АНАЛИТИКА

УДК 004

Научно обоснованный методический аппарат обоснования рациональных способов мониторинга и реагирования на возможные инциденты безопасности в цифровой информационной инфраструктуре МЧС России и отчет о результатах апробации

Аннотация

Представлены научно обоснованные способы мониторинга и реагирования на возможные инциденты информационной безопасности в ведомственной цифровой информационной инфраструктуре и результаты их апробации в Санкт-Петербургском университете ГПС МЧС России на типовых фрагментах.

Материалы предназначены для руководителей и специалистов подразделений обеспечения информационной безопасности ведомственных информационных инфраструктур с целью возможной актуализации и совершенствования объектовых процедур мониторинга и реагирования на инциденты информационной безопасности.

Материалы разработаны с целью внедрения результатов научных исследований, выполненных Санкт-Петербургским университетом ГПС МЧС России в 2025 году по теме «Кибермониторинг».

Ключевые слова: информационная инфраструктура, информационная безопасность, инцидент, мониторинг, реагирование, способы, апробация.

Scientifically based methodological apparatus for justifying rational methods of monitoring and responding to possible information security incidents in the digital information infrastructure of the EMERCOM of Russia and the results of its approval

Abstract

The paper presents scientifically substantiated methods of monitoring and responding to possible information security incidents in the departmental digital information infrastructure and the results of their testing at the St. Petersburg University of the State Fire Service of the Ministry of Emergency Situations of Russia on typical fragments.

The materials are intended for managers and specialists of the information security units of departmental information infrastructures in order to possible updating and improvement of object procedures for monitoring and responding to information security incidents.

The materials were developed with the aim of implementing the results of scientific research conducted by the St. Petersburg University of the State Fire Service of the Ministry of Emergency Situations of Russia in 2025 on the topic of "Cyber Monitoring".

Keywords: information infrastructure, information security, incident, monitoring, response, methods, testing.

Введение

Научно обоснованные способы мониторинга и реагирования на инциденты информационной безопасности ведомственной информационной инфраструктуры и отчет о результатах апробации разработаны кафедрой прикладной математики и безопасности информационных технологий, а также апробированы совместно с центром информационных и коммуникационных технологий Санкт-Петербургского университета ГПС МЧС России в составе

авторского коллектива: Буйневич М.В., Грызунов В.В., Израилов К.Е., Метелькова А.Н., Матвеев А.В., Максимов А.В., Папырина Е.В., Рассказов М.С., Синещук М.Ю., Тукмачева М.А., Уткин О.В., Шакин Д.Н., Шестаков А.В.

Исходные данные для разработки научно обоснованных способов мониторинга и реагирования на инциденты информационной безопасности ведомственной цифровой инфраструктуры приняты в соответствии с аналитическим обзором, опубликованным в научном журнале «Национальная безопасность и стратегическое планирование» №1 (49), 2025.

1. Методология на базе оригинальных моделей и методов, связанных с выявлением и идентификацией источников деструктивных воздействий инфраструктурного генезиса

Рассмотрим основные положения методологии на базе оригинальных моделей и методов, связанных с выявлением и идентификацией источников деструктивных воздействий (далее – ДВ) инфраструктурного генезиса (далее – ИГ).

Понятие источника деструктивных воздействий инфраструктурного генезиса

Ответ на вопрос обоснования рациональных способов мониторинга возможных инцидентов информационной безопасности в цифровой информационной инфраструктуре, безотносительно ведомственной принадлежности предполагает достаточно однозначное определение собственно объекта реагирования.

Речь идет не только о терминологической «чистоте» и гармонизации самого термина – инцидент (информационной, компьютерной) безопасности, что является, в том числе предметом исследования, проведенного в рамках НИР «Модель» и НИР «Гармония» [1, 2].

Для научного обоснования соответствующего методического аппарата не меньший интерес представляет понимание его «природы», онтологии (сущности), интенсионала (смысла) и экстенсионала (значения), а также определение границ его применимости в контексте решаемой задачи.

Онтологически, это понятие напрямую связано с такой сущностью информационной безопасности как угроза. Угрозы информационной безопасности (здесь – угроза безопасности информации) являются неотъемлемой частью процесса информатизации любой организации, как частного, так и государственного сектора. Возникает потребность противодействия им не только после установления факта свершения, но и до – применяя превентивные меры. Для этого необходимо определение конкретного списка существующих угроз, возможностей их реализации, а также оценка результирующих последствий. Как следствие, создаются так называемые модели угроз информационной безопасности, систематизирующие все перечисленные потребности.

Ситуация существенно усложняется, если источником угроз является не какой-либо сторонний, по отношению к информационной системе (далее – ИС), источник, а собственные внутренние особенности структуры системы и параметры ее функционирования. В пункте 5 Методического документа Регулятора [3] раскрыто содержание понятия «источник угроз безопасности информации», в частности в подпункте 5.1.1 детализированы антропогенные источники угроз, а в подпункте 5.1.8 – приведена ссылка на техногенные источники, на которые требования документа не распространяются, однако приводятся сведения о двух нормативных технических документах [4, 5]. Регулятор допускает установление техногенных источников как факторов угроз безопасности информации в отдельных случаях, при предъявлении требования к устойчивости и надежности функционирования, в части целостности и доступности информационных ресурсов и компонентов систем и сетей.

В том же Методическом документе (Приложение 1. Термины и определения ...), дается определение сущности угрозы и уязвимости. А так как угроза принципиально не может быть реализована в отсутствие уязвимости(ей), эксплуатируемых источником, то последнюю также приходится признать фактором. Об этом косвенно утверждается в подпункте 5.1.8 Методического документа Регулятора в части трех перечисленных факторов возникновения угроз по отношению

с техногенными источникам (программное обеспечение и программно-аппаратные средства, обеспечивающие системы и сервисы сторонних организаций).

Очевидно, что не все уязвимости могут эксплуатироваться всеми источниками, поэтому имеют смысл только «результативные» (в смысле деструкции) комбинации. Тогда, если условиями признать одновременное наличие подобной деструктивной пары факторов «уязвимость – источник», а также возможность для эксплуатации, то сама пара факторов (согласно вышеприведенного определения «угрозы») должна быть отнесена к источникам, а не только собственно ее активный элемент: например, хакеры и инсайдеры – для антропогенных, и пожар или наводнение – для техногенных (включая природные).

Эту коллизию нетрудно разрешить, категориально разделив техногенные и сточники угроз на активные (эксплуатирующие уязвимости) и пассивные (собственно уязвимости). Согласно подпункту 5.1.8 Методического документа Регулятора их определение должно быть основано на статистических методах или экспертных методах. Однако в реальности статистика по угрозам, источниками которой являются вышеупомянутые внутренние особенности структуры системы и параметры ее функционирования, не ведется ввиду сложности их идентификации. Эксперты по данному вопросу (вместе с BestPractice) отсутствуют (или присутствуют, но в единичном количестве) по причине значительной наукоемкости на настоящий момент нерешенной задачи. Эксперта для решения подобного класса задач необходимо вооружить соответствующей методологической и методической базой и средствами, которые позволят оперативно выявлять и идентифицировать источники таких воздействий.

Проблематика задач подобного класса исследуется, по различным направлениям, например, в части телекоммуникаций – в рамках методологии обработки геопространственных данных генотипа и фенотипа телекоммуникаций и генезиса их развития, представленной с 2016 в работах Шестакова А.В.; в части информационной безопасности – в рамках научной школы «Инфраструктурный деструктивизма», ее руководителем проф. Буйневичем М.В. и его учениками (Максимовой Е.А., Израиловым К.Е., Покусовым В.А. и др.). Содержание научной школы определяется исследованием следующего феномена деструктивных воздействий инфраструктурного генезиса (тезисно):

- во-первых, ЦИИ и ее сегменты представляют собой, по сути, интегрированные системы, элементы которых обмениваются информационными объектами;
- во-вторых, как результат, обеспечивающее их совместную работу информационное взаимодействие становится (активным) источником угроз; взаимосвязанные подсистемы (модули) – функциональная организационная структура – же начинает обладать рядом уязвимостей (пассивный источник), также «проявляющихся» в результате интеграции;
- в-третьих, этот (активный) источник, «эксплуатируя» уязвимости, с неизбежностью приводят к угрозам, которые в рассматриваемом аспекте имеют последствия для нарушения не только информационной безопасности, но и общей работоспособности интегрированной системы;
- в-четвертых, для противодействия этому в системе должны применяться меры, направленные как на (активный) источник угроз, так и на «эксплуатируемые» им уязвимости.

Этими мерами, например, по отношению к уязвимостям интегрированной системы защиты информации (СОИБ) выступает структурная унификация (а именно – типизация подсистем и модулей), а по отношению к (активному) источнику угроз – процессная унификация (а именно – унификация информационных объектов обмена и стандартизация протокола информационного межмодульного взаимодействия). Ниже представлены подходы к решению подобного класса сложных задач, основанные на оригинальных (не заимствованных напрямую из существующих практик) теоретических конструкциях и алгоритмах. Она включает:

- оригинальные модели – формализованные представления процессов или явлений, разработанные для конкретной предметной области;
- оригинальные методы – авторские техники анализа, обработки данных или управления (в том числе, прогнозирования), не сводимые к стандартным процедурам.

Ключевая особенность – новизна и адаптивность: методология не просто комбинирует известные решения, а предлагает новые принципы работы с проблемой, причем с учетом специфики разных типов инфраструктур.

Антропоморфический подход к выявлению и идентификации источников деструктивных воздействий инфраструктурного генезиса

Цифровая информационная инфраструктура является сложнейшей программно-аппаратной и организационно-технической системой, но суть функционирования которой, в конченом итоге, состоит в информационном взаимодействии ее элементов. Это взаимодействие стратифицировано и происходит на различных уровнях: на самом нижнем (микро-) уровне взаимодействуют программы, на среднем (макро-) – сервисы и их серверы (как их программно-аппаратная реализация), на высших (мега- и мета-) – объекты (например, ИС) и субъекты.

Микро-уровень

Так как межобъектное взаимодействие в ЦИИ реализуется через программные системы, то ярким примером ДВ ИГ в ЦИИ являются уязвимости программного кода [6–9].

Исходя из описанных предпосылок, для частичного решения поставленной задачи – получения новых знаний об источниках ДВ ИГ микро-уровня – под объектом исследования возьмем непосредственно сами уязвимости в программе, а под предметом исследования – их сосуществование в единой программной среде (названное авторами взаимодействием) со всеми вытекающими эффектами (ИД). Методами достижения этого выбрана совокупность анализа жизненного цикла программы и применение антропоморфического подхода для более понятной интерпретации объектов и процессов IT-мира.

Несмотря на некоторую неопределенность и различия в существующей терминосистеме понятий «уязвимость», все они обладают тремя общими особенностями:

- во-первых, уязвимость так или иначе связана с кодом программы – логичнее всего считать, что она является частью этого кода;
- во-вторых, уязвимость может быть занесена в программу, а значит появляется в некоторый момент жизни последней – при разработке или функционировании программы;
- в-третьих, уязвимость является дефектом – то есть некоторым (пусть даже ничтожно малым) образом нарушает корректное функционирование программы; под последним можно понимать выполнение операций над данными с выдачей соответствующего результата.

Следовательно, «уязвимость» целесообразно рассматривать, привязывая ее к некоторой области в коде программы на некотором временном промежутке существования последней, что при этом будет являться причиной нарушения обработки информационных потоков.

Одно из исследований исполнителей настоящей НИР [10, 11] позволило определить жизненный цикл ПО (частичное подтверждение которого присутствует и в работах других авторов [12-14]), состоящий из ряда представлений (далее – Представление), между которыми программа осуществляет переходы в процессе своей разработки.

Были выделены следующие Представления программы: Идея, Концептуальная модель, Архитектура, Алгоритмы, Исходный код, Ассемблерный код, Машинный код и Файл образа. Каждое Представление задавалось частями категориальной пары – Форма VS Содержание; первая определяла условный синтаксис программы в этом Представлении, а вторая – суть программы на этом синтаксисе. Очевидно, что идеальным случаем будет тот, когда содержание программы на всем протяжении ее жизни остается неизменным (назовем его идеальным) – конечный программный продукт полностью соответствует изначально задуманному. Необходимо отметить, что возможен и обратный процесс – от Файла образа до Идеи (так называемый реверс-инжиниринг), исследованию которого также были посвящены работы различных авторов [15–25].

Поскольку человеко-ориентированные синтаксисы в преобладающем большинстве случаев имеют большее количество языковых элементов (по аналогии с множествами – мощность), чем машинно-ориентированные, то по мере преобразования Представлений их форма будет

уменьшаться. Так, Идея программы может быть описана с использованием разговорного языка (имеющего огромное количество слов), Алгоритмы представляют собой формализованные блок-схемы со вставками с описанием необходимых вычислений (то есть более формальный и короткий вид, но все еще с элементами разговорного языка), а Машинный код – полностью формализованный способ описания программы при помощи ограниченного и сравнительно небольшого набора инструкций процессора.

Следовательно, способ описания содержания программы в каждом из Представлений будет расти – при меньшем количестве элементов языка требуется большее количество конструкций, построенных из этих элементов. Таким образом, элемент содержания на начальных Представлениях будет разрастаться и расплываться на несколько элементов по мере движения к конечным Представлениям.

В процессе преобразований программа переходит от человека-ориентированной к машинно-ориентированной форме, поэтому в первых Представлениях могут более преобладать случайно вносимые уязвимости (поскольку злонамеренные будут с бо́льшей вероятности обнаружены другими разработчиками), а на последних – злонамеренно вносимые уязвимости (поскольку машинная автоматизация преобразований будет снижать риск любых случайно вносимых дефектов). Важным является то, что в любом из Представлений (кроме первого – собственно определяющего требуемый идеал программы) возможно появление своего типа уязвимости, обладающего отличительными особенностями.

Очевидно, что уязвимость может появиться в одном из Представлений в виде некоторой части программного кода, имеющего собственное (обособленное) содержание, вносящее отклонение в содержания программы от идеального – точка возникновения (рождения) уязвимости.

