как до, так и после объекта с $v_{s,j,l}$.

Зададим обобщенное представление данных из табл. 3 матрицей смежности. Для этого введем обозначение объектов памяти переменными и их представление в виде линейных многочленов с коэффициентами. Действительно, каждому объекту, сведения о котором представлены в таблицах файловых систем, можно поставить во взаимно-однозначное соответствие переменную с индексом. Поскольку количество таких объектов конечное, тогда многочлен будет иметь конечное количество слагаемых. Построим коэффициенты переменных многочлена таким образом, чтобы они содержали информацию о MSA-RT, о его происхождении и различных других атрибутах.

Таблица 4. Элементы MSA-RT в памяти

| Номер сегмента | Количество объектов в памяти | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | … | $k_m$ | … | $k_{z(s)-1}$ | $k_{z(s)}$ |
| 1 | | | | $v_{1,j,0}$ $\left(v_{s',j,0'}\right)$ | | … | | | $v_{1,j,1}$ | |
| … | … | … | … | … | … | … | … | … | … | |
| s-2 | | | | | $v_{s-2,j,0}$ $\left(v_{s',j,0'}\right)$ | … | $v_{s-2,j1}$ | | | |
| s-1 | | | | $v_{s-1,j,0}$ $\left(v_{s',j,0'}\right)$ | | … | $v_{s-1,j,1}$ $\left(v_{s+1',j,0'}\right)$ | | | |
| s | $v_{s,j,1}$ | | | $v_{s,j,0}$ | | … | $v_{s,j,l}$ | … | $v_{s,j,l+1}$ | |
| s+1 | | | $v_{s+1,j,0}$ $\left(v_{s',j,0'}\right)$ | | | … | | | | |
| s+2 | | $v_{s+2,j,1}$ | | | | … | $v_{s+2,j,0}$ $\left(v_{s',j,0'}\right)$ | | | |
| … | … | … | … | … | … | … | … | … | … | |
| n | | $v_{n,j,0}$ $\left(v_{s',j,1'}\right)$ | | | | … | | | | |

Выделим из них такие и введем их обозначения: (1) номер объекта из этого же места пребывания; (2) номер объекта этой же сети но другого места пребывания; (3) номер объекта с другой сети, с которого поступило MSA-RT; (4) номер сегмента, с которого поступил пакет с MSA-RT. Для их отражения коэффициентами используем базис $i, j, k$ следующим образом: $\alpha_{1,s,p}+\alpha_{2,s,p}i+\alpha_{3,s,p}j+\alpha_{4,s,p}k$, где $\alpha_{x,s,p}$ – $x$ -й коэффициент, который выбирается из множества $\{1;2;3;4\}$, $s$ – номер сегмента в сети, $p$ – номер объекта, для которого задан коэффициент. Кроме того, для упрощения представления таких коэффициентов, в которых хранятся сведения об MSA-RT, осуществим их представление в матричном виде по формуле (9):

$$\begin{pmatrix} \alpha_{1,s,p} & \alpha_{2,s,p}i \\ \alpha_{3,s,p}j & \alpha_{4,s,p}k \end{pmatrix}, \begin{pmatrix} \alpha_{1,s,p} & \alpha_{2,s,p} \\ \alpha_{3,s,p} & \alpha_{4,s,p} \end{pmatrix} \quad (9)$$

Обозначим $A_{s,p}=\alpha_{1,s,p}+\alpha_{2,s,p}i+\alpha_{3,s,p}j+\alpha_{4,s,p}k$ и, тогда, в результате, получаем такое представление объектов в сети по формуле (10):

$$\begin{matrix} A_{1,1}x_{1,1} & +A_{1,2}x_{1,2} & +A_{1,3}x_{1,3} & \dots & +A_{1,p_1}x_{1,p_2} \\ A_{2,1}x_{2,1} & +A_{2,2}x_{2,2} & +A_{2,3}x_{2,3} & \dots & +A_{2,p_2}x_{2,p_2} \\ \dots & \dots & \dots & \dots & \dots \\ A_{n,1}x_{n,1} & +A_{n,2}x_{n,2} & +A_{n,3}x_{n,3} & \dots & +A_{n,p_n}x_{n,p_n} \end{matrix} \quad (10)$$

где каждая строка – выражение, отражающее состояние объектов в сегменте, а все строки – состояние объектов в локальной сети, причем $p_s$ – количество объектов в сегменте, $s$ – количество сегментов в сети. Для представления матрицей сведений об объектах необходимо дополнить все

последовательности выражений к слагаемым, в которых переменные $x_{s,p_{s+j}}=0$, причем $p_s+j\le max(p_s)$ для всех $j=0,1,\ldots,max(p_s)-p_s$. Таким образом, получим такую матрицу сведений об объектах в сети:

$$\begin{pmatrix} \begin{pmatrix}\alpha_{1,1,1} & \alpha_{2,1,1}\\ \alpha_{3,1,1} & \alpha_{4,1,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,1,1} & \alpha_{2,1,1}\\ \alpha_{3,1,1} & \alpha_{4,1,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,1,1} & \alpha_{2,1,1}\\ \alpha_{3,1,1} & \alpha_{4,1,1}\end{pmatrix} & \cdots & \begin{pmatrix}\alpha_{1,1,1} & \alpha_{2,1,1}\\ \alpha_{3,1,1} & \alpha_{4,1,1}\end{pmatrix} \\ \begin{pmatrix}\alpha_{1,2,1} & \alpha_{2,2,1}\\ \alpha_{3,2,1} & \alpha_{4,2,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,2,1} & \alpha_{2,2,1}\\ \alpha_{3,2,1} & \alpha_{4,2,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,2,1} & \alpha_{2,2,1}\\ \alpha_{3,2,1} & \alpha_{4,2,1}\end{pmatrix} & \cdots & \begin{pmatrix}\alpha_{1,2,1} & \alpha_{2,2,1}\\ \alpha_{3,2,1} & \alpha_{4,2,1}\end{pmatrix} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \begin{pmatrix}\alpha_{1,n,1} & \alpha_{2,n,1}\\ \alpha_{3,n,1} & \alpha_{4,n,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,n,1} & \alpha_{2,n,1}\\ \alpha_{3,n,1} & \alpha_{4,n,1}\end{pmatrix} & \begin{pmatrix}\alpha_{1,n,1} & \alpha_{2,n,1}\\ \alpha_{3,n,1} & \alpha_{4,n,1}\end{pmatrix} & \cdots & \begin{pmatrix}\alpha_{1,n,1} & \alpha_{2,n,1}\\ \alpha_{3,n,1} & \alpha_{4,n,1}\end{pmatrix} \end{pmatrix} \quad (11)$$