Также очевидно, что некоторое случайное или намеренное изменение кода программы на более поздних Представлениях может нейтрализовать уязвимость, заложенную туда на более ранних – точка исчезновения (смерти) уязвимости. Между же этими точками код в процессе разработки программы меняется, что может приводить и к некоторым изменениям в содержании уязвимости – ее мутации.

Следовательно, на протяжении жизненного цикла программы каждая из возможных уязвимостей может проживать собственный уникальный период жизни – эволюционировать.

Рассмотрим ситуацию, когда в программе присутствует несколько уязвимостей – для упрощения предположим, что их две. Поскольку каждая из уязвимостей могла появиться в своем Представлении и подвергнуться мутациям, то, следовательно, в последних Представлениях программы (код которой уже может непосредственно выполняться на процессорах) каждая из уязвимостей будет иметь совершенно различное содержание (как по размеру, так и по своей структуре).

Опишем такую ситуацию с помощью следующего примера. Предположим, что первая уязвимость внесена в Архитектуру программы и заключается в реализации стандартного интерфейса защиты с помощью устаревшего механизма – в Машинном коде уязвимость будет представлять собой использование множества библиотек и алгоритмов, слабых к современным атакам. Вторая же уязвимость внесена непосредственно в исходный код и заключается в реализации собственной, хотя и полностью защищённой, но достаточно ресурсоемкой функции шифрования в обход стандартных интерфейсов – это потенциально приведет к угрозе нарушения доступности на больших потоках данных. Реализация стандартных интерфейсов в системе, очевидно, должна быть оптимизирована (как по скорости, так и по используемой памяти). В результате Машинный код будет содержать обе уязвимости, хотя первая и будет нивелирована второй (в части функционала шифрования). В этом случае можно говорить о взаимодействии двух таких уязвимостей при частичной нейтрализации первой уязвимости за счет второй. Подобные сложно-возникающие эффекты практически не рассматривались ранее, хотя они могут

существенно влиять на уровень безопасности программы. Именно таким особенностям взаимодействия уязвимостей и посвящено настоящее авторское исследование.

Предложим следующую терминологию объектов и процессов IT-мира, исходя из интерпретации Представлений программы и ее уязвимостей.

Функция программы – обработка информационных потоков (получение данных, вычисление и выдача результата).

Форма программы – способ описания программы (синтаксис).

Содержание программы – суть программы, подходящая для описания в заданной форме.

Идеальное содержание программы – суть программы, соответствующая ее первоначальной задумке (то есть идее). В процессе разработки программы ее содержание может становиться отличным от идеального по различным причинам (случайным, злонамеренным, вынужденным). Также, содержание может приближаться или совмещаться с идеальным (как правило, вследствие действий по исправлению ошибок).

Представление программы – совокупность формы и содержания программы.

Уязвимость программы – отличие содержания программы от идеального. Любое отклонение функционирования программы от задуманного соответствует ее уязвимости.

Функция уязвимости – нарушение корректной обработки информационных потоков.

Возникновение уязвимости – момент изменения программы в некотором Представлении, когда ее содержание отклонилось от идеального. То есть появление недекларированного функционала в программе, и как следствие – увеличение ее некорректного поведения, и есть внесение уязвимости.

Исчезновение уязвимости – момент изменения программы в некотором Представлении, когда ее содержание приблизилось к идеальному. То есть исчезновение недекларированного функционала из программы, и как следствие – уменьшение ее некорректного поведения, и есть удаление уязвимости.

Мутация уязвимости – изменение во времени содержания уязвимости; точка возникновения и исчезновения уязвимости является вырожденным случаем ее мутации. Последняя происходит из-за неточечных изменений программы, затрагиваемых в том числе уже содержащиеся в ней уязвимости.

Взаимодействие уязвимостей – эффект от пересечения в одном Представлении содержания нескольких уязвимостей. Так, например, внесение уязвимости в более позднем Представлении может повлиять на функционирование уязвимости, внесенной в более раннем.

Опишем возможные варианты взаимодействия уязвимостей, приводя реальные примеры для различных пар. Поскольку такие взаимодействия в принципе сложно воспринимаемы человеком (даже экспертом по безопасности программного кода), то для лучшей их интерпретации будем использовать антропоморфический подход, заключающийся в перенесении объектов и процессов на живую природу. В интересах этого сделаем следующие соответствия элементов IT-мира и мира живой природы (таблица 1).

Обоснованием такого приема может служить тот факт, что все производимое человеком так или иначе связано с окружающим физическим (в том числе, животным) миром, а также сделано по его подобию. И поэтому законы (в основном логики), используемые человеком, совпадают с законами этого мира. Правильным было бы в принципе описывать законы в абстрактном виде, а затем переносить их на любые области внутреннего и внешнего окружения человека – на физические объекты, мир живой природы или же информационный мир.

Согласно широко распространенному в науке делению отношений организмов известны следующие типы их взаимодействий: симбиоз (облигатный и факультативный симбиоз, комменсализм, паразитизм, хищничество) – когда хотя бы один из организмов получает выгоду, антибиоз (аменсализм, аллелопатия, конкуренция) – когда один из организмов ограничивает возможности другого, и нейтрализм – сосуществования организмов без взаимного влияния.

Таблица 1 – Антропоморфическое сопоставление IT-мира и мира живой природы

IT-мир	Мир живой природы
Информационные потоки	Информационные потоки внешней среды
Программа	Организм
Функция программы	Реакция организма на внешнюю среду
Форма программы	Микроэлементы
Содержание программы	Биологическая структура организма (из микроэлементов)
Представление программы	Состояние организма в некоторый период его жизни, качественно отличный от других
Идеальное содержание программы	Здоровый организм (без вирусов)
Уязвимость	Вирус (другой инфекционный организм)
Функция уязвимости	Нарушение работы организма путем неадекватного поведения во внешней среде, в т.ч. его саморазрушение (болезнь)
Возникновение уязвимость	Рождение вируса
Исчезновение уязвимость	Смерть вируса
Мутация уязвимости	Развитие вируса (эволюция)
Взаимодействие уязвимостей	Сожительство (симбиоз, антибиоз, нейтрализм) вирусов, как отдельных организмов

Опишем схемы и примеры взаимодействия уязвимостей, антропоморфически аналогичные приведенным типам взаимодействий организмов. В скобках типа укажем эффект от такого взаимодействия для каждого из участвующих вирусов/уязвимостей по порядковому номеру с разделителем «|»: «+» для выгоды, «–» для ущерба, «0» в случае отсутствия какого-либо эффекта.

Введем также для формализации взаимодействий следующие обозначения: V – уязвимость (*от англ.* Vulnerability); T – функционирование уязвимости или угроза (*от англ.* Threat); 0 – отсутствие угрозы; $T_X(T_Y)$ – использование для функционирования уязвимости X функционала уязвимости Y; $\rightarrow$ – реализация угроз вследствие наличия уязвимостей; $<, >, =$ – операции сравнение угроз по их возможному ущербу. Тогда, реализация обособленной угрозы N вследствие наличия в программе соответствующей уязвимости N может быть записана следующем образом:

$$V_N \rightarrow T_N. \quad (1)$$

Исключение из правил составляет случай, когда каждая уязвимость по отдельности не ведет к какой-либо угрозе (как будет показано далее, это верно для первого типа взаимодействий – облигатного симбиоза):

$$V_N \rightarrow 0. \quad (2)$$

Для визуализации взаимодействий используем графические элементы (обрабатываемая информация; атака на информацию с использованием уязвимости; уязвимость в коде; угроза информации, реализуемая в результате атаки), как представлено на рисунке 1.

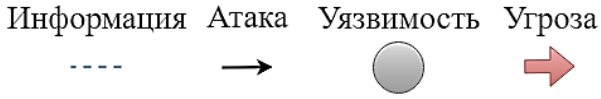


Рисунок 1 – Графические элементы для визуализации взаимодействия уязвимостей

Примечание. Ширина стрелки, обозначающая угрозу, соответствует значимости последней. Так, уменьшение ширины означает снижение риска проведения атаки с использованием уязвимостей, а увеличение – повышение риска.

Каждый их типов взаимодействия двух уязвимостей (с эффектами для каждой из них), описание из мира живой природы и IT-мира, формализованная запись, схема взаимодействия, а также конкретный пример приводятся далее.

Тип 1. Облигатный симбиоз (+|+).

Описание из живой природы. Данный тип характеризуется необходимостью совместного сосуществования организмов.

Описание из IT-мира. В IT-мире к данному типу могут быть отнесены уязвимости, каждая из которых по отдельности не может быть использована для проведения атак.

Схема взаимодействия уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 \to 0 \\ V_2 \to 0 \\ V_1 + V_2 \to T_{12} \end{cases} \quad (3)$$

и имеет вид, как представлено на рисунке 2.

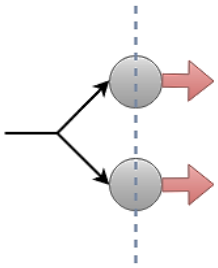


Рисунок 2 – Схема облигатного симбиоза уязвимостей (Тип 1)

Подобный тип взаимодействия может возникнуть между уязвимостями, когда каждая из них в полном смысле не приводит к какой-либо угрозе, поскольку они вынуждены действовать совместно; например, злоумышленник внедряет в программу свой деструктивный код частями, хотя каждую часть уже можно считать дефектом программы. Так, прослушивание некоторого порта и вызов деструктивных команд согласно сетевым пакетам на нем вполне может реализовать угрозы информации, хотя по отдельности эти уязвимости ни к чему не приведут – прослушивание порта само по себе бессмысленно для злоумышленника, а деструктивные команды не будут выполняться, поскольку без открытого порта они не дойдут до необходимого командного интерпретатора.

Тип 2. Факультативный симбиоз (+|+).

Описание из живой природы. Данный тип характеризуется взаимной выгодой от совместного сосуществования организмов, но без необходимости как таковой. В IT-мире к данному типу могут быть отнесены уязвимости, использование которых приводит к эффекту, большему чем сумма эффектов от использования уязвимостей по отдельности (так называемый эффект синергии, достаточно часто встречающийся во многих областях IT-мира [26–28]). Таким образом, уязвимости повышают риск проведения атак на информацию при использовании любой из них.

Схема взаимодействия уязвимостей формализуется как:

$$\begin{cases} V_1 + V_2 \to T_{12} \\ T_{12} > T_1 + T_2 \end{cases}. \quad (4)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 3).

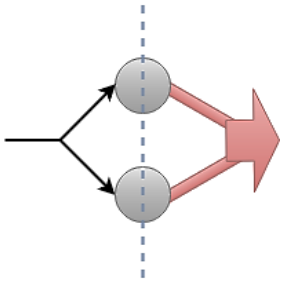


Рисунок 3 – Схема факультативного симбиоза уязвимостей (Тип 2)

Подобный тип взаимодействия может возникнуть между уязвимостями в виде встроенных по умолчанию логина и пароля администратора программы. По отдельности каждая из уязвимостей незначительно снижает общую безопасность (знание логина не избавляет от необходимости перебора пароля для входа в программу и наоборот). Однако совместно, эти уязвимости приводят к существенной угрозе информации, обрабатываемой в программе.

Тип 3. Комменсализм (+|0).

Описание из живой природы. Данный тип характеризуется выгодой от существования одного организма при отсутствии какого-либо эффекта для другого.

Описание из IT-мира. В IT-мире к данному типу могут быть отнесены уязвимости, одна из которых в присутствии другой может приводить к более существенному нарушению информационной безопасности.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \rightarrow T_1' + T_2 \\ T_1' > T_1 \end{cases}. \quad (5)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 4).

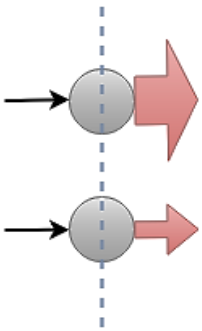


Рисунок 4 – Схема комменсализма уязвимостей (Тип 3)

Подобный тип взаимодействия может возникнуть между уязвимостью, дающей возможность полного перебора паролей (например, подавлением блокировки механизма аутентификации в случае превышения максимального количества попыток ввода) и уязвимостью, позволяющей не менять пароль по прошествии определенного времени (как правило, политики безопасности требуют обратного – периодической смены пароля). Наличие второй уязвимости упростит перебор с помощью первой – не будет необходимости произвести перебор в достаточно короткий промежуток времени. Возможность перебора пароля не повлияет на атаки посредством второй уязвимости, поскольку она снижает защиту в основном от подсматривания вводимого

пароля, его передачи в службу поддержки [29] или сотруднику на время болезни и от прочих социально-организационных [30], а также форс-мажорных ситуаций.

Тип 4. Паразитизм (+|–).

Данный тип характеризуется извлечением выгоды от сосуществования одним организмом, используя при этом другого как источник питания, среду обитания и т.п., возлагая на него часть своих отношений с внешней средой. В IT-мире к этому типу могут быть отнесены уязвимости, одна из которых использует принципы функционирования или непосредственно код другой (например, создаваемые информационные потоки или внедренные злонамеренные функции) для реализации собственных угроз обрабатываемой в программе информации.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \to T_1' + T_2' \\ T_1' = T_1 + D \\ T_2' = T_2 - D \end{cases}, \quad (6)$$

где D – выгода, извлекаемая первой уязвимостью из второй. Угроза от второй уязвимости не всегда снижается пропорционально D; тем не менее, такое упрощение не снижает общую корректность восприятия формулы.

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 5).

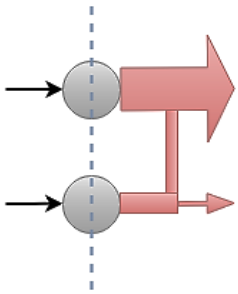


Рисунок 5 – Схема паразитизма уязвимостей (Тип 4)

Достаточно нетривиальным примером такого типа взаимодействия может стать следующий. Предположим, что в программе присутствует уязвимость по перехвату конфиденциальной информации (например, по причине слабого алгоритма шифрования или хэширования) без механизма непосредственной передачи ее злоумышленнику (например, информация сохраняется во временном каталоге, доступ к которому злоумышленник может получить лишь в редких случаях). В программе заложена уязвимость в виде рассылки сетевых пакетов без какого-либо значимого содержания (например, как часть BotNet сети для DDoS-атаки). Первая уязвимость может использовать механизмы сетевой рассылки второй для передачи конфиденциальной информации, которая естественно будет получена и злоумышленником. Эффективность рассылки второй уязвимостью может незначительно снизиться из-за использования сетевого канала первой уязвимостью.