Учитывая, что возможными местами нахождения MSA-RT могут быть основная и вторичная память и сетевые пакеты в каждой сети, то зададим их множеством $Q=\{0;1;2\}$, где элемент «0» будет отвечать за нахождение в основной памяти, элемент «1» – во вторичной памяти, элемент «2» – в сетевом пакете. Нахождение копии элемента множественного числа $V$ в сетевом пакете может быть в момент времени, когда уже осуществлено формирование пакета и происходит его пересылка и получение, поэтому именно в этот момент времени копия не будет находиться в основной или вторичной памяти. Обозначив место пребывания $M_q$, где $q\in Q$, введем соответствующую функцию по формуле (12):

$$R_{1,k}\left(\nu_{s,j,l}\right)=\begin{cases}0,\nu_{s,j,l}\in M_0\\ 1,\nu_{s,j,l}\in M_1\\ 2,\nu_{s,j,l}\in M_2\end{cases} \quad (12)$$

где $\nu_{s,j,l}$ – элемент из множества $V$ в каждом узле.

Далее используем значение этой функции $R_{1,k}\left(\nu_{s,j,l}\right)$ в матрице соответствия каждой сети места пребывания элемента множества $V$ и такие матрицы строим для каждого $j$-го элемента.

Зададим алгебру для свойства, которое характеризует поиск MSA-RT места в памяти для размещения своей копии в компьютерных системах. При этом установим $P=2$ для одного из сегментов сети по формуле (13):

$$\mathcal{B}_{V_{s,2}}=V_s,\Omega_{F_2} \quad (13)$$

где $s$ – количество узлов; $\Omega_{F_2}$ – множество функций, заданных на множестве $V$, которая осуществляет поиск MSA-RT места в памяти для размещения своей копии в компьютерных системах.

Введем предикаты на множестве $V$ таким образом, что они будут отображать результат выполнения соответствующих функций во множественном числе $\{0;1\}$, то есть наличие связи между элементами $\nu_{s,j,l}$ и $\nu_{s,j,l=1}$, по формуле (14):

$$R_{1,k}\left(\nu_{s,j,l},\nu_{s,j,l+1}\right)=\begin{cases}1,F_{1,k}\left(\nu_{s,j,l}\right)=\nu_{s,j,l+1}\\ 0,F_{1,k}\left(\nu_{s,j,l}\right)\ne\nu_{s,j,l+1}\end{cases} \quad (14)$$

где $s$ – номер сегмента в сети; $l$ – номер копии MSA-RT; $\nu_{s,j,l}\in V_s$.

Тогда зададим модель по формуле (15):

$$\mathcal{M}_{V,1}=V;\Omega_{P_{1,k}} \quad (15)$$

где $\Omega_{P_{1,k}}$ – множество предикатов, заданных на множестве $V$.

Разделим все функции, заданные на множестве $V$ и выполняющие действия по воплощению MSA-RT в выполняемые программы, на подмножества так: функции записи в начало исполняемой программы с сохранением ее функционала, функции записи в середину выполняемой программы с сохранением ее функционала, функции записи в конец исполняемой программы с сохранением ее функционала, функции записи в разные части исполняемой программы с сохранением ее функционала, функции записи в начало исполняемой программы без сохранением ее функционала, функции записи в середину выполняемой программы без сохранением ее функционала, функции записи в конец исполняемой программы без сохранением ее функционала, функции записи в разные части выполняемой программы без сохранением ее функционала. Зададим алгебру для этого свойства, что характеризует механизм воплощения MSA-RT в выполняемую программу при $p=3$ для одного из сегментов сети по формуле (16):

$$\mathcal{B}_{V_{s,3}} = V_s, \Omega_{F_3} \tag{16}$$

где $s$ – количество узлов; $\Omega_{F_3}$ – множество функций, заданных на множестве $V$, которое осуществляет воплощение своих копий MSA-RT в исполняемые программы.

Это множество функций $\Omega_{F_2}$ разделим на восемь подгрупп по способу воплощения в полезную программу $\Omega_{F_3} = \sum_{r=1}^{8} \Omega_{F_{3,r}}$. Процесс воплощения в избранную выполняемую программу осуществляется путем выполнения соответствующей функции $\Omega_{F_{3,r}} \left(r = 1,2,...,8\right)$ с учетом структуры выполняемой программы и типа операционной системы. Различие между полезной запущенной программой и программой, в которую воплощено MSA-RT появится в структуре и каждый раз при использовании типовой функции будет одинаковым. То есть, результатом успешного выполнения функции будет типичная последовательность действий, результат которой отобразится в результирующем коде выполняемой программы. Тип операционной системы, в которой могут активизироваться функции $\Omega_{F_{3,r}} \left(r = 1,2,...,8\right)$ тоже заложен их разработчиками и учитывается при поиске объектов для воплощения. Введем для отображения типа операционной системы множество $O = \left\{o_1, o_2 ..., o_q\right\}$, где $g$ – количество типов.

Добавим к этому множеству функций $\Omega_{F_3}$ подмножества по способу воплощения не в полезную программу, а для формирования отдельного файлового объекта. Для таких функций примем $r=9$ и, тогда, соответствующее подмножество будет $\Omega_{F_{3,9}}$.

Зададим алгебру для свойства, которое характеризует механизмы записи MSA-RT в оперативную память при $p=4$ для одного из сегментов сети по формуле (17):

$$\mathcal{B}_{V_{s,4}} = V_s, \Omega_{F_4} \tag{17}$$

где $s$ – количество узлов; $\Omega_{F_4}$ – множество функций, заданных на множестве $V$, которое осуществляет запись MSA-RT в оперативную память.

В набор функций входят те функции, которые относятся к тому MSA-RT, которое постоянно находится в оперативной памяти, и обозначим его подмножеством $\Omega_{F_{4,1}}$.

Ко второму подмножеству отнесем те функции, которые переводят объекты из вторичной памяти в оперативную память, и после этого они будут распространяться уже находясь там. Обозначим это подмножество $\Omega_{F_{4,2}}$. К третьей подгруппе $\Omega_{F_{4,3}}$ отнесем все объекты из вторичной памяти, которые при распространении будут использовать оперативную память, и эти функции будут выполнять действия по реализации механизмов переноса и выполнения в ней команд MSA-RT.

Зададим алгебру для свойства, которая характеризует механизмы сокрытия MSA-RT своего пребывания в компьютерных системах при $p=5$ для одного из сегментов сети по формуле (18):

$$\mathcal{B}_{V_{s,5}} = V_s, \Omega_{F_5} \quad (18)$$

где $s$ – количество узлов; $\Omega_{F_5}$ – множество функций, заданных на множестве $V$, которое осуществляет сокрытие MSA-RT своего пребывания в компьютерных системах.

Совокупность разработанных алгебр является основой для системного распределения информации о характерных особенностях MSA-RT в процессе своего жизненного цикла. Использование таких характеристик позволит осуществлять выявление MSA-RT путем анализа особенностей, которые будут проявляться при выполнении функций. То есть, выполнение каждой функции на множественном MSA-RT будет осуществляться типичным способом, знание о котором будет использоваться при обнаружении.

Формализованные свойства MSA-RT представлены разработанными алгебрами заданными моделями, которые позволили создать усовершенствованную модель MSA-RT в локальных сетях на основе алгебраической системы, которая в отличие от классической модели Коэна, модели Адлемана и модели Бонфанте детализирована до уровней свойств MSA-RT, позволяет представить MSA-RT через механизмы его распространения в плоской модели памяти, особенностью которой является рассмотрение параллельных сред, распространения в памяти различных сегментов в локальной сети. Это позволит формализовано представить MSA-RT в локальных компьютерных сетях с целью его идентификации согласно характеристическим свойствам.

Разработка алгебраических структур алгебры и алгебраических моделей предоставила возможность осуществить обобщение и формализацию предметной области для структурирования MSA-RT по правилам его построения на этапе создания. Эти алгебраические структуры содержат наполнение конкретными знаниями функционирования MSA-RT в процессе его жизненного цикла и будут основой для разработки методов их обнаружения, а также за счет переноса результатов, полученных в теории алгебраических структур, достичь оптимизации решений и соответственно эффективности при разработке практических методов, направленных на поиск конкретных подмножеств вредоносного программного обеспечения.

Для отражения различных особенностей MSA-RT, которые проявляются в его разных типах, представим множество MSA-RT $V;k$ подмножествами $V_l \subseteq V, l=\overline{1,w}, V=\sum_{l=1}^{w} V_l$, где $w$ – количество типичных разделений MSA-RT по определенным признакам или критериям. Все вредоносное программное обеспечение, которое может функционировать в локальных компьютерных сетях, можно реализовать с помощью базовых операторов, которые допускаются для выполнения конкретной компьютерной системой, и их множество обозначим через $P_1$, и базовые предикаты, которые соответствуют элементарным заданным отношениям между данными программы. Базовыми операторами будем считать вызовы библиотечных функций, базовый элемент программы или автономный фрагмент программы. Базовые предикаты принимают значение истинное или ложное.

Множество всех подмножеств множества $P_2$ обозначим через $\mathcal{V}(P_2)$ и тогда элементы этого множества соответствуют всевозможным значениям базовых предикатов. Важным при этом представлении программы через базовые операторы и предикаты является то, что для создания программы одни и те же базовые операторы могут использоваться многократно, а не однократно. Кроме того, к вредоносному программному обеспечению также отнесем завершенные фрагменты программ, которые предназначены для выполнения вспомогательных действий при подготовке злонамеренных действий. Базовые операторы и предикаты, которые будем рассматривать при дальнейших случаях, будем считать соотнесенными с языком программирования наиболее приближенным к аппаратуре компьютерной системы или оборудования компьютерной сети.

Элементы языка программирования обозначим через $p_j, j=\overline{1,m}, p_j \in P, P=P_1 \cup P_2$. Элементы множества MSA-RT обозначим с учетом их принадлежности конкретному подмножеству по формуле (19):

$$\begin{gathered} V_1=\left\{V_{11}, V_{12}, \ldots V_{1n_1}\right\} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ V_2=\left\{V_{21}, V_{22}, \ldots V_{2n_2}\right\} \\ V_n=\left\{V_{n1}, V_{n2}, \ldots V_{nn_k}\right\} \end{gathered} \quad (19)$$

где $n_j$ – количество элементов MSA-RT $j$ -ого класса; $j=\overline{1,n_k}$.

Для конкретной вредоносной программы $v_{i,j}$ связи с элементами множества $P$ другого MSA-RT и аналогичных элементов с $V$ количество и последовательность использованных элементов можно представить в табл. 5.

Таблица 5. Количество и последовательность использованных элементов

| $V_{11}$ | $\alpha_1$ | $\alpha_2$ | $\alpha_2$ | … | $\alpha_S$ |
|---|---|---|---|---|---|
| | $p_1$ | $p_2$ | $p_3$ | … | $p_S$ |

где $\alpha_k, k=\overline{1,s}$ – количество вхождений элемента $p_k$ в структуру $v_{i,j}$, если учитывать сохранение последовательности, тогда представление будет в виде строгой последовательности. Как в первом, так и во втором представлении элемента множества $V$ можно выделить уникальную последовательность, как правило представленную байтами, по которой можно идентифицировать элемент MSA-RT, фактически устанавливая соответствие шаблону, если такое MSA-RT уже известное и с него получен шаблон. Использование шаблона возможно на практике, если он является неизменным при каждом тиражировании (размножении) элемента MSA-RT. Если же при распространении MSA-RT меняет свой код, тогда использование такого шаблона неэффективно.

Количество элементов $p_i, i=\overline{1,s}$ множества $P$ конечно и учитывая особенности архитектуры современных процессоров являются не очень большим. Но через возможность многократного использования одних и тех же элементов $p_j$, то есть коэффициенты $\alpha_j, j=1,s$ могут быть большими, то представление элементов $v_{i,j} \in V$ может быть очень разнообразным и, как следствие, сводится к комбинации с повторениями: $P_{\alpha'}\left(\alpha_1, \alpha_2, \ldots \alpha_s\right)=\frac{\alpha'!}{\alpha_1!, \alpha_2!, \ldots \alpha_s!}, \alpha'=\alpha_1+\alpha_2+\ldots+\alpha_s$. На оценку возможного результата значения $P_{\alpha'}$ будут влиять величины $\alpha_j$, которые могут быть большими, но конечными и, как следствие, будет справедливым соотношение (20):

$$\left|\lim_{\substack{\alpha_j \to q \\ 1 \le j \le s}} \frac{\left(\alpha_1+\alpha_2+\ldots+\alpha_s\right)!}{\alpha_1! \times \alpha_2! \times \ldots \times \alpha_s!}\right| \le \frac{(s \times q)!}{s \times q!} \quad (20)$$

где $q=max\left\{\alpha_j\right\}, j=\overline{1,s}$.