Тип 5. Хищничество (+|–).

Данный тип характеризуется тем, что один организм питается частями другого при отсутствии каких-либо симбиотических (то есть взаимовыгодных) отношений и зачастую с умерщвлением первым второго. В IT-мире к данному типу могут быть отнесены уязвимости, одна из которых оказывает более агрессивное паразитирующее воздействие на другую путем подмены ее основного функционала на собственный с существенным использованием дополнительного; как правило, риск от использования первой уязвимости превосходит риск от второй.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \to T_1'(T_2) \\ T_1' > T_1 \end{cases}. \quad (7)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 6).

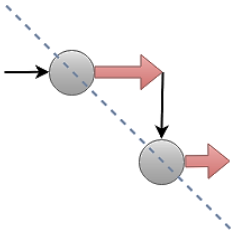


Рисунок 6 – Схема хищничества уязвимостей (Тип 5)

Примером взаимодействия может быть усиление функционала уязвимостей для случая паразитизма (описанного в Типе 5). Если второй внедренной в программу уязвимостью была рассылка сетевых пакетов в интересах DDoS-атаки, то первая уязвимость может подменить ядро второй на собственное – из перехватываемой конфиденциальной информации (для простоты предположим, что она лежит в памяти в открытом виде) будут создаваться специальные пакеты, используемые при сетевой рассылке. Первая уязвимость должна быть более агрессивной, чем простой перехват конфиденциальной информации и вызов функционала второй уязвимости – она должна именно подменять функционал уязвимости-жертвы; одной из таких уязвимостей может быть переполнение буфера, потенциально приводящее к записи «чужих» данных и кода. Итоговая комбинация будет состоять из механизма рассылки второй уязвимости, эксплуатация которого будет полностью осуществляться механизмом перехвата второй уязвимости. Полноценная DDoS-атака второй уязвимостью уже проводиться не будет (хотя бы потому, что программа не будет реагировать на команды центрального сетевого узла, управляющего работой всех участников атаки).

Тип 6. Нейтрализм (0|0).

Данный тип характеризуется отсутствием каких-либо воздействий друг на друга. В IT-мире к данному типу могут быть отнесены все уязвимости, которые никаким образом не влияют друг на друга, а функционируют полностью раздельно.

Взаимодействие уязвимостей формализуется следующим образом:

$$V_1 + V_2 \to T_1 + T_2 . \quad (8)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 7).

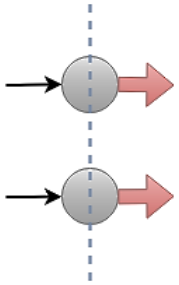


Рисунок 7 – Схема нейтрализма уязвимостей (Тип 6)

Подобный тип взаимодействия может возникнуть между любыми уязвимостями, по функциональному и целевому признаку не связанными друг с другом; например, между встроенным в программу логином администратора и ошибками в криптографическом алгоритме хэша для проверки пароля.

Тип 7. Аменсализм (0|–).

Данный тип характеризуется отрицательным влиянием одного организма на другого, не испытывая при этом какого-либо обратного влияния. В IT-мире к данному типу могут быть отнесены уязвимости, одна из которых оказывает сдерживающее действие на другую, снижая тем самым риск от использования последней.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \to T_1 + T_2' \\ T_2' < T_2 \end{cases}. \quad (9)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 8).

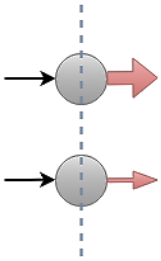


Рисунок 8 – Схема аменсализма уязвимостей (Тип 7)

Подобный тип взаимодействия возникает между уязвимостями, ведущими к угрозам нарушения доступности и конфиденциальности информации; суть второй – в выдаче конфиденциальной информации злоумышленнику, а первой – в предотвращении какой-либо выдачи информации в принципе путем захвата необходимых для этого ресурсов (например, сетевых портов). Например, вторая уязвимость может перехватывать пароли, обрабатываемые в программе, и возвращать их в стандартном ответе, а первая будет блокировать (или существенно замедлять) генерацию и отправку ответов из программы. Как результат, доступность информации в программе будет нарушена, а конфиденциальность – практически нет.

Тип 8. Аллелопатия (–|–).

Данный тип характеризуется взаимно-вредным влиянием организмов друг на друга. В IT-мире к данному типу могут быть отнесены уязвимости, использование которых приводит к эффекту, меньшему чем сумма эффектов от использования уязвимостей по отдельности (по аналогии с факультативным симбиозом – так называемый эффект диссинергии). Уязвимости снижают риск проведения атак на информацию при использовании любой из них.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \to T_{12} \\ T_{12} < T_1 + T_2 \end{cases}. \quad (10)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 9).

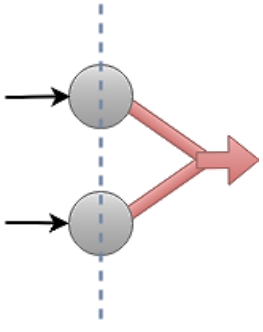


Рисунок 9 – Схема аллелопатии уязвимостей (Тип 8)

Подобный тип взаимодействия, как и в случае аллелопатии, может возникнуть между уязвимостями, ведущими к угрозам нарушения доступности и конфиденциальности информации. Однако в этом случае вторая уязвимость должна себя вести более агрессивно – не просто пытаться отправить конфиденциальную информацию злоумышленнику, а стремиться сделать это всеми возможными способами. Как следствие будет идти борьба (в том числе) каждой из уязвимости с другой – первая будет пресекать всякие попытки второй отправить информацию вне программы, а вторая искать все возможные пути обойти блокировки первой.

Тип 9. Конкуренция (–|–).

Данный тип характеризуется косвенным отрицательным влиянием организмов друг на друга по причине борьбы за общие ресурсы. В IT-мире к данному типу могут быть отнесены уязвимости, для функционирования которых необходимы общие программные ресурсы (память, место в хранилище, сетевые порты и пр.) и из-за чего уязвимости начинают конфликтовать.

Взаимодействие уязвимостей формализуется следующим образом:

$$\begin{cases} V_1 + V_2 \to T_1' + T_2' \\ \quad \begin{cases} T_1' \le T_1 \\ T_2' \ll T_2 \end{cases} \\ \quad \begin{cases} T_1' \ll T_1 \\ T_2' \le T_2 \end{cases} \end{cases} . \qquad (11)$$

Схема взаимодействия уязвимостей имеет следующий вид (рисунок 10).

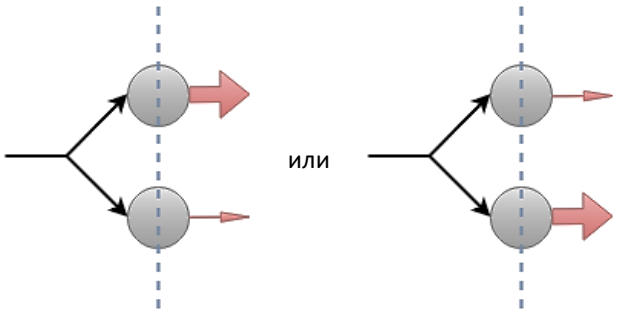


Рисунок 10 – Схема конкуренции уязвимостей (Тип 9)

Подобный тип взаимодействия может возникнуть между уязвимостями, каждая из которых стремится открыть сетевой порт для организации удаленного доступа злоумышленника (так называемый *backdoor*). Так, в случае ограниченного количества доступных портов (а точнее, одного порта) работоспособной окажется та уязвимость, которая его быстрее зарезервирует в сетевой библиотеке программы.

Составим сводную таблицу из взаимодействий всех вышеприведенных в примерах типов уязвимостей. Также, эти уязвимости были выбраны из списка наиболее актуальных (по мнению авторов) для standalone-программ (к которым применимы упомянутые Представления); таким образом, в данный список не входят характерные для распределенных систем (например, SQL-инъекция или слабости протоколов передачи), Web-сервисов (например, инъекция E-mail или межсайтовый скриптинг) и пр.

Таким образом, Топ-8 уязвимостей состоит из следующих.

У_1. Переполнение буфера – запись некоторым модулем данных и кода вне выделенного буфера – в память других модулей.

У_2. Встроенные по умолчанию логин или пароль – наличие в программе встроенных данных, позволяющих произвести несанкционированную идентификацию и/или аутентификацию.

У_3. Слабая реализация криптографических алгоритмов – подверженность криптографических алгоритмов взлому с кражей конфиденциальной информацией или отсутствие защиты последней как таковой.

У_4. Backdoor – намеренно встроенный дефект в алгоритме программы, позволяющий получить несанкционированный доступ к данным или удаленное управление ею.

У_5. Логическая бомба – скрыто внедренный фрагмент программы, позволяющий выполнить действия злоумышленника при возникновении определенных условий в программе.

У_6. Слабая парольная политика – ошибки или слабости в регламентах по управлению паролями, применяемыми в программе.

У_7. Исчерпание ресурсов – закачивание ресурсов, необходимых для работы программы, таких как память, процессорное время, сетевые порты, дисковое пространство, сетевой канал и пр.

У_8. Агент DDoS – встраивание в программу специального алгоритма, являющегося частью DDoS и выполняющего роль отправителя сетевых пакетов в интересах проводимой атаки (то есть «зомбирование» в интересах Botnet-сети).

Сводные данные взаимодействий уязвимостей из Топ-8 представлены в таблице 2.

Таблица 2 – Взаимодействия уязвимостей из Топ-8 (для описанных примеров)

	У_1	У_2	У_3	У_4	У_5	У_6	У_7	У_8
У_1								B_5 (–)
У_2		B_2 (+)	B_6 (0)					
У_3		B_6 (0)					B_7 (–) B_8 (–)	B_4 (–)
У_4					B_1 (+)			
У_5				B_1 (+)				
У_6						B_3 (+, 0)		
У_7			B_7 (0) B_8 (–)				B_9 (–,–)	
У_8	B_5 (+)		B_4 (+)					

В ячейках таблицы указаны типы взаимодействий (в виде B_N, соответствующего N-ному типу взаимодействия) для соответствующих типов уязвимостей в столбце (первая уязвимость в примере) и строке (вторая уязвимость в примере); также в скобках указан эффект для первой уязвимости от взаимодействия (в случае одинаковых типов уязвимостей указываются оба эффекта).

Несмотря на безусловный интерес (с научной точки зрения) к описанным взаимодействиям между уязвимостями, актуальность простого факта их существования без практической пользы будет крайне мала.

Для недопущения последнего рассмотрим с формальной точки зрения влияние таких взаимодействий на общую безопасность кода, свойства которой могут быть оценены с помощью различных метрик.

На сегодняшний день не существует каких-либо «безупречных» и широко распространённых метрик оценки программного кода в аспекте их безопасности.

В связи с этим предложим собственную метрику, соответствующую величине эффекта от эксплуатации того или иного типа уязвимости в коде. Метрика будет отражать не каждую конкретную уязвимость в коде, а их множество каждого из типов.

Такая метрика может быть полезна при сравнении и выборе программ, требования к которым связаны с источниками атак и рисками угроз – часть атак могут использовать только одни типы уязвимостей, а часть угроз – появиться в результате использования других типов.

В этом случае необходимо знать не конкретное количество уязвимостей и их местоположение в коде, а вероятность существования некоторого типа уязвимости и эффекта от ее использования. Так, с одной стороны, нецелесообразно использование программ с потенциальными уязвимостями в виде встроенных логинов и паролей в системах доступа к конфиденциальным данным с высокой вероятность проведения *brute-force* атак. С другой стороны уязвимость, приводящая к исчерпанию ресурсов процессора, окажется критичной в системах, где нарушение доступности информации недопустимо.

Очевидная и часто используемая неявно метрика уязвимостей программного кода (далее – Метрика) может быть записана следующим образом:

$$M = \begin{vmatrix} v_1 \\ \dots \\ v_N \end{vmatrix}, \tag{12}$$

где v_i – эффект от i-го типа уязвимости, N – количество типов уязвимостей.

Так, например, эффект от уязвимости переполнения буфера отсутствует в языках без конструкций явного выделения памяти или с проверками выхода за диапазон выделенного буфера памяти (C#, Java): $v_i \cong 0$. Очевидно, что v_i зависит от множества параметров, но основными из них является вероятность нахождения в коде данного типа уязвимости – e_i (как следствие свойств языка программирования, а не конкретной реализации программы), вероятность ее использования – u_i, и вероятность ее не обнаружения – n_i :

$$v_i \sim e_i \times u_i \times n_i. \tag{13}$$

Приведем сравнительные примеры этих параметров для различных групп языков – среднеуровневых (C, Pascal) и интерпретируемых (Ruby, Perl).

С одной стороны, возможность исчерпание ресурсов в языках с возможностью среднеуровневого программирования будет иметь меньший эффект, чем та же уязвимость для скриптовых языков – $e_i(C, Pascal) < e_i(Ruby, Perl)$, поскольку в последних это является следствием простоты написания кода.

С другой стороны, создание бэкдора на скриптовых языках обнаруживается более результативно, поскольку в них завуалировать вызовов открытия портов или вызовов внешних файлов существенно сложнее, чем в среднеуровневых языках – $n_i(C, Pascal) < n_i(Ruby, Perl)$. И, наконец, уязвимость встроенного логина и пароля по умолчания абсолютно одинаково будет использована в обеих рассмотренных группах языков.

Таким образом, даже примерное вычисление итоговой эффективности каждой из уязвимостей представляется достаточно нетривиальной задачей.

Рассмотрим теперь случаи программ с более, чем одной возможной уязвимостью.

Как было показано на многочисленных примерах, наличие двух уязвимостей одного или нескольких типов в одном коде может приводить к их взаимодействию с различными результатами для каждой из них.

Это может быть отражено путем добавления члена M_1 корректирующего показателя, учитывающего такие взаимодействия.

Очевидно, что M_1 должен быть вектором такой же размерности, как и M (то есть размерности N). При этом каждый из элементов вектора должен зависеть от эффекта соответствующего типа уязвимости, а также от ее взаимодействия с остальными; последнее, что может быть записано с помощью матрицы взаимодействия.

Итоговая формула для M_1 имеет следующий вид:

$$M_1 = |v_1, \dots, v_N| \times \begin{vmatrix} I_{11} & \dots & I_{1N} \\ \vdots & \ddots & \vdots \\ I_{N1} & \dots & I_{NN} \end{vmatrix}, \quad (14)$$

где I_{ij} – коэффициент влияния взаимодействия i-го и j-го типа уязвимости на i-ю уязвимость.

Таким образом, для невзаимодействующих типов уязвимостей их коэффициент $I_{ij} \equiv 0$, в случае усиления первой уязвимости второй $I_{ij} > 0$, а в случае подавления – $I_{ij} < 0$. Суммарный же эффект влияния всех типов уязвимостей на уязвимость данного типа (с индексом i) будет иметь достаточно сложный характер, определяемый как:

$$K_i = \sum_{j=1}^{N} v_j \times I_{ij}. \quad (15)$$

Развивая предложенный антропоморфический подход логично предположить, что могут существовать более редкие взаимодействия одновременно 3-х и более типов уязвимостей, для которых потребуется введение дополнительных корректирующих членов (соответственно, M_2, M_3 и так далее). Впрочем, по причине незначительности таких эффектов в первом приближении ими можно, скорее всего, пренебречь.

Таким образом, следствие применения антропоморфического подхода – перенесение взаимодействия организмов живой природы на такое же взаимодействие уязвимостей – может быть отражено не только на гипотетических примерах, но и в математическом виде.

Произведем расчет метрики уязвимостей, включающей корректирующий член взаимодействия уязвимостей для следующего примера гипотетической программы. Расчет значений (а точнее их взаимное соотношение) будем производить на основании достаточно большого опыта исполнителей настоящей НИР в IT-области.

Предположим, программа представляет собой простейшую операцию получения конфиденциальной информации из внутренней базы данных с использованием механизмов идентификации и аутентификации (то есть ввода логина и пароля). Очевидно, что в такой программе возможны нарушения конфиденциальности – когда один пользователь получил информацию другого, целостности – когда данным при получении пользователем были изменены, и доступности – когда пользователь не получил запрашиваемые данные. Будем считать, что программа написана на языке C# (и, соответственно, работает на платформе .Net). Также, программа предоставляет минимальный функционал: запросить логин и пароль у пользователя, найти соответствующую информацию, вернуть информацию пользователю.

Произведем расчет вероятностей эффекта каждой уязвимости v_i, считая вероятность использования u_i и не обнаружения n_i для каждой уязвимости равными, а вероятность нахождения e_i – согласно данным таблицы 3.

При этом, отсутствующая/низкая/средняя/высокая вероятности будут находиться в равных пропорциях, а именно: 0 : 0,33 : 0,66 : 1.

Таблица 3 – Вероятность нахождения уязвимостей в гипотетической программе

Тип	Вероятность	Обоснование
1	2	3
У_1	Отсутствует	Программа написана на C#, в котором присутствуют встроенные проверки выхода за пределы выделенного буфера.
У_2	Средняя	Программа может содержать встроенные логин и пароль для доступа к информации (например, случайно забытые разработчиками или оставленные ими злонамеренно).
У_3	Низкая	Криптографические библиотеки платформы .Net (используемые в C#) являются достаточно качественными и современными. Из-за отсутствия же работы с памятью несанкционированный доступ к конфиденциальной информации из других модулей практически невозможен.
У_4	Низкая	Поскольку программа является stand-alone offline (то есть функционирующей отдельно от каких-либо крупных систем и без использования сети), принимая на вход лишь логин и пароль пользователя, то полноценный backdoor в ней крайне маловероятен.
У_5	Средняя	Существует вероятность, что случайно или в злонамеренных целях в программу «заложена» логическая бомба, реагирующая на какой-либо встроенный логин или комбинацию логина/пароля (например, возвращение всех зарегистрированных логинов в системе или уничтожение данных).
У_6	Высокая	Поскольку в приведенный функционал программы не заложены требования касательно безопасности вводимых паролей – например, необходимость и возможность смены пароля, то такая политика заведомо является слабой.
У_7	Средняя	Язык C# (как и платформа .Net) позволяют писать код, менее оптимизированный по скорости и потребляемым ресурсам, чем, например, C/C++, поскольку оперируют более высокоуровневыми конструкциями и операциями (а, например, не буферами памяти и указателями).
У_8	Отсутствует	Поскольку программа не работает с сетью, то и агентом DDoS она выступать не может.

Таким образом, Метрика будет иметь следующий вид:

$$M = K_v \times \begin{vmatrix} 0 \\ 0.66 \\ 0.33 \\ 0.33 \\ 0.66 \\ 1 \\ 0.66 \\ 0 \end{vmatrix}, \qquad (16)$$

где K_v – нормировочный коэффициент ($K_v \ll 1$), поскольку очевидно, что таблица 3 позволяет судить лишь об отношении вероятностей нахождения уязвимостей друг между другом, а не об их абсолютных значениях.

Аналогично, будем считать, что коэффициент влияния взаимодействия i-го и j-го типа уязвимости на i-ю уязвимость I_{ij} равен произведению K_I (нормировочный коэффициент) на эффект от взаимодействия: «0» – в случае отсутствия эффекта, «1» – для выгоды, «–1» – для ущерба; в случае нескольких эффектов они суммируются.

Вычислим корректирующий член взаимодействия уязвимостей (пустые клетки в матрице означают, что в приведенных примерах отсутствуют взаимодействия, что тождественно нулевому значению соответствующей ячейки):

$$M_1 = K_v \times |0, 0.66, 0.33, 0.33, 0.66, 1, 0.66, 0| \times K_i \times M_1 = K_v \times K_i \times \begin{vmatrix} 0 \\ 0.66 \\ -0.66 \\ 0.66 \\ 0.33 \\ 1 \\ -1.98 \\ -0.33 \end{vmatrix} \quad (17)$$

где $M_1 =$

							1
	1						
						1−1	1
				1			
			1				
					1+0		
		−1				1−1	
1		1					

Несмотря на то, что коэффициенты K_v и K_i в данном примере никак не определены, тем не менее, из (5) можно сделать выводы касательно поправок к каждому типу уязвимостей (относительно других типов). Во-первых, никакого дополнительного эффекта от взаимодействия для переполнения буфера не будет (1-й компонент вектора равен 0). Во-вторых, будет существенное снижение эффектов уязвимостей исчерпания ресурсов (7-й компонент вектора равен –1.98). В-третьих, эффект уязвимости слабой парольной защиты будет значительно увеличен (6-й компонент вектора равен 1). Эффекты же остальных уязвимостей будут незначительно увеличены (2-й, 4-й компоненты вектора равен 0,66, а 5-й компонент равен 0,33) или уменьшены (3-й и 8-й компоненты вектора равны – 0,33). Таким образом, при нейтрализации уязвимостей в рассматриваемой гипотетической программе особое внимание необходимо уделить именно парольной политике; уязвимость исчерпания ресурсов гипотетически даже может самонейтрализоваться (это крайне маловероятно на практике, но для гипотетического примера теоретически возможно – и поэтому представляет чисто научный интерес).

Мега- и мета-уровень

Несмотря на текущий уровень понимания и решения вопросов обеспечения киберустойчивости ИС, как ЦИИ, комплексный учет различных факторов инфраструктурного, технического, экономического и регулятивного характера для поддержки принятия решений во время управления эксплуатацией объектов ЦИИ требуют дальнейшего исследований.

Это обосновано, в первую очередь, изменениями, происходящими в условиях эксплуатации объектов ЦИИ, подтверждающимися увеличением значимости межэлементного взаимодействия при обеспечении безопасности цифровой информационной инфраструктуры на фоне отсутствия соответствующих методов и методик идентификации и оценки.

Кроме того, исходными данными для аттестации ИС в ЦИИ, согласно [31], являются модели угроз безопасности информации, при разработке которых «учитываются структурно-

функциональные характеристики информационной системы, включающие структуру и состав информационной системы, физические, логические, функциональные и технологические взаимосвязи между сегментами информационной системы».

Согласно исследованию, выполненному Максимовой Е.А. [32], в качестве одного из результатов функционирования ЦИИ при активных межобъектных и межсубъектных связях определенного характера выявлен феномен ее саморазрушения как системы. Данное явление обозначено выше, как «инфраструктурный деструктивизм».

Межобъектные связи в ЦИИ являются, с одной стороны, элементом катаболизма информационной инфраструктуры, с другой стороны, одним из средств достижения цели управления киберустойчивостью объектов ЦИИ – создание условий для обеспечения ИБ ЦИИ. При этом, так как взаимодействующие субъекты ЦИИ могут обеспечивать реализацию разных услуги на уровнях разных территориальных органов, то это будет определять мультипликативность межобъектного взаимодействия при оценке показателей управляющих воздействий.

Информационные системы ЦИИ представляют собой однородные системы, информационно-технологическое взаимодействие между которыми можно рассматривать как взаимодействие программных систем. Таким образом, можно констатировать, что на инфраструктурном уровне саморазрушение системы может происходить без реализации атаки, то есть без воздействия на систему злоумышленника – субъектной личности. В это же время, субъектная личность может выступать в роли злоумышленника, если будет создавать условия развития ИД на уровне создания уязвимостей ИГ.

Следовательно, ИД можно рассматривать как новый класс (имманентных) угроз киберустойчивости – угрозы ИБ инфраструктурного генезиса. В отличие от «классической» угрозы ИБ, инфраструктурный деструктивизм носит ярко выраженный разрушающий характер.

Важным вопросом в исследовании феномена ИД остается интенсионал его генезиса, т.е. вопрос об уязвимостях ИГ.

Выделим следующие виды источников ДВ ИГ – уязвимостей (У) инфраструктурного генезиса.

У_ИГ1 – ошибки при анализе требований к организации работы субъекта ЦИИ.

У_ИГ2 – ошибки инфраструктурного анализа. Инфраструктурный анализ определяет взаимосвязь и взаимовлияние объектов ЦИИ. Последствия определяются результатом влияния атаки на неатакованные, но связанные с атакованным объектами ЦИИ. Ошибки в инфраструктурном анализе влияют на оценку рисков киберустойчивости.

У_ИГ3 – ошибки в проектировании СЗИ субъекта ЦИИ. Следует выделить следующие возможные ошибки: при аттестации ИС ЦИИ; в формировании перечня требований по обеспечению безопасности объектов ЦИИ; при формировании мер для перекрывания требований по обеспечению безопасности объектов ЦИИ.

У_ИГ4 – ошибки при реализации СЗИ объекта ЦИИ.

У_ИГ5 – ошибки при внедрении системы защиты объекта ЦИИ. У_ИГ4 может привести к У_ИГ5.

У_ИГ6 – ошибки при сопровождении объекта ЦИИ.

Исходя из У_ИГ1 – У_ИГ6 возможна реализация атаки на субъекте ЦИИ (обозначим как У_ИГ7). Наиболее вероятно проявление У_ИГ7 в случае наличия ошибок в СЗИ субъекта ЦИИ (У_ИГ 4) при не перекрытых уязвимостях.

Исследование вопроса поведения субъекта ЦИИ под воздействием ДВ ИГ невозможно без анализа существующих в данной системе взаимосвязей. На межсубъектном уровне решается путем реализации последовательности шагов.

Шаг 1. Определение решаемых взаимодействующими субъектами ЦИИ задач и заданных их функционалов.

Шаг 2. Определение вида структурной межсубъектной связи.

Шаг 3. Определение типа межсубъектного отношения.

Шаг 4. Определение уровня межсубъектного отношения.

Шаг 5. Формализация (формулирование характеристического свойства) межсубъектного отношения.

Шаг 6. Определение класса межсубъектного отношения в рамках решаемых субъектами ЦИИ задач и заданных функционалов.

Шаг 7. Оценка значимости межсубъектного отношения в рамках решаемых субъектами ЦИИ задач и заданных функционалов.

Шаг 8. Идентификация межсубъектного отношения как источника ИД.

Исходя из структурного состава ЦИИ, возможно выделить межобъектные взаимодействия на уровне объектов ЦИИ, принадлежащих разным субъектам ЦИИ (гетерогенная система) и на уровне объектов одного субъекта ЦИИ (моносубъектная система). Возможна также ситуация комбинированной формы взаимодействия.

Виды связей в структуре этих систем подробно рассмотрены в [32], а в качестве основных выделены следующие: жесткие и гибкие, главные и второстепенные, внутренние и внешние; прямые, обратные и комбинированные связи. Для выявления и идентификации межобъектных связей как источника ИД на субъекте ЦИИ, необходимо сформировать более чувствительную (точки зрения рассмотрения полного спектра вариантов межобъектного взаимодействия на субъекте ЦИИ) классификацию. Задача разрешима с использованием вышеупомянутого антропоморфического подхода.

Предложенная М.В. Буйневичем и К.Е. Израиловым в [33, 34] классификация взаимодействия уязвимостей ПО является наиболее «чувствительной» и способной достаточно полно отражать всевозможные варианты межобъектного взаимодействия в ЦИИ. Использование предложенного подхода позволяет рассмотреть возникающий на субъекте ЦИИ синергетический эффект и определить момент появления точки бифуркации. Чувствительность антропоморфических форм межобъектных связей к возможным изменениям в системе в [32] обозначена как бифуркационный эффект (таблица 4).

Таблица 4 – Наличие бифуркационного эффекта при различных формах межобъектного взаимодействия в ЦИИ

Форма межобъектного взаимодействия в ЦИИ	Бифуркационный эффект от связи объект 1 / объект 2	Новая связь – взаимодействие как вид ДВ	Комментарий
Симбиоз			
Облигатный симбиоз	– / –	–	Взаимовыгодно
Факультативный симбиоз	– / –	–	Взаимовыгодно, но не обязательно
Комменсализм	– / –	–	Односторонняя выгода, нейтрально для второго
Нейтрализм	– / –	–	Взаимонейтрально
Антибиоз			
Аменсализм	+ / –	+	Одностороннее отрицательное, для второго – нейтрально
Аллелопатия	+ / +	+	Взаимонегативно
Конкуренция	+ / +	+	Взаимовраги

Примечание. Используются следующие обозначения: «+» – наличие признака, «+/–» – отсутствие признака у одного из объектов, «+/+» – наличие признака у обоих объектов.