Как правило, элементы MSA-RT стремятся создавать с использованием минимизированного количества команд. Поэтому величина $q$ считается небольшой. Но использование современных сред

программирования и разнообразие средств сокрытия вредоносного кода приводит к существенному увеличению величины $q$ и, как следствие, величины $P_L$. Это существенно влияет на увеличение элементов уникального шаблона и, соответственно, на время поиска через усложнение вычислительного процесса.

Базы сигнатур вредоносных программ (шаблонов), которые сформированы в современных антивирусных средствах пополнялись более 20 лет и содержат достаточно большие объемы данных. Создание новых противовирусных средств на основе сигнатурного метода не позволит конкурировать с теми, которые уже накопили свои базы сигнатур, а учитывая снятие ограничений на величину $q$ , это становится неперспективным. Сигнатурный метод можно отнести к точным методам, если ставить задачу нахождения полного совпадения с шаблоном.

В связи с невозможностью пополнения базы сигнатур шаблонами известных вирусов, разработка новых систем обнаружения MSA-RT на основе такого метода не перспективна, поэтому актуальным направлением исследования является разработка новых и совершенствование существующих моделей, методов и систем обнаружения MSA-RT, которые позволяют осуществлять поиск нового MSA-RT не по их сигнатурам, а использование базы сигнатур было бы вспомогательным средством для ускорения поиска известного MSA-RT.

Формализуем область множества $P$ и отношения между ее величинами с помощью алгебраических систем. Введем операцию объединения $(\cup)$ на множестве $\mathcal{V}(P)$ и будем рассматривать ее как объединение всевозможных подмножеств множества $P$ (21):

$$\mathcal{S}_P = \langle \mathcal{V}(P); \cup; \subseteq \rangle \tag{21}$$

Действительно, элементами множества $\mathcal{V}(P)$ являются всевозможные подмножества множества $P$, тогда их объединение образует множество, в котором повторяющиеся элементы заменяются один раз, и тогда образованный перечень фактически будет одним из подмножеств множества $\mathcal{V}(P)$. В качестве нулевого элемента выступает пустое множество.

Операция включения $(\subseteq)$ будет отвечать за то, есть ли среди востребованных к выполнению инструкций все элементы из $P$ или нет. Элемент MSA-RT может включать пустое множество. Может оказаться, что среди элементов множества $V$ есть такие, что в их структуре использованы элементы, которых нет в $P$.

Алгебраическая структура $\mathcal{S}_P$ имеет тип $\langle 2,2 \rangle$, ибо операции, заданные на множестве $\mathcal{V}(P)$ являются двухместными. Операции $\cup$ и $\subseteq$ являются главными. Множеством функций являются $\sum_{E_T} = \{\cup\}$, а множеством предикатов – $\Omega_{\mathcal{V}(P)_T} = \{\subseteq\}$. Исходя из алгебраической структуры $\mathcal{S}$, задана алгебра компонентов вредоносных программ $\mathcal{V}_T = \langle \mathcal{V}(P); \cup \rangle$ и модель $\mathcal{M}_T = \langle \mathcal{V}(P); \subseteq \rangle$.

Мощность множества $\mathcal{V}(P)$ обозначим $|\mathcal{V}(P)|$, и она является для алгебраической системы $\mathcal{V}_T$ ее порядком. Мощность $\mathcal{V}(P)$ является конечной, ибо множество $P$ – конечно. Это ограничение является необходимым условием для целесообразности построения моделей поиска MSA-RT по критерию эффективности. Действительно, если бы множество $P$ было бесконечным и эта его бесконечность достигалась бы за счет композиции элементов, тогда она была бы сосчитана, и, как следствие, мощность рассматриваемого множества $\mathcal{V}(P)$ , порожденного множеством $P$ имела бы мощность континуума.

Введение таких алгебраических структур позволяет формализовать пространство, в котором находятся рассматриваемые элементы из множества MSA-RT и которое является основой для исследования различных типов MSA-RT путем добавления новых операций над элементами множества $\mathcal{V}(P)$. Представим элементы $v_{i,j} \in V_i, V_i \subseteq V$, для $i=\overline{1,n}, j=\overline{1,n_l}$ через элементы множества $\mathcal{V}(P)$ следующим набором: $v_{i,j} = (\{p_1\};\{p_1,p_2\};...)$.

Множество $V$ формируется из элементов, которые обязательно имеют хотя бы один из признаков – свойств, относящихся к MSA-RT. По этим свойствам, как по отношениям, множество $V$ подразделяется на классы – подмножества $V_i, i=\overline{1,n}$. Все элементы множества $V$ входят во множество всех программ, но по выделенным признакам они формируют лишь множество $V$, которое является подмножеством множества всех программ. В состав множества $V$ по выделенным признакам могут входить и не только элементы множества MSA-RT, которые имеют согласно требованиям к своему функционалу функции, которые есть среди выделенных признаков. Если $S$ – множество признаков-свойств, тогда $\mathcal{V}(S)$ – множество подмножеств множества $S$ и $V = \cup\mathcal{V}(S)$, поэтому можно ввести алгебраическую структуру по формуле (22):

$$\mathcal{S}_V = \langle V; \cup; \subseteq \rangle \tag{22}$$

где $\cup$ – операция объединения, которая задана на множестве $V$; $\subseteq$ – операция включения, то есть, если программа имеет одиз из признаков-свойств, она означает включение во множество $V$ всех программ, в которых есть признак из множества $S$.

Для достоверной классификации программы из множества $V$ необходимы методы, которые бы учитывали эти признаки с $S$. Есть часть полезных программ (например, архиваторы), которые имеют сходные свойства и могут быть ошибочно отнесенными во множество $V$.

Рассмотрим множественное число $V_1$, в которое входят все элементы MSA-RT с признаками сочетаниями, и зададим алгебраическую структуру по формуле (23):

$$\mathcal{S}_{V_1} = \langle V_1; F_1; P_1 \rangle \tag{23}$$

Зададим функцию $F_1 : V_1 \to P$, где предикат $P$, по формуле (24):

$$F_1(v_{1,i}, p_j) = \begin{cases} 0, p_j, v_{1i} \\ 1, p_j, v_{1i} \end{cases} \tag{24}$$

Функция $F_1$ может быть задана несколькими способами:

$$\begin{array}{lll} F_{11:} & \{P_1\} & \alpha_{11} \\ & \{P_2\} & \alpha_{21} \\ & \dots & \dots \\ & \{P_n\} & \alpha_{n1} \\ & \{P_1P_2\} \to & \alpha_{121}, \alpha_{221} \\ & \{P_1P_3\} & \alpha_{122}, \alpha_{322} \\ & \dots & \dots \\ & \{P_1P_2P_3\} & \alpha_{131}, \alpha_{23}, \alpha_{33} \\ & \dots & \dots \end{array} \tag{25}$$

То есть $F_{11}(\nu_{1j}) = (\alpha_{11};\alpha_{21};...\alpha_{n1};\alpha_{121};\alpha_{221};\alpha_{122};\alpha_{131};\alpha_{23};\alpha_{33};...)$, что означает представление последовательного вхождения каждого элемента с $\mathcal{V}(P)$ в $\nu_{1j} \in V_1, i = \overline{1,n_1}$ числом. Для установления взаимно-однозначного соответствия элементы $P$ предварительно строго упорядочиваются, и тогда в дальнейшем каждый элемент с $\nu_{1j}$ сравнивается с ними. В результате за числами $\alpha_{i_1(i_2,i_3,i_4,...,i_2n_1)}$ может следовать $\nu_{1j}$. Взяв фрагмент или фрагменты такой числовой последовательности получим сигнатуру, представленную в числовом виде.

Вторая функция может не включать количество вхождений $p_i, i = \overline{1,s}$, а будет ставить в соответствие $\nu_{1j}$ соответствующие элементы $j$ соответственно: $F_1(\nu_{1i}) = (pi_1, pi_2, ...pi_t)$, где $t$ – количество элементов $V_{1i}$.

Третий вариант функции можно построить так:

$$F_{13} : \nu_{1i} \rightarrow \begin{matrix} p_1, \alpha_1 \\ p_2, \alpha_2 \\ \dots \\ p_s, \alpha_s \end{matrix}$$

тогда $F_{13}(\nu_{1i}) = ((p_1;\alpha_1),(p_2;\alpha_2),...(p_s;\alpha_s))$, то есть на основе определения количества вхождений каждого из элементов $p_j$ в $\nu_{i1} \in V$.

Функции $F_{11}, F_{12}$, однозначно определяющие элементы, из которых состоят $\nu_{1i} \in V_1$. Функция $F_{13}$ не всегда является взаимно-однозначной, особенно в случаях, когда количество элементов $p_i$ в $\nu_{1i} \in V_1$ является минимальным, тогда вхождения $p_i$ могут быть одинаковыми для разных $\nu_{1i} \in V_1$, а также и для полезных программ.

Для осуществления классификации введем отношение эквивалентности на каждом из множеств $V_i, i = 1,n$ и представим их соответствующими алгебрами по формулам (26) и (27):

$$\mathcal{V}_S = < V, \cup \langle \leftrightarrow \rangle \tag{26}$$

$$\mathcal{V}_{V_i} = \langle V_i, \cup, \leftrightarrow \rangle \tag{27}$$

где $V_i \subseteq V$ множества для всех $i = \overline{1,n}, \sum_{i=1}^{n} V_i = \varnothing$.

Разработанные алгебраические системы и алгебры с введенными операциями на множестве MSA-RT являются основой для создания поведенческих сигнатур MSA-RT с целью их формализованного представления в системах обнаружения. Особенностью разработанных алгебраических систем является структуризация MSA-RT по типам, которая позволяет осуществлять их распределение и отнесение к подмножествам на основе характеристических свойств MSA-RT для проведения идентификации и классификации.

Рассмотрим виды угроз, которые могут быть осуществлены в локальной сети. Их анализ связан с требованиями, предъявляемыми к безопасности компьютерных систем в сети: конфиденциальность, целостность, доступность и аутентичность. Для нарушения этих требований разработчики MSA-RT закладывают в него механизмы осуществления угроз в виде таких атак: прерывания, перехват, изменение, подделка. В локальных сетях осуществление таких атак или их комбинаций происходит по отношению к аппаратному обеспечению, программному обеспечению, линиям связи и данным.

Функционирование компьютерных систем в локальных сетях связано с обработкой, хранением и распространением информации. Именно при выполнении этих действий возможным является осуществление атак, обобщенные виды которых выделим следующим образом:

1) $Z_{i,1}$ – множество несанкционированных изменений;

2) $Z_{i,2}$ – множество поддельных объектов, размещенных в систему в результате атаки;

3) $Z_{i,3}$ – множество перехватов со стороны злоумышленника средствами программ или компьютеров;

4) $Z_{i,4}$ – множество прерываний, которые совершены для вывода из строя компонентов системы.

Рассмотрим подробнее возможные события при проведении атак. В частности, элемент множества $P_{i,1}$ может быть скопирован или перенесен в другое место памяти. Тогда возможны два варианта: это событие произошло, нужный результат получен; это событие не произошло (или из-за ошибок, связанных с работой операционных систем или компонентов компьютерной системы; или в результате проведенной атаки). Зададим возможные события по формуле (28):

$$p_{f,1} \underset{\Rightarrow}{\mathrm{f}} p_{f,1}$$
$$p_{f,1} \underset{\Rightarrow}{\mathrm{z}} p_{f,1,z} \quad (28)$$

Зададим матрицами категории атак. Введем следующие вершины матрицы: источник информации, получатель информации, событие прерывания, событие перехвата, событие изменения, подделка. Связи между этими вершинами изобразим направленными дугами. Матрица инцидентности, соответствующая графе, изображена табл. 6.