Для выявления и идентификации межобъектных связей как источника ИД на субъекте ЦИИ предлагается последовательности шагов, которую можно считать соответствующим методом.

Шаг 1. Построение архитектуры информационной инфраструктуры субъекта ЦИИ.

Шаг 2. Обозначение режима функционирования субъекта ЦИИ.

Шаг 3. Определение наличия связей между объектами субъекта ЦИИ.

Шаг 4. Определение антропоморфических форм межобъектных связей, определенных на Шаге 3; для этого используется экспертная оценка.

Шаг 5. Определение наличия и вида биффуркационного эффекта по каждой исследуемой межобъектной связи;

Шаг 6. Выделение межобъектных связей антибиотического вида в форме аллелопатия и конкуренция. Переход на Шаг 10 (по условию).

Шаг 7. Выделение межобъектных связей антибиотического вида в форме аменсализма.

Шаг 8. По результатам Шагов 6 и 7 формирование базы данных межобъектных связей – источников ИД на субъекте ЦИИ; переход на Шаг 11 (по условию).

Шаг 9. По результатам Шага 7 выделение объектов – источников межобъектных связей в форме «аменсализм».

Шаг 10. Выделение систем взаимодействующих объектов, элементы которых являются объектами-участниками по результатам Шагов 6 и 8; данные системы взаимодействующих объектов являются системами, содержащими деструктивно-образующие элементы.

Шаг 11. Рассмотрение вопроса об уточнении класса защищенности взаимодействующих объектов (ИС ЦИИ – результатов Шагов 8 и 10).

Предложенный метод позволяет выявить и идентифицировать межобъектные связи как источники ИД.

Так как один из этапов работы по обеспечению киберустойчивости ЦИИ связан с определением классов защищенности объектов ЦИИ, то рассмотрим, как наличие деструктивно-образующих связей влияет на их уточнение.

Важно отметить, что в данном контексте интересны не только виды межобъектных связей, но и последствия от их влияния на инфраструктурную функциональность субъекта ЦИИ.

На пуле полученных оригинальных моделей и методов выявления и идентификации межсубъектных и межобъектных связей как источника ИД в ЦИИ стало возможным решение целого ряда наукоемких прикладных задач ИБ, например: оценка инфраструктурной устойчивости субъектов ЦИИ [35] или проактивное управление ИБ субъектов ЦИИ как сложных организационных систем с динамически изменяющейся структурой [36].

Макро-уровень

На Макро-уровне под эффектами ИД понимаются результаты взаимодействия минимум двух сервисов ИС, например Веб-серверов (что в общем случае может быть расширено и на их произвольное количество) [37], потенциально приводящие к изменению с различным знаком итоговой эффективности функционирования системы. Деление эффектов на типы здесь также осуществляется по аналогии с типизацией на микро-уровне, следующим образом [38].

Тип 1. Факультативный симбиоз – второй Веб-сервер приводит к улучшению производительности обработки запросов на первом (за счет кэширования); при этом, первый Веб-сервер также улучшает производительность обработки запросов на втором, но только при условии совместной обработки запросов.

Тип 2. Комменсализм – второй Веб-сервер приводит к улучшению производительности запросов на первом; при этом, первый Веб-сервер не влияет на кэширование запросов второго Веб сервера без кэширования запросов на первом;

Тип 3. Нейтрализм – как первый, так и второй Веб-сервера не влияют на производительность обработки запросов друг друга.

Тип 4. Облигативный симбиоз – второй Веб-сервер приводит к улучшению производительности обработки запросов на первом (за счет кэширования); при этом, первый Веб-

сервер также улучшает производительность обработки запросов на втором, но без обязательного условия совместной обработки запросов.

Тип 5. Паразитизм – второй Веб-сервер приводит к ухудшению производительности запросов на первом, повышая тем самым собственную производительность обработки запросов.

Тип 6. Хищничество – второй Веб-сервер приводит к ухудшению производительности запросов на первом, повышая собственную производительность обработки запросов в безвозвратном режиме до перезагрузки инфраструктуры.

Тип 7. Аменсализм – второй Веб-сервер ухудшает производительность обработки запросов на первом; при этом, первый Веб-сервер не влияет на производительность обработки запросов на втором.

Тип 8. Конкуренция – второй Веб-сервер приводит к ухудшению производительности обработки запросов на первом, повышая тем самым собственную производительность, однако исходная производительность восстанавливается без перезагрузки инфраструктуры.

Тип 9. Аллелопатия – второй Веб-сервер приводит к ухудшению производительности обработки запросов на первом; при этом, первый Веб-сервер также ухудшает производительности обработки запросов на втором.

Для научного обоснования и доказательства работоспособности антропометрического подхода к выявлению эффектов ИД на макро-уровне Русаковым А.М. разработана частная методика проверки (апробации) распределенной системы ситуационного мониторинга (РССМ) на базе сетевой инфраструктуры основные процедуры которой и результаты апробации (проверки) представлены в разделе 3 настоящей статьи.

Протокол-ориентированный подход к выявлению и идентификации источников деструктивных воздействий инфраструктурного генезиса

Еще одним из источников ДВ ИГ можно считать информационно-техническое взаимодействие (протокол) между подсистемами защиты информации при их интеграции в единую систему [44, 45].

Если классические модели угроз (с конкретным списком нарушителей, объектами воздействия и базой накопленных инцидентов) достаточно хорошо описываются, то такой новый вид модели угроз – имманентных (свойственных самой системе, исходящих и действующих внутри нее) – оставлен практически без внимания.

Такое положение дел является недопустимым, поскольку современная ситуация в области информационной безопасности привела к необходимости объединения и тесного взаимодействия (т.е. интеграции) качественно разных подсистем защиты информации, что (как показывает теоретическое прогнозирование и практическое функционирование) ведет к множеству диссинергетических эффектов [46], не только снижающих итоговую защиту информации, но и попросту приводящих к отказу всей интегрированной системы.

В интересах этого далее будет предложена модель угроз (Модель) информационно-технического взаимодействия подсистем интегрированной СЗИ, схема которой подробно описана в работах [47, 48].

Прежде чем перейти к описанию Модели, введем основополагающие понятия предметной области – (источников) угроз от интеграции подсистем в единую СЗИ.

Интеграция – объединение и обеспечение тесного взаимодействия подсистем в рамках единой системы для достижения общей цели.

Цель – направление противодействия (для подсистем и интегрированной СЗИ) заданному множеству классов атак на защищаемые информационные ресурсы.

Информационный ресурс – элемент в информационной системе, подверженный атакам, на противодействие которым направлена деятельность СЗИ.

Класс атаки – класс, определяемый среду, через которую проводится атака на информационный ресурс; ранее были выделены следующие классы:

1) программно-математические (например, сетевые атаки и ошибки в программном обеспечении);

2) перемещением (например, нарушение контрольно-пропускного режима);

3) через физические поля (например, посредством технических каналов утечки и силового деструктивного воздействия);

4) природные;

5) техногенные (например, пожары и аварии).

Подсистема СЗИ – часть СЗИ после интеграции, изначально предназначенная для достижения собственной цели, как правило, по противодействию некоторому классу атак.

Модуль – элемент подсистемы СЗИ, решающий определенную задачу (в рамках достижения подсистемой собственной цели).

Задача – ситуация, на разрешение которой направлено функционирование модуля; может быть декомпозирована на последовательность определенных действий.

Информационно-техническое взаимодействие (ИТВ) – взаимодействие модулей подсистем в рамках работы в интегрированной СЗИ [49, 50].

Протокол – формализованный набор данных и алгоритмов их обмена, обеспечивающих ИТВ модулей подсистем СЗИ [51].

Информационный объект – минимальная единица данных протокола, которыми обмениваются модули в процессе ИТВ [52]; имеет определенный тип.

Воздействие – сигнал, поступающий на информационный ресурс, потенциально являющийся частью атаки и который анализируется СЗИ.

Противодействие – мероприятие, синтезируемое СЗИ, направленное на защиту информационного ресурса от обнаруженной атаки.

Также интерпретируем основные элементы «классической» Модели угроз в контексте рассматриваемой предметной области.

Угроза – совокупность условий и факторов (уязвимостей и источников угроз), создающих потенциальную опасность нарушений интегрированной СЗИ.

Эффективность СЗИ – совокупность показателей результативности, оперативности и ресурсоэкономности СЗИ, определяемых соответственно, как доля отражаемых атак, скорость их отражения атак и размер сохраненных ресурсов [47, 48].

Уязвимость системы (пассивный источник) – некая особенность подсистем СЗИ (рассматриваемых как по отдельности, так и в виде единого целого), которая после проведения процесса их интеграции в единую систему может «эксплуатироваться» источником угрозы.

Источник угрозы (активный источник) – фактор, угрожающий интегрированной СЗИ за счет «эксплуатации» ее уязвимостей.

Способ реализации угрозы – создание ситуации, при которой потенциальная опасность переводится в реальную.

Объект воздействия – элемент интегрированной СЗИ, на который непосредственно направлена угроза.

Последствие от реализации угрозы – результат (как правило, негативный) реализации угрозы для интегрированной СЗИ.

Нарушение внутренней информационной безопасности системы – нарушения, относящиеся к информационной безопасности интегрированной СЗИ без учета нарушений защищаемых ею информационных ресурсов.

Нарушение работоспособности системы – нарушения, относящиеся к работоспособности интегрированной СЗИ без учета нарушений информационной безопасности.

Работоспособность здесь рассматривается, как способность ИСЗИ целенаправленно функционировать на заданном уровне эффективности в течение определённого времени.

Мера противодействия – мера, направленная на источник или уязвимость с целью частичной или полной их нейтрализации; представляет собой требование к протоколу ИТВ, на базе которого строится интеграция подсистем СЗИ [53].

Исходя из специфики предметной области, процесс возникновения угроз ИТВ в интегрированной СЗИ и противодействия им со стороны последней можно описать с помощью следующей онтологической модели (рисунок 11).

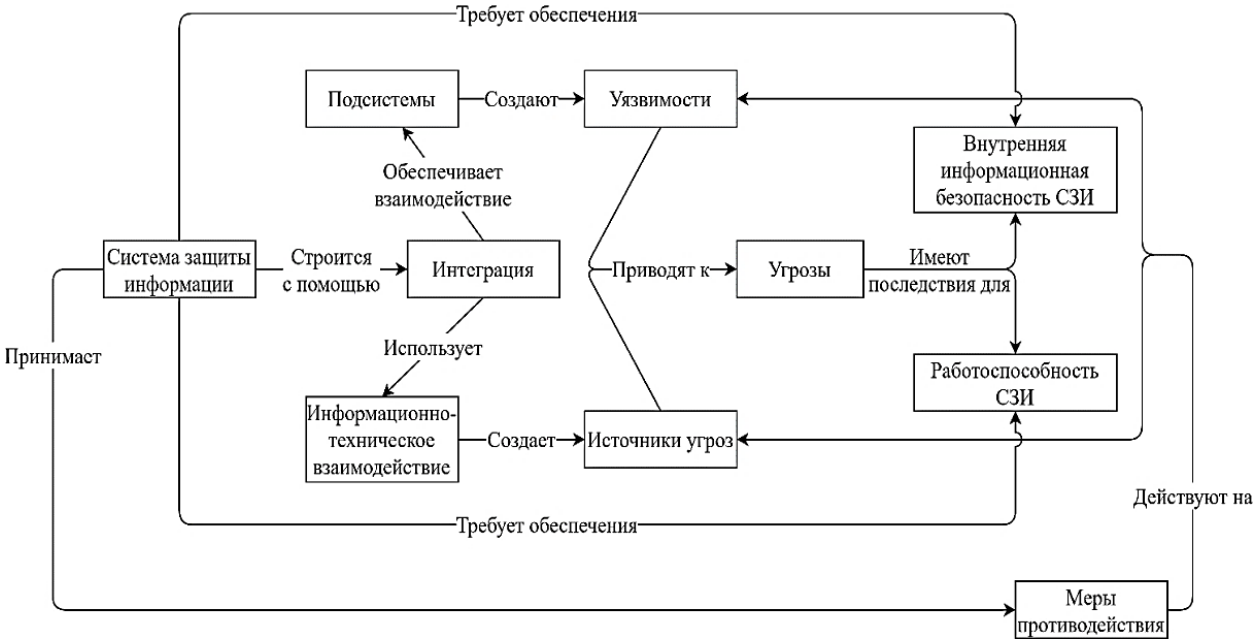


Рисунок 11 – Схема возникновения угроз ИТВ в интегрированной СЗИ и противодействия им

Согласно схеме онтологической модели [54, 55], построение СЗИ из качественно различных подсистем осуществляется с помощью процесса их интеграции в единую.

Как результат, обеспечивающее их совместную работу ИТВ ведет к возникновению источников угроз; подсистемы же начинают обладать рядом уязвимостей, также «проявляющихся» в результате интеграции. Эти источники угроз, «эксплуатируя» уязвимости, с неизбежностью приводят к угрозам, которые в рассматриваемом аспекте имеют последствия для нарушения не только внутренней информационной безопасности интегрированной СЗИ, но и ее общей работоспособности. Для противодействия этому СЗИ должна применять меры, направленные как на источники угроз, так и на «эксплуатируемые» ими уязвимости.

Результатами исследований вышеупомянутой научной школы и накопленного опыта по созданию и эксплуатации комплектных систем обеспечения информационной безопасности и защиты информации, а также в соответствии с предложенной онтологической моделью, стали следующие 6-ть основных угроз, приводящих к нарушениям работоспособности и безопасности СЗИ после интеграции в нее различных подсистем, использующих для ИТВ различные протоколы [56].

Рассмотрим их в канонической нотации: источник (активный источник) и уязвимость (пассивный источник) как факторы и условиях появления угрозы, способ ее реализации, объект атаки, последствия от реализации в аспектах нарушения внутренней ИБ и ущерба работоспособности интегрированной СЗИ.

Отметим, что источниками (активными) всех угроз является ИТВ, обеспечивающее взаимную работу подсистем после интеграции; уязвимости же (пассивные источники) для каждой угрозы свои.

Угроза 1. Создание для интегрированной СЗИ опасности, связанной с нарушением доступности ее информации и снижением результативности работы вследствие нехватки информационных объектов.

Уязвимость представляет собой «информационный дефицит функционирования» или «информационный голод» модулей, означающие нехватку информации для их работы. Способ реализации угрозы заключается в создании ситуации, когда информационных объектов, создаваемых одними модулями, оказывается недостаточно для решения задач другими модулями.

Объектом атаки выступают данные в виде информации, используемой модулями для обмена.

Последствия от реализации угрозы в аспекте внутренней информационной безопасности состоят в том, что информация может не доходить до модуля-получателя, что означает нарушение ее доступности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в некорректном противодействии (или невозможности противодействия) атакам, что означает снижение результативности интегрированной СЗИ.

Мерой противодействия данной угрозе выступает обеспечение модулей необходимыми информационными объектами, что означает требования к протоколу ИТВ в части обеспечения необходимости информационных объектов.

Угроза 2. Создание для интегрированной СЗИ опасности, связанной с нарушением конфиденциальности ее информации и снижением оперативности и ресурсоэкономности работы вследствие избытка информационных объектов.

Уязвимость представляет собой излишнюю функциональность модулей, означающую избыток информации для их работы.

Способ реализации угрозы заключается в генерации невостребованных («излишних») информационных объектов, т.е. когда информационные объекты, создаваемые одними модулями, оказываются невостребованными при решении задач другими модулями.

Объектом атаки выступают данные в виде информации, используемой модулями для обмена.

Последствия от реализации угрозы в аспекте внутренней ИБ состоят в том, что информация может распространяться сторонним получателям, что означает нарушение ее конфиденциальности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в бесполезных ресурсо-временны́х затратах, что означает снижение оперативности и ресурсоэкономности интегрированной СЗИ.

Мерой противодействия данной угрозы выступает недопущение появления избыточных информационных объектов, что означает требования к протоколу ИТВ в части обеспечения достаточности информационных объектов.

Угроза 3. Создание для интегрированной СЗИ опасности, связанной с нарушением целостности ее информации и снижением оперативности и ресурсоэкономности работы вследствие внутренних конфликтов модулей при решении одинаковых задач.

Уязвимость представляет собой дублирование задач (а, следовательно, и функционала), решаемых различными модулями СЗИ, означающее «смешивание» генерируемой ими информации.

Способ реализации угрозы заключается в негативной конкуренции модулей (приводящей, очевидно, к внутренним конфликтам) при решении одной и той же задачи, т.е. когда модули получают и создают одни и те же информационные объекты.

Объектом атаки выступает логика в виде схемы взаимосвязи между модулями.

Последствия от реализации угрозы в аспекте внутренней ИБ состоят в том, что информация может обрабатываться различными модулями в интересах решения одной задачи (т.е. генерируя один набор выходных объектов на одном наборе входных объектов), что означает нарушение ее целостности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в ощутимо бесполезных ресурсо-временных затратах, что означает снижение оперативности и ресурсоэкономности интегрированной СЗИ.

Мерой противодействия данной угрозы выступает обеспечение уникальности модулей, что означает требования к протоколу ИТВ в части обеспечения специализации модулей.

Угроза 4. Создание для интегрированной СЗИ опасности, связанной с нарушением доступности ее информации и результативности работы вследствие отсутствия возможности поддержки модулями входных классов атак.

Уязвимость представляет собой несоответствие классов атак СЗИ классам ее подсистем, означающее отсутствие противодействия одному из таких классов.

Способ реализации угрозы заключается в создании ситуации, когда СЗИ не способна анализировать необходимый набор входных классов атак, т.е. когда модули не реагируют на весь набор внешних воздействий.

Объектом атаки выступает совокупность поддерживаемых классов атак.

Последствия от реализации угрозы в аспекте внутренней ИБ состоят в том, что информация, необходимая для обработки атакующих воздействий, не будет поступать на соответствующие модули-анализаторы, что означает нарушение ее доступности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в том, что она не может воспринимать внешние сигналы атакующих воздействий, что означает снижение ее результативности.

Мерой противодействия данной угрозы выступает наличие модулей без входных информационных объектов (т.е. воспринимающих воздействия снаружи системы), что означает требования к протоколу ИТВ в части обеспечения входной терминальности модулей.

Угроза 5. Создание для интегрированной СЗИ опасности, связанной с нарушением доступности ее информации и результативности работы вследствие отсутствия возможности поддержки модулями выходных противодействий классам атак.

Уязвимость полностью тождественна «эксплуатируемой» при реализации Угрозы 4.

Способ реализации угрозы заключается в создании ситуации, когда СЗИ не способна синтезировать необходимый набор противодействий соответствующим классам атак, т.е. когда модули не создают весь набор внешних команд для нейтрализации атак.

Объектом атаки выступает совокупность поддерживаемых классов атак.

Последствия от реализации угрозы в аспекте внутренней ИБ состоят в том, что информация, необходимая для противодействия атакующим воздействиям, не будет поступать от соответствующих модулей-синтезаторов, что означает нарушение ее доступности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в том, что она не может осуществлять внешние мероприятия по противодействию атакам, что означает снижение ее результативности.

Мерой противодействия данной угрозы выступает наличие модулей без выходных информационных объектов (т.е. генерирующих команды наружу из системы), что означает требования к протоколу ИТВ в части обеспечения выходной терминальности модулей.

Угроза 6. Создание для интегрированной СЗИ опасности, связанной с нарушением конфиденциальности и целостности ее информации, а также всех показателей эффективности ее работы вследствие наличия невзаимодействующих модулей.

Уязвимость представляет собой «превосхождение» целями подсистем единой цели СЗИ, означающее некоторое «целевое или задачное излишество» в реализации результирующей интегрированной СЗИ.

Способ реализации угрозы заключается в активности модулей, не участвующих в работе интегрированной СЗИ (по причине ориентированности на решение собственных внесистемных задач), т.е. когда некоторые модули не несут полезного для СЗИ функционала.

Объектом атаки выступает состав и функциональность модулей.

Последствия от реализации угрозы в аспекте внутренней ИБ состоят в том, что модули, бесконтрольно действующие в СЗИ, хотя и участвующие в ИТВ «нелегально», могут тем не менее как распространять конфиденциальную информацию в одни элементы СЗИ (или вне ее), так и

мешать работе других элементов, «зашумляя» их лишней информацией; это ведет к нарушению конфиденциальности и целостности.

Последствия от реализации угрозы в аспекте работоспособности СЗИ состоят в наличии вырожденных модулей, не задействованных полностью в работе СЗИ, приводящих как к существенным бесполезным ресурсо-временным затратам, так и к их слабому внутреннему контролю со стороны СЗИ; это ведет к снижению всех показателей ее эффективности.

Мерой противодействия данной угрозы выступает недопущение в составе СЗИ «лишних» модулей в виде выполнения требования к протоколу ИТВ, что означает требования к протоколу ИТВ в части обеспечения задействованности модулей.

Отличиями данной модели от типовых, как правило, используемых в организациях, являются следующие.

Во-первых, источником угроз является не конкретный объект или субъект, а собственно ИТВ (само по себе являющееся достаточно сложной сущностью [57, 58]), необходимое для осуществления интеграции подсистем в единую, являющейся неотъемлемой и неотделимой частью результирующей СЗИ.

Во-вторых, результатами реализации угроз являются нарушения самой СЗИ, а не защищаемого ею информационного ресурса, влияющие как на ее информационную безопасность, так и на общую работоспособность. Основным недостатком предложенной модели является высокий уровень абстракции, сложность интерпретации входящих в модель элементов и неочевидные причинно-следственные взаимосвязи.

Продолжением исследования должна стать детализация угроз модели, понижающая уровень их абстракции от обобщенного для всех СЗИ до частного [59-64] для конкретных реализаций последних.

Анализ и прогнозирование временных рядов инцидентов информационной безопасности в цифровой информационной инфраструктуре: возможности и ограничения методов

В триаде основных типов задач, решаемых учеными-исследователями, а именно – анализе, моделировании и прогнозировании, последняя является наиболее сильным вариантом постановки исследовательской проблемы [65].

Существует большое количество методов прогнозирования, качество которых зависит от имеющихся исходных данных. Если данные о прошлом представлены в числовой форме, а также имеются некоторые предположения, что выявленная на основе исследования ретроспективных данных тенденция может быть пролонгирована, то в этом случае используются количественные методы и, в частности, методы теории временных рядов.

За последние годы в ней разработано большое количество методов, алгоритмов и моделей, издается международный журнал прогнозирования, опубликовано множество книг, например [66, 67]. Создано значительное число пакетов прогнозирования для языков Python, R, что позволяет автоматизировать решение задач прогнозирования. Существующие статистические пакеты и графические надстройки, например, gretl, Loginom, JASP, jamovi позволяют в ходе исследования применять low-code, no-code подходы.

Популярность количественных методов прогнозирования увеличивается с возрастанием количества наборов данных, их размера (десятки гигабайт, например Kaggle), созданием специальных репозиториев. Только для решения задач прогнозирования на момент написания статьи [68], опубликованной в рамках НИР «Кибермониторинг», их число превысило 8000. Более 500 датасетов посвящено проблемам кибербезопасности. Набор данных машинного обучения Калифорнийского университета UC Irvine Machine Learning Repository содержит 90 временных рядов, позволяющих решать задачи прогнозирования.

Анализ данных различной природы, включая и временные ряды, основан на модели, предложенной Дж. Тьюки, который утверждал, что «не метод определяет схему исследования, а характер данных». Вместо традиционно используемой последовательности исследования «модель – анализ – данные – результат», им была предложена схема «данные – разведочный

анализ – модель – подтверждающий анализ». Первичной в этой схеме являются данные: их характер, используемые шкалы, объем, учет времени, качество, – все это определяет выбор инструмента исследования.

Поэтому большинство публикаций, посвященных решению задач прогнозирования, в том числе инцидентов ИБ (в нашем случае – кибератак), непосредственно связано с характером исследуемых данных, наличием в них тренда, сезонных составляющих, характера случайной компоненты и др.

Так, например, в статье [69] исследуется динамика кибератак на веб-сервисы корпоративной сети, в том числе профили атак для различных стран. В основном использованы методы и инструменты графического и корреляционного анализа при допущении о стационарности исследуемых временных рядов.

В [70] основное внимание уделяется прогнозированию общего числа атак, а также атак из определенных географических регионов на «сеть-приманку» (honeynet). Авторы использовали несколько подходов, таких как экспоненциальное сглаживание, ARIMA, SARIMA, GARCH и Bootstrapping. При этом показано, что различные методы обеспечивают различную точность для разных временных рядов.

В [71] решена задача моделирования сезонных временных рядов количества атак на прикладное ПО с помощью гармонических составляющих.

В этих статьях была дана характеристика источников данных и методики их предобработки.

В настоящем разделе выполнен анализ динамики КА на информационную систему ведомственного вуза как элемент (объект и субъект) ЦИИ.

Для получения исходных данных была использована BI-платформа, позволяющая графически представить динамику временных рядов в разрезе различных видов из разных стран.

Периодичность поступления данных от источников позволяет сформировать временные ряды, получить многомерный временной ряд, сформировать панельные данные – то есть дополнительно к задачам визуального анализа решать задачи прогнозирования.

В качестве источников статистических данных инцидентов ИБ были использованы логи программно-аппаратного межсетевого экрана.

В зависимости от версий этот межсетевой экран имеет спектр модулей фильтрации трафика: контент-фильтр, контроль приложений, предотвращение вторжений, антивирус веб-трафика.

Рассматривались ряды количества атак по уровню угрозы столбиковые диаграммы условного количества кибератак, построенные с помощью ведомственной BI, приведены на рисунке 12.

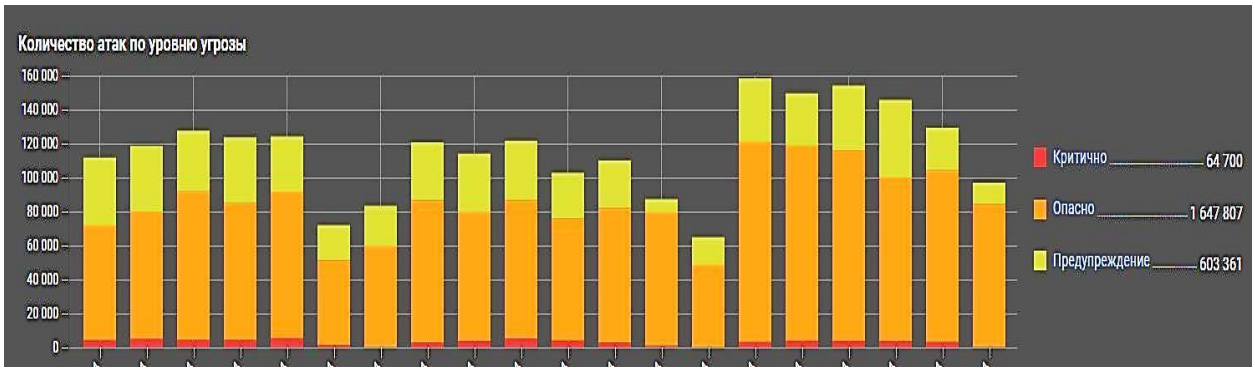


Рисунок 12 – Количество инцидентов ИБ (атак) по уровню угрозы

В ходе анализа рассматривались логи за 3 месяца межсетевого экрана. Полученные статистические данные позволили построить следующие интервальные временные ряды числа критических и опасных атак, числа предупреждений, а также – суммарного числа атак. В качестве временного шага был выбран один час, что позволило построить сравнительно длинные ряды и

сформулировать гипотезы о наличии сезонных составляющих, а также решить традиционные задачи разведывательного анализа: исследования стационарности временных рядов, построения их моделей и прогнозирования уровней временных рядов с их помощью.

Результаты апробации авторского подхода приведены в разделе 3 настоящей статьи.

Задача прогнозирования кибератак является задачей краткосрочного или текущего прогнозирования и, следовательно, должна быть включена в средства мониторинга и бизнес-аналитики, например, в BI-платформы. Отметим, что в существующих рейтингах BI-платформ, таких как квадрант Гартнера, в 2023 г. появился критерий оценки «Интеграция с data science».