Таблица 6. Матрица инцидентности видов угроз

| | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ | $l_6$ | $l_7$ | $l_8$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | -1 | 0 | 0 | -1 | 0 | -1 | 0 | -1 |
| 3 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | -1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | -1 | 1 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Виды угроз зависят от особенностей компьютерных систем и их компонентов. Выделим компоненты, для которых установим их взаимосвязь с видами угроз: аппаратное обеспечение, программное обеспечение, данные, средства организации связи. Аппаратное обеспечение из-за его доступности после сбоев, вызванных вмешательством, может отказывать в обслуживании. Программное обеспечение может иметь следующие виды угроз: отказ пользователям в доступе, несанкционированное копирование, изменение функционала программы. Данные, которые хранятся, могут быть удалены через удаление файлов, может быть отказано в доступе к ним пользователям, они могут быть несанкционированно прочитаны, может быть изменено их содержимое. Средства организации связи могут иметь такие угрозы, при осуществлении которых позволят чтения сообщений, слежение за трафиком, изменение контента, изменение времени доставки, порядка доставки сообщений или их дублирование, подделка сообщений, удаление сообщений. Таким образом, осуществлено выделение типовых угроз в локальных сетях и предложено их формализованное представление, использование которого является важным при создании распределенных систем обнаружения вредоносного программного обеспечения.

Предотвращение угроз и вредоносного ПО в агробизнесе существенно влияет на качество управления предприятием и его конкурентоспособность.

Во-первых, защита от кибератак и вредоносного ПО обеспечивает сохранность и конфиденциальность данных, что является ключевым фактором для эффективного управления предприятием. Если данные о посевах, урожаях, погодных условиях и других параметрах производства попадут в руки злоумышленников, это может привести к серьезным проблемам для предприятия, включая потерю конкурентных преимуществ и снижение доходности.

Во-вторых, предотвращение угроз и вредоносного ПО также повышает надежность и эффективность производственных процессов. К примеру, кибератаки на системы автоматизации и управления оборудованием могут привести к нарушению или остановке производства, что вызовет значительные финансовые потери. Кроме того, защита от вредоносного ПО помогает сохранять целостность данных и обеспечивать надежность работающих систем, что снижает вероятность сбоев и простоев.

Защита от кибератак и вредоносного ПО является одним из ключевых элементов стратегии управления рисками в агробизнесе. Предприятие, которое не обеспечивает достаточный уровень кибербезопасности, рискует подвергнуться серьезным убыткам, включая потерю клиентов, репутации и позиций на рынке.

Таким образом, предотвращение угроз и вредоносного ПО в агробизнесе имеет прямое влияние на качество управления предприятием, его надежность, эффективность и конкурентоспособность.

## Заключение

Для того чтобы обеспечить эффективную защиту от угроз и вредоносного ПО в агробизнесе, необходимо принимать ряд мер. Прежде всего необходимо проводить регулярную оценку уязвимостей и анализировать уровень рисков в системе. Также необходимо использовать современные технологии и программное обеспечение для защиты системы, включая средства антивирусной защиты, брандмауэры и системы мониторинга.

Необходимо проводить регулярное обучение персонала правилам кибербезопасности, так как человеческий фактор является одним из наиболее уязвимых мест в системе. Персонал должен знать, как защищать пароли, как устанавливать программное обеспечение и как отличать подозрительную активность в системе; необходимо создавать регулярные резервные копии данных и разрабатывать планы действий в случае кибератаки или взлома системы. Это поможет минимизировать потери и снизить воздействие на производственные процессы.

Защита от угроз и вредоносного ПО в агробизнесе имеет критическое значение для эффективного управления предприятием и обеспечения его конкурентоспособности. Внедрение соответствующих мер по защите системы является одним из основных элементов стратегии управления рисками и успешной деятельности в аграрном секторе.

Разработанные алгебры поведений MSA-RT являются основой для системного распределения информации о характерных особенностях MSA-RT в процессе своего функционирования. Использование таких характеристик позволит осуществлять выявление MSA-RT путем анализа особенностей, которые будут проявляться при выполнении функций. То есть выполнение каждой функции на множественном MSA-RT будет осуществляться типичным способом, знание о котором будет использоваться при обнаружении.

Свойства MSA-RT представлены разработанными алгебрами и заданными моделями, которые позволили создать усовершенствованную модель MSA-RT в локальных сетях, которая в отличие от классической модели Коэна, детализирована до уровней свойств MSA-RT. Она позволяет представить MSA-RT через механизмы его распространения в плоской модели памяти, особенностью которой является рассмотрение параллельных сред распространения в памяти различных сегментов в локальной сети. Это позволит формализовано представить MSA-RT в локальных компьютерных сетях с целью его идентификации согласно характеристическим свойствам.

Разработанные алгебраические системы и алгебры с введенными операциями на множестве MSA-RT являются основой для создания поведенческих сигнатур MSA-RT с целью их формализованного представления в системах обнаружения. Особенностью разработанных алгебраических систем является структуризация MSA-RT по типам, которая позволяет осуществлять их распределение и отнесение к подмножествам на основе характеристических свойств MSA-RT для проведения идентификации и классификации.

Выделение типичных угроз MSA-RT в локальных сетях и их формализованное представление позволят создавать распределенные системы обнаружения вредоносного программного обеспечения. Усовершенствованные модели типов вредоносного программного обеспечения представлены их алгебрами поведения, и это является основой создания базиса поведенческих сигнатур, и, в отличие от известных представлений, учитывающих особенности их функционирования в локальных сетях, они позволяют осуществить классификацию по типам поведения.

**Список литературы**

1. Насс О.В., Камалова Г.А. Опыт применения элементов искусственного интеллекта при разработке информационных систем // Вестник КазНТУ, Алматы, 2015. №6(112). С. 441-445

2. Планирование задач в распределенных вычислительных системах на основе метаданных: отчет о НИР (заключительный) // ЗКАТУ им. Жангир хана: рук. Насс О.В.; исполн.: Камалова Г.А., Касымова А.Х., Муталова Ж.С., Нургалиева А.Ә., Есенғали Қ.Қ. Уральск, 2017. 53 с. № ГР 0115РК00068. Инв. № 0218РКИ0011.