За последние годы появилось много новых методов прогнозирования временных рядов, например, STL [78], BSTS [79], Prophet [80]; широкое применения нашли методы пространства состояний, байесовской статистики и др. Их использование позволит повысить качество прогнозирования.

Однако следует помнить, что нет «серебряной пули» – не метод, а качество исходных данных может обеспечить успех в прогнозировании. Заметим, что эта задача очень трудоемка (авторы статьи еще раз убедились в этом, формируя анализируемые временные ряды) и не может быть решена без разработки специальных средств парсинга данных.

Можно предположить, что использование «мягких вычислений» и нейронных сетей наряду с традиционными методами прогнозирования позволит получить более обоснованные результаты

2. Методический аппарат обоснования рациональных способов мониторинга и реагирования на возможные инциденты информационной безопасности на основе обобщенной эффективностной модели

Мониторинг и реагирование на возможные инциденты ИБ в ведомственной ЦИИ является комплексным процессом, требующим учета множества различных факторов – организационных, программных, технических и иных. Повышение же эффективности процесса (по крайне мере до рациональных показателей) является сложной задачей, требующей системного подхода к обоснованному выбору рациональных способов. Так, например, выбор одного набора методов мониторинга может оказаться неприменимым, из-за отсутствия необходимых для этого средств. Или же, выбор программных средств реагирования будет неудовлетворительным, из-за отсутствия должностных лиц по защите информации с необходимым уровнем компетенции.

Ситуация может оказаться еще более неоднозначной, если подходящими окажется несколько методов и средств, выбор рационального набора среди которых потребует учета их влияния на эффективность всего процесса мониторинга и реагирования.

Следовательно, требуется создание единого математического аппарата, позволяющего как отбирать комплексы методов и средств, работоспособные в рамках ЦИИ, но и обоснованно выбирать среди них наиболее рациональные вариации с позиции итоговой эффективности.

Поскольку отбор совокупности методов и средств для решения задач мониторинга и реагирования на возможные инциденты ИБ требует комбинирования множества различных связанных сущностей (с условиями и ограничениями), то требуется создание соответствующей модели, описание и формализация элементов которой дается далее.

Способы мониторинга и реагирования могут быть представлены, как совокупность методов и средств, используемых некоторым должностным лицом ЦИИ; при этом, первые два элемента не является жестко заданными и могут вариативно выбираться, а третий – определяется условиями их использования.

Изначально существует некоторый набор всех возможных методов, подходящих для мониторинга и реагирования на инциденты ИБ:

$$\begin{cases} m_i \in M \\ i \in [1 \dots N_M]' \end{cases} \quad (18)$$

где m_i – i-ый метод; M – множества всех возможных методов; N_M – число всех возможных методов.

Также, имеется аналогичный набор всех возможных средств (программах, аппаратных, иных):

$$\begin{cases} t_j \in T \\ j \in [1 \dots N_T]' \end{cases} \quad (19)$$

где t_j – j-ое средство; T – множества всех возможных средств; N_T – число всех возможных средств.

Поскольку для решения задач мониторинга и реагирования с помощью каждого метода могут использоваться лишь определенные средства (оборудование), то имеется их взаимное соотношение с позиции совместной применимости, которые можно записать с помощью соответствующей матрицы (MT):

$$\begin{cases} MT = \begin{pmatrix} mt_{1,1} & \cdots & mt_{N_M,1} \\ \vdots & mt_{i,j} & \vdots \\ mt_{1,N_T} & \cdots & mt_{N_M,N_T} \end{pmatrix}, \\ mt_{i,j} \in Bool \end{cases} \quad (20)$$

где $mt_{i,j}$ – применимость средства t_j в методе m_i; $Bool$ – одно из булевских значений (здесь и далее), соответствующее 1 в случае положительности утверждения и 0 в ином случае. При этом, один метод для своей работы выбирает лишь одно средство из их пула имеющихся, а одно средство может одновременно использоваться в различных методах.

Задачи (мониторинга и реагирования) можно записать как:

$$\begin{cases} p_k \in P \\ k \in [1 \dots N_K]' \end{cases} \quad (21)$$

где p_k – k-ая задача; P – множества всех необходимых для решения задач; N_K – число всех поставленных задач.

Тогда связь задач и методов их решения также можно записать с помощью соответствующей матрицы (PM):

$$\begin{cases} PM = \begin{pmatrix} pm_{1,1} & \cdots & pm_{N_p,1} \\ \vdots & pm_{k,i} & \vdots \\ pm_{1,N_m} & \cdots & pm_{N_p,N_m} \end{pmatrix}, \\ pm_{k,i} \in Bool \end{cases} \quad (22)$$

где $pm_{k,i}$ – применимость метода m_i для решения задачи p_k.

Итоговая постановка задачи отбора методов и средств мониторинга и реагирования на инциденты ИБ может быть поставлена в следующем виде:

$$\begin{cases} D = \{d_x\} \\ d_x \in [1 \dots N_D] \\ d_x = \langle m_i', t_j' \rangle \Longrightarrow \left(\forall m_i' \in M, \exists t_j' \in T : mt_{i,j} = 1\right) \wedge \left(\forall p_k \in P, \exists m_i : pm_{k,i} = 1\right) \end{cases} \quad (23)$$

где D – множество всех решений; d_x – x-ое решение (как кортеж из метода и используемого им средств); N_D – число всех решении; m_i' и t_j' – отобранные методы и средства в интересах решения всех необходимых задач.

Формальная запись постановки задачи может быть интерпретирована следующим образом – «Результатом отбора является множество совокупность метода и средства из множества доступных, таких, что для каждого метода существует необходимое ему средств, а для каждой задачи мониторинга и реагирования существует решающий ее метод».

Решение задачи отбора в рамках ЦИИ требует учета ее специфики, связанной наличием строго определенного пула должностных лиц, осуществляющих процесс мониторинга и реагирования на инциденты:

$$\begin{cases} o_l \in O \\ l \in [1 \dots N_O]' \end{cases} \quad (24)$$

где o_l – l-ой должностное лицо, осуществляющее процесс мониторинга и реагирования; O – множество всех должностных лиц; N_O – количество всех должностных лиц.

В области ИБ существует некоторый набор компетенций, необходимый для решения актуальных задач области:

$$\begin{cases} c_s \in C \\ s \in [1 \dots N_C]' \end{cases} \quad (25)$$

где c_s – s-ая компетенция в ИБ; C – множество всех компетенций; N_C – количество всех компетенций.

Соответственно, с каждым должностным лицом связан набор компетенций, необходимых для решения определенных частных задач мониторинга и реагирования на инциденты, что можно записать в форме матрице должностных компетенций (CO):

$$\begin{cases} CO = \begin{pmatrix} co_{1,1} & \cdots & co_{N_C,1} \\ \vdots & co_{s,l} & \vdots \\ co_{1,N_O} & \cdots & co_{N_C,N_O} \end{pmatrix}, \\ co_{s,i} \in Bool \end{cases} \quad (26)$$

где $co_{s,l}$ – необходимость наличия компетенции c_s у должностного лица o_l.

Очевидно, что методы для своего применения требуют у использующих их лиц заданного набора компетенций, что может быть записана с помощью соответствующей матрицы (CM):

$$\begin{cases} CM = \begin{pmatrix} cm_{1,1} & \cdots & cm_{N_C,1} \\ \vdots & cm_{s,i} & \vdots \\ cm_{1,N_M} & \cdots & cm_{N_C,N_M} \end{pmatrix}, \\ cm_{s,i} \in Bool \end{cases} \quad (27)$$

где $cm_{s,i}$ – необходимость наличия компетенции c_s у должностного лица, применяющего метод m_i.

Аналогичным образом, соответствие требуемых компетенций средствам имеет следующую матричную запись (CT):

$$\begin{cases} CT = \begin{pmatrix} ct_{1,1} & \cdots & ct_{N_C,1} \\ \vdots & ct_{s,j} & \vdots \\ ct_{1,N_T} & \cdots & ct_{N_C,N_T} \end{pmatrix}, \\ ct_{s,j} \in Bool \end{cases} \quad (28)$$

где $ct_{s,j}$ – необходимость наличия c_s компетенции у лица, применяющего t_j-ое средство.

Тогда ограничение возможных решений задачи отбора заключается в том, что для каждого метода и его средства должно существовать должностное лицо, которое его способно (в рамках своих компетенций) применять:

$$\forall d_x = \langle m'_i, t'_j \rangle, \exists o_l \in O: \left(\forall s, cm_{s,i} = 1 : co_{s,l} = 1\right) \wedge \left(\forall s, ct_{s,j} = 1 : co_{s,l} = 1\right). \quad (29)$$

Формальная запись ограничения к решениям задачи может быть интерпретирована следующим образом – «Для каждого отобранного решения должно существовать должностное лицо, компетенции которого полностью покрывают компетенции, требуемые для применения метода и его средства, используемые для этого решения».

Следуя аналитической записи модели, задаваемая по ней задача относится к разряду комбинаторных оптимизаций, у которой есть различные способы решения, такие, как следующие.

Во-первых, привлечение группы экспертов позволит в ряде случаев производить такого рода комбинаторные оптимизации, что, впрочем, по мере усложнения задачи (за счет увеличения количества ее элементов и сложности их связей) приведет к логичному уменьшению результативности и оперативности решения.

Во-вторых, полный перебор является гарантированно результативным средством осуществления комбинаторной оптимизации, который, однако, из своей как правило крайне низкой оперативности не может применяться на практике для решения NP-задач. Его альтернативами можно считать применение динамического и линейного программирования.

В-третьих, существуют эвристические алгоритмы, которые хотя и не гарантированно позволяют получить глобальный экстремум, однако получаемые с помощью них решения могут считаться удовлетворительными. К таким алгоритмам можно отнести основанные на эволюциях полулицый, роевом интеллекте, различных механизмах поиска и др.

Исходя из того, что множество решений, полученных при отборе методов и средств мониторинга и реагирования на инциденты ИБ (с учетом имеющихся и требуемым компетенций должностных лиц) может быть огромным, то, очевидно, что не все из решений будут одинаково оптимальными с позиции эффективности всей такой подсистемы ИБ.

Так, ручной метод будет иметь высокую результативность при средней человеческой ресурсоэкономности и крайне низкой оперативности; в противовес этому, полностью автоматические средства потребуют лишь вычислительные мощности, будут иметь высокую оперативность, но обнаружение сложных многошаговых инцидентов (по крайне мере без экспертной подстройки и/или проверки) окажется не всегда результативным.

Таким образом, задача поиска множества пар методов и средств с исключительно однокритериальной комбинаторной должна быть расширены до многокритериальной путем учета показателей эффективности – результативности, оперативности и ресурсоэкономности; при этом, последний показатель логично декомпозировать на человеческую и машинную составлявшие.

Следовательно, необходима соответствующая эффективностная модель, оценивающая вклад каждого метода (и его средства) в общую эффективность процесса мониторинга и реагирования на инциденты ИБ; затем, потребуется свертка полученных эффективностей решений задачи комбинирования в единую, максимизация которой и является дополнительным критерием:

$$\begin{cases} E = Eff(\{EM_x\} \cup \{ET_x\}) \\ E \to Max \end{cases}, \quad (30)$$

где EM_x и ET_x – эффективность методов и средств из x-го решения; E – эффективность всего процесса мониторинга и реагирования на инциденты ИБ; $Eff(\dots)$ – операция получения единой интегральной эффективности по множеству частных для методов и средств; «$\to Max$» – критерий максимизации.

При этом, поскольку на разных этапах процесса возникают и должны решаться качественно разные задачи, то так или иначе применение методов и средств на одних этапах будет напрямую или косвенно влиять и на эффективность других этапов; например, точность обнаружения уязвимостей в программном обеспечении обновлений оборудования существенно повлияет и на эффективность их устранения [81–83].

Таким образом, весь процесс мониторинга и реагирования можно поделить на 4 крупных (общих) этапа – 1) сбор и подготовка данных, 2) обнаружение инцидента, 3) создание сценария противодействия и 4) непосредственное реагирования. Соответственно, каждый метод и средство – т.е. некоторая альтернатива в комбинаторной оптимизации, помимо влияния на эффективность своего этапа (т.е. того, для решения задач которого они предназначены) может иметь косвенное влияние и на другие этапы; что необходимо учесть в эффективностной модели.

Приведенную модель будем называть общей, т.к. без дополнительного учета предметной области (в данном случае, особенностей процесса мониторинга и реагирования на инциденты ИБ, а также, специфики ЦИИ) она может применяться для большого класса достаточно абстрактно описанных задач.

Схема обобщенной эффективностной модели в виде шестиугольника между вспомогательными показателями и параметрами альтернатив соответствует связям «каждый-с-каждым» представлена на рисунке 13.

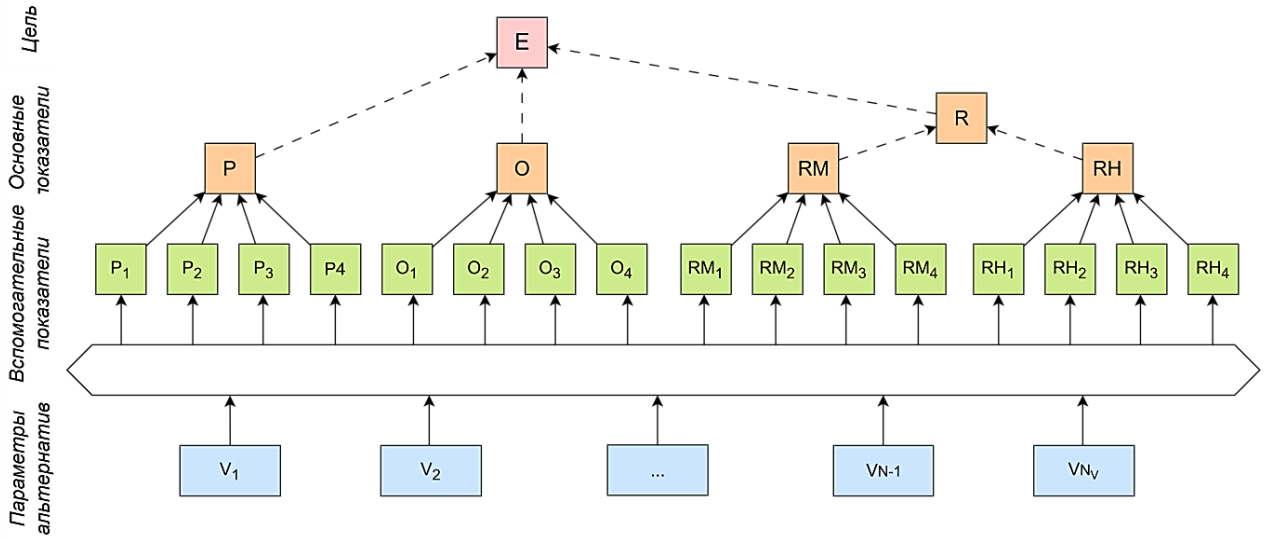


Рисунок 13 – Иерархическая общая модель эффективности процесса мониторинга и реагирования на инциденты ИБ

Следуя модели, она состоит из 4-х уровней со следующим назначением.