3. Abdelraoof A., Azab, M., & Stoppa, I. (2020). Write-protection enforcement: Hypervisor-backed kernel hardening. In Proceedings of the ACM Symposium on Applied Computing. Pp. 1736–1744. https://doi.org/10.1145/3341105.3373919

4. Basu K., Krishnamurthy, P., Khorrami, F., & Karri, R. (2020). A Theoretical Study of Hardware Performance Counters-Based Malware Detection. IEEE Transactions on Information Forensics and Security, 15, Pp. 512–525. https://doi.org/10.1109/TIFS.2019.2924549

5. Bhardwaj A., & Goundar, S. (2020). Keyloggers: silent cyber security weapons. Network Security, 2020(2). Pp. 14–19. https://doi.org/10.1016/S1353-4858(20)30021-0

6. Deyannis D., Karnikis, D., Vasiliadis, G., & Ioannidis, S. (2020). An enclave assisted snapshot-based kernel integrity monitor. In EdgeSys 2020 - Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2020. Pp. 19–24. https://doi.org/10.1145/3378679.3394539

7. Dong S., Xiong, Y., Huang, W., & Ma, L. (2020). KIMS: Kernel Integrity Measuring System based on TrustZone. In Proceedings - 2020 6th International Conference on Big Data Computing and Communications, BigCom 2020. Pp. 103–107. https://doi.org/10.1109/BigCom51056.2020.00022

8. Harini C., & Fancy, C. (2021). A study on the prevention mechanisms for kernel attacks. Lecture Notes in Networks and Systems, 130, pp. 11–17. https://doi.org/10.1007/978-981-15-5329-5_2

9. Jurgensen G., Neises, M., & Alexander, P. (2020). An seL4-based architecture for layered attestation. In ACM International Conference Proceeding Series. Pp. 147–148. https://doi.org/10.1145/3384217.3386398

10. Khan F., Ncube, C., Ramasamy, L. K., Kadry, S., & Nam, Y. (2020). A Digital DNA Sequencing Engine for Ransomware Detection Using Machine Learning. IEEE Access, 8. Pp. 119710–119719. https://doi.org/10.1109/ACCESS.2020.3003785

11. Kuzminykh I., & Yevdokymenko, M. (2019). Analysis of Security of Rootkit Detection Methods. In 2019 IEEE International Conference on Advanced Trends in Information Theory, ATIT 2019 – Proceedings. Pp. 196–199. https://doi.org/10.1109/ATIT49449.2019.9030428

12. Li H., Huang, J., Liang, B., Shi, W., Wu, Y., & Bai, S. (2020). Identifying parasitic malware as outliers by code clustering. Journal of Computer Security, 28(2), Pp.157–189. https://doi.org/10.3233/JCS-191313

13. Li J. P., & Sun, R. (2019). Prediction of Virtual Machine State Based on BP Neural Network. In Journal of Physics: Conference Series (Vol. 1345). https://doi.org/10.1088/1742-6596/1345/4/042078

14. Nass O. V., Yessengali K. K. Robotic plant watering system based on the Arduino Microcontroller // Ғылым және білім. — 2017. — № 4 (49). — С. 155–161. (http://nauka.wkau.kz/index.php/arkhiv-proshlykh-nomerov)
15. 15. Ngo F. T., Agarwal, A., Govindu, R., & MacDonald, C. (2020). Malicious software threats. The Palgrave Handbook of International Cybercrime and Cyberdeviance. Pp. 793-813. https://doi.org/10.1007/978-3-319-78440-3_35
16. Rehman Javed A., Jalil, Z., Atif Moqurrab, S., Abbas, S., & Liu, X. (2020). Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles. Transactions on Emerging Telecommunications Technologies. https://doi.org/10.1002/ett.4088
17. Singh J., & Singh, J. (2020). A survey on machine learning-based malware detection in executable files. Journal of Systems Architecture. https://doi.org/10.1016/j.sysarc.2020.101861
18. Subairu S. O., Alhassan, J., Misra, S., Abayomi-Alli, O., Ahuja, R., Damasevicius, R., & Maskeliunas, R. (2020). An experimental approach to unravel effects of malware on system network interface. Lecture Notes in Electrical Engineering, 612, 225–235. https://doi.org/10.1007/978-981-15-0372-6_17
19. Sudhakar & Kumar, S. (2020). An emerging threat Fileless malware: a survey and research challenges. Cybersecurity, 3(1). Pp. 1-12. https://doi.org/10.1186/s42400-019-0043-x
20. van Oorschot P. C. (2020). Malicious Software. Information Security and Cryptography. Pp. 183–211. https://doi.org/10.1007/978-3-030-33649-3_7
21. Yang M., & Fu, F. (2021). Secure big data computing based on trusted computing and key management. Advances in Intelligent Systems and Computing, 1244 AISC. Pp. 1114–1118. https://doi.org/10.1007/978-3-030-53980-1_169
22. Zhan D., Ye, L., Fang, B., & Zhang, H. (2019). SAVM: A practical secure external approach for automated in-VM management. Concurrency Computation, 31(23). https://doi.org/10.1002/cpe.4482
23. Zhang Z., Cheng, Y., Gao, Y., Nepal, S., Liu, D., & Zou, Y. (2021). Detecting Hardware-Assisted Virtualization with Inconspicuous Features. IEEE Transactions on Information Forensics and Security, 16. Pp. 16–27. https://doi.org/10.1109/TIFS.2020.3004264

## Development of digital technologies in the application of methods for detecting atypical network activity in the agricultural sector of Russia

**Yuri V. Zabaykin**
Candidate of Economic Sciences, Associate Professor, Associate Professor of the Department "Business Management and Service Technologies"
Rosbiotech
Moscow, Russia
89264154444@Yandex.ru
ORCID 0000-0000-0000-0000

Received 25.03.2023
Accepted 14.04.2023
Published 15.05.2023

**Abstract**

Automation of industrial production has become an integral part of modern industry, but at the same time there is a problem of detecting atypical network activity, which may indicate a cyber attack or system malfunction. To solve this problem, methods of detecting anomalies in network activity are used. In this paper, the application of such methods in the conditions of automation of industrial production is considered. An

overview of existing methods was carried out and their applicability to this field was analyzed. Experiments were also conducted on real data of industrial networks, as a result of which anomalies in network activity were identified and conclusions were drawn about the applicability of methods in this area. The results of the study showed that the methods of detecting anomalies in network activity can be successfully applied to detect atypical network activity in the conditions of industrial automation. Automation methods are increasingly used in industrial production to achieve high efficiency and safety. However, there are a number of problems associated with ensuring security and protection against cyber attacks. In this regard, an urgent task is to detect atypical network activity in automation systems. This article proposes a method for detecting abnormal network activity based on machine learning, which allows you to identify suspicious events in real time. The method is based on the analysis of user behavior and monitoring of traffic on the network. It allows you to detect anomalies in network activity, such as unusual requests, data leaks, DDoS attacks, and many others. The proposed method is effective and allows for high protection of industrial automation systems from cyber threats. Its application can significantly increase the safety and reliability of production processes, as well as reduce the risks associated with potential cyber attacks.

**Keywords**
management, teaching, technology, protection, training.

**References**

1. Nass O.V., Kamalova G.A. Experience of using artificial intelligence elements in the development of information systems / Bulletin of KazNTU, Almaty, 2015, №6 (112). - Pp. 441-445

2. Task planning in distributed computing systems based on metadata: Research report (final) / Zhangir Khan State Technical University: ruk. Nass O. V.; performed by: Kamalova G. A., Kasymova A. H., Mutalova Zh. S., Nurgalieva A. A., Yesengali K. K. – Uralsk, 2017. – 53 p. – no. GR 0115RK00068. – Inv. no. 0218RKI0011.

3. Abdelrauf A., Azab M. and Stoppa I. (2020). Forced write protection: Enhanced kernel protection with hypervisor support. In Proceedings of the ACM Symposium on Applied Computing (pp. 1736-1744). https://doi.org/10.1145/3341105.3373919

4. Basu K., Krishnamurti P., Khorrami F. and Curry R. (2020). A theoretical study of malware detection based on hardware performance counters. IEEE Transactions on Information Forensics and Security, 15, 512-525. https://doi.org/10.1109/TIFS.2019.2924549

5. Bhardwaj A. and Gundar S. (2020). Keyloggers: the silent weapon of cybersecurity. Network Security, 2020(2), 14-19. https://doi.org/10.1016/S1353-4858 (20)30021-0

6. Deiannis D., Karnikis D., Vasiliadis G. and Ioannidis S. (2020). Snapshot-based kernel integrity monitor with enclave support. EdgeSys 2020 - Proceedings of the 3rd ACM International Seminar on Edge Systems, Analytics and Networking, which is part of EuroSys 2020 (pp. 19-24). https://doi.org/10.1145/3378679.3394539

7. Dong S., Xiong Yu., Huang W. and Ma L. (2020). KIMS: A core integrity measurement system based on TrustZone. In Proceedings of the 6th International Conference on Computing and Communications with Big Data - 2020, BigCom 2020 (pp. 103-107). https://doi.org/10.1109/BigCom51056.2020.00022

8. Harini K., & Fancy K. (2021). Investigation of mechanisms to prevent attacks on the core. Lecture notes on Networks and Systems, 130, 11-17. https://doi.org/10.1007/978-981-15-5329-5_2

9. Jurgensen G., Neizes M. and Alexander P. (2020). seL4-based architecture for multi-level certification. In the series of proceedings of the ACM International Conference (pp. 147-148). https://doi.org/10.1145/3384217.3386398

10. Khan F., Nkube S., Ramasami L. K., Kadri S. and Nam Yu. (2020). A digital DNA sequencing mechanism for detecting ransomware using machine learning. Access to IEEE, 8, 119710-119719. https://doi.org/10.1109/ACCESS.2020.3003785

11. Kuzminykh I., Evdokimenko M. (2019). Security analysis of rootkit detection methods. In 2019, the IEEE International Conference on Advanced Trends in Information Theory, ATIT 2019 - Proceedings (pp. 196-199). https://doi.org/10.1109/ATIT49449.2019.9030428

12. Li H., Huang J., Liang B., Shi W., Wu Y. and Bai S. (2020). Identification of malware parasites as outliers using code clustering. Computer Security Journal, 28(2), 157-189. https://doi.org/10.3233/JCS-191313

13. Lee J. P. and Sun R. (2019). Predicting the state of a virtual machine based on the BP neural network. In Journal of Physics: Conference Series (Volume 1345). https://doi.org/10.1088/1742-6596/1345/4/042078

14. Nass O. V., Esengali K. K. Robotic plant watering system based on Arduino microcontroller // GYlym feminine bilim. — 2017. — № 4 (49). — Pp. 155-161. (http://nauka.wkau.kz/index.php/arkhiv-proshlykh-nomerov )

15. 15. Ngo F. T., Agarwal A., Govindu R. and McDonald S. (2020). Threats of malicious software. Palgrave's Handbook on International Cybercrime and Cyber Threats. https://doi.org/10.1007/978-3-319-78440-3_35

16. Rehman Javed A., Jalil Z., Atif Mokurrab S., Abbas S. and Liu H. (2020). Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles. Transactions related to new telecommunication technologies. https://doi.org/10.1002/ett.4088

17. Singh J., & Singh J. (2020). A survey on malware detection in executable files based on machine learning. Journal of System Architecture. https://doi.org/10.1016/j.sysarc.2020.101861

18. Subairu S. O., Alhassan J., Misra S., Abayomi-Alli O., Ahuja R., Damasevicius R. and Maskelunas R. (2020). An experimental approach to detecting the influence of malware on the network interface of the system. Lecture notes on Electrical Engineering, 612, 225-235. https://doi.org/10.1007/978-981-15-0372-6_17

19. Sudhakar and Kumar, S. (2020). Emerging threat - malware without files: overview and research tasks. Cybersecurity, 3(1). Pp. 1-12. https://doi.org/10.1186/s42400-019-0043-x

20. van Orschot P. S. (2020). Malicious software. Information Security and Cryptography, 183-211. https://doi.org/10.1007/978-3-030-33649-3_7

21. Yang M., & Fu, F. (2021). Secure big data computing based on trusted computing and key management. Advances in Intelligent Systems and Computing, 1244 AISC, 1114-1118. https://doi.org/10.1007/978-3-030-53980-1_169

22. Zhang D., E. L., Fang B. and Zhang H. (2019). SAVM: A practical secure external approach for automated virtual machine management. Parallel Computing, 31(23). https://doi.org/10.1002/cpe.4482

23. Zhang Z., Cheng Yu., Gao Yu., Nepal S., Liu D. and Zou Yu. (2021). Hardware virtualization detection with inconspicuous features. IEEE Transactions on Information Forensics and Security, 16, 16-27. https://doi.org/10.1109/TIFS.2020.3004264