Уровень 1. Цель (E) – итоговая эффективность, вычисление которой для альтернативы необходимо осуществить.

Уровень 2. Основные показатели – декомпозиция эффективности на общую результативность (P), оперативность (O) и ресурсоэкономность (R), которая в свою очередь подразделяется на человеческую (RM) и машинную (RH):

$$R = \langle RM, RH \rangle. \tag{31}$$

Уровень 3. Вспомогательные показатели – учет при вычислении показателей эффективности влияния альтернатив на каждый из общих этапов процесса мониторинга и реагирования на инциденты:

$$\begin{cases} P = Funct^{P}(P_1, P_2, P_3, P_4) \\ O = Funct^{O}(O_1, O, O_3, O_4 \\ RM = Funct^{RM}(RM_1, RM_2, RM_3, RM_4)' \\ RH = Funct^{RH}(RH_1, RH_2, RH_3, RH_4) \end{cases} \tag{32}$$

где P_i, O_i, RM_i и RH_i – результативность, оперативность, машинная и человеческая ресурсоэкономность i-го этапа; а $Funct^P(\dots)$, $Funct^O(\dots)$, $Funct^{RM}(\dots)$, $Funct^{RH}(\dots)$ – операции получения на основании их соответствующих основных показателей.

Уровень 4. Параметры альтернатив – учет конкретных показателей альтернатив, влияющих на все этапы процесс мониторинга и реагирования; по данным параметрам происходит вычисление вспомогательных показателей эффективности:

$$S_i = Funct_i^S(V_1 \dots V_{N_V}), \tag{33}$$

где S_i – следующие вспомогательные показатели: $S_1 \dots S_4$ – для $P_1 \dots P_4$, $S_5 \dots S_8$ – для $O_1 \dots O_4$, $S_9 \dots S_{12}$ – для $RM_1 \dots RM_4$, $S_{13} \dots S_{16}$ – для $RH_1 \dots RH_4$; V_j – j-й параметр альтернативы; N_V – число параметров альтернатив; $Funct_i^S(\dots)$ – операция получения i-го вспомогательного показателя по всему множеству параметров альтернатив, которая производит вычисления интегрального показателя с весами, заданными следующей матрицей (SV):

$$SV = \begin{pmatrix} sv_{1,1} & \cdots & sv_{16,1} \\ \vdots & sv_{i,j} & \vdots \\ sv_{1,N_V} & \cdots & sv_{16,N_V} \end{pmatrix}, \tag{34}$$

где $sv_{i,j}$ – вес влияния j-го параметра альтернативы на i-ый вспомогательный показатель.

Таким образом, элементы каждого вышестоящего уровня вычисляются по элементам нижестоящего, например, через интегральный показатель с весами. Параметры альтернатив же являются элементами самого низкого уровня модели и задаются на основании оценки показателей каждого метода или средства.

Для обоснования работоспособности общей эффективностной модели был проведен практический эксперимент по оценке процесса нейтрализации уязвимостей программы с позиции используемых в нем подходов-альтернатив для реверс-инжиниринга машинного кода. В интересах этого была построена более простая и «понятная» частная эффективностная модель, уточняющая описанную общую за счет учета предметной области – нейтрализации уязвимостей программ.

Поскольку программа в процессе своего создания «развивается» из представления концептуальной модели или архитектуры в представление машинного кода, то такой процесс можно назвать ее эволюцией; обратный соответственно – деэволюцией.

Схожесть процесса с мониторингом и реагированием на инциденты ИБ заключается в следующем:

- процесс нейтрализации представляет собой обнаружение и устранение уязвимостей, что соответствует мониторингу и реагированию на инциденты ИБ;
- наличие уязвимости в программе, уже попавшей в защищаемую систему, является существенной предпосылкой к реальным инцидентам ИБ, поскольку ее эксплуатация подготовленным злоумышленником с высокой вероятностью приведет к нарушениям конфиденциальности, целостности и доступности информации;
- процесс нейтрализации состоит из 4-х основных этапов, которые по своей логике подобны аналогичным для процесса мониторинга и реагирования;
- для выбора подходящего подхода к реверс-инжинирингу машинного кода требуется оценка его вклада в процесс нейтрализации уязвимостей, что аналогично оценке эффективности процесса мониторинга и реагирования для выбора наиболее эффективной комбинации методов и средств.

Частная модель хотя и подобна применяемым в методе анализа иерархий (при выборе наилучшей альтернативы), но отличается от них назначением, а также формализацией и интерпретацией элементов (например, отсутствием в ряде случаев нормализованных приоритетов) [84].

Системный анализ процесса нейтрализации уязвимостей позволил ее выделить ее основные элементы, их особенности, влияющие факторы и связи. Здесь и далее ограничимся нейтрализацией уязвимостей в классических программах, построенных с применением императивной и процедурной парадигм программирования (в отличие, например, от программ с декларативной парадигмой, таких, как HTML). При этом, конечным представлением программы будем считать машинный код из инструкции, непосредственно выполняемые на физическом процессоре (а, например, не байт-код для виртуальных машин или скрипты для интерпретаторов).

Весь процесс нейтрализации уязвимостей может быть разделен на 4 следующих этапа.

Этап 1. Получение требуемого представления программы из конечного. Вначале требуется получение представления программы, в котором будет производится поиск уязвимостей. Изначально, как правило, имеется конечное представление программы, выполняемое на аппаратной составляющей системе, такое, как машинный код. Однако, поиск уязвимостей по нему, как правило, имеет низкую эффективность из-за «расплывания» средне- и высокоуровневых уязвимостей, слабой интерпретируемости сложных конструкций экспертом, огромного размера и ряда других причин. Поэтому целесообразно преобразование машинного кода в более высокоуровневые представления, такие, как исходный код, алгоритмы, архитектуру и т.д. Получение каждого предыдущего представления из текущего может быть логично названо его деэволюцией (а в частном случае получения исходного кода по машинному – декомпиляцией), весь же процесс получения предыдущих представлений – реверс-инжинирингом.

Этап 2. Обнаружение уязвимости в полученном представлении программы. Получив требуемое представление программы, в нем производится непосредственный поиск уязвимостей, заданных в тех конструкциях, которые являются базовыми для представления (в том числе, применяя достаточно сложные техники, например, анализ помеченных данных графов [85], использование сверточных нейронных сетей [86], языковой модели Reformer [87] и др.). Так, уязвимость в машинном коде будет построена на инструкциях процессора, в исходном коде – на синтаксических элементах языка программирования, в алгоритмах – на логических примитивах с детализацией их работы и т.д.

Этап 3. Создание алгоритма устранения уязвимости в полученном представлении программы. После обнаружения уязвимости необходимо создание стратегии, а затем и точного алгоритма ее устранения. Например, ошибка в вычислениях может быть устранена корректировкой последних, программная закладка – ее полным устранением, отсутствие необходимых проверок входных данных – их добавлением и т.п. При этом очевидно, что алгоритм оперирует конструкциями полученного представления программы, а не конечного.

Этап 4. Исправление уязвимости в конечном представлении программы. И, наконец, полученный алгоритм устранения уязвимости должен быть непосредственно применен для конечного представления программы – для ее исправления. В случае работы на Этапах 2 и 3 с машинным кодом, исправление уязвимости будет производиться в нем же и не потребует каких-либо дополнительных сложных действий. Однако, если для обнаружения уязвимостей был вначале восстановлен ее исходный код (корректнее говорить «псевдо-», поскольку истинный код практически никогда получить нельзя), то потребуется либо внесение исправлений в него с пересборкой машинного кода, либо экспертный или частично автоматизированный «перенос» исправлений в машинный код, т.е. «патчинг».

Выделим основные подходы-альтернативы, используемые для деэволюции представлений программы при нейтрализации ее уязвимостей в порядке их эволюционного возникновения; данные подходы считаются альтернативами в частной модели и именно среди них, исходя из эффективности каждого, необходимо делать обоснованный выбор.

Альтернатива 1. Ручной подход. Исторически первым подходом к деэволюции представлений может считаться ручной, в процессе которого эксперт анализировал конструкции текущего представления, стараясь создать аналогичные им конструкции предыдущего. Например, машинные инструкции сравнения регистра с числом, за которым в случае выполнения условия

следовал переход на заданный адрес, могли быть преобразованы им в конструкцию условного перехода на языке программирования С.

Альтернатива 2. Алгоритмический подход. Появление вспомогательных программных утилит и накопление опыта реверс-инжиниринга программ позволил сформировать алгоритмический подход, производящий преобразование представлений (от текущего к предыдущему) согласно правилам, созданным и запрограммированным экспертами [88]; наиболее ярким представителем таких средств (утилит) являются декомпиляторы (например, плагин Hex-Rays из состава IDA Pro, встроенные функции в Ghidra и др.) [89].

Альтернатива 3. Интеллектуальный подход. Повсеместное успешное внедрение искусственного интеллекта в огромное количество областей ИБ позволило решать отдельные задачи реверс-инжиниринга, такие, как распознание шаблонов обфускации кода, восстановление структур и свойств данных [90], генерация псевдокода логики работы, документирование и пр. Например, по математическим операциями посредством инструкций машинного кода может быть с некоторой вероятностью предсказан тип используемых переменных в исходном коде.

Альтернатива 4. Полный перебор. Наращивание аппаратных мощностей современных систем позволяет, хотя и только теоретически, рассмотреть полный перебор конструкций предыдущего представления для получения экземпляра, в точности преобразуемого в текущее, как еще одну из альтернатив деэволюции представлений. Например, перебор токенов исходного кода в соответствии с формальным синтаксисом и семантическими правилами позволит скомпилировать множество экземпляров машинного кода, что в случае полного совпадения с заданным будет говорить о деэволюции последнего. Естественно, в общем случае время такого перебора будет критически высоким

Альтернатива 5. Генетический подход. Данный развиваемый подход [91, 92], предназначенный для деэволюции представлений, основан на сведении данной задачи к оптимизационной, для решения которой обосновано применение генетических алгоритмов – т.е. путем итерационного подбора конструкций предыдущего представления с целью приближения его (после преобразования, например, компиляции) к текущему. Например, деэволюция машинного кода для сложения двух переменных может быть произведена путем «умного» подбора переменных и бинарной математической операции исходного кода путем их компиляции в соответствующий машинный код по мере его приближения к заданному конечному. И хотя такой подход формально относится к группе интеллектуальных, тем не менее, из-за своего качественного отличия, он может быть выделен в отдельную группу.

Как было указано ранее, общая модель эффективности является стратифицированной и состоит из 4 следующих уровней, элементы каждого вышестоящего из которых вычисляются на основании элементов нижестоящего; ее графическая схема будет приведена ниже после детализации всех элементов; введем ее уточнения в соответствии с предметной областью эксперимента для получения частной.

Первый уровень частной модели соответствует общей и не требуют уточнения.

На втором уровне модели под результативностью R понимается доля нейтрализованных уязвимостей выбранным подходом, которая увеличивается за счет каждого этапа:

$$\begin{cases} P = \prod_i P_i = \dfrac{V_4}{V_0} \\ P_i = \dfrac{V_i}{V_{i-1}} \\ i \in \{1,2,3,4\} \end{cases}, \quad (35)$$

где P_i – результативность каждого i-го этапа согласно его назначению, т.е. доля уязвимостей, подготовленных для нейтрализации (отраженных в представлении, обнаруженных, имеющих алгоритмы устранения и исправленных); V_i – количество уязвимостей на выходе i-го этапа; V_0 – изначальное количество уязвимостей в программе.

Под оперативностью O понимается скорость нейтрализации уязвимостей (их количество в единицу времени) выбранным подходом, которая растет за счет сокращения времени выполнения каждого этапа:

$$O = \frac{V_0}{\sum_i T_i}, \tag{36}$$

где T_i – время выполнения i-го этапа (*от англ.* Time).

Ресурсоэкономность подхода была декомпозирована на две противоположные (категориальные по своей сути: человек vs машина) составляющие – машинную (RM) и человеческую ресурсоэкономность (RH). Таким образом, общее число элементов на данном уровне частной модели равно 5.

Машинная ресурсоэкономность отражает «слабость» используемого аппаратного обеспечения (например, малое количество потребных процессоров и их частота, малый объем потребляемой памяти и т.п.). Аналогичным образом, «человеческая» ресурсоэкономность отражает низкие требования к уровню квалификации эксперта (поскольку исходя из интеллектуальности задачи нейтрализации простое увеличение количества экспертов при их слабом уровне не принесет существенного эффекта).

Третий уровень частной модели, элементы которого необходимы для вычисления элементов второго, состоит из вспомогательных показателей эффективности. При этом показатели разбиты на группы, аналитически связанные с каждым из основных показателей (кроме родительского для декомпозированных, т.е. R). В каждой группе присутствует по 4 показателя, отражающих особенности соответствующих этапов процесса нейтрализации уязвимостей. Таким образом, общее число элементов на данном уровне $4 \times 4 = 16$.

Первая группа состоит из показателей результативности каждого из этапов; при этом единый показатель результативности не может быть вычислен аналитически и представляет собой кортеж:

$$P = \langle P_1, P_2, P_3, P_4 \rangle. \tag{37}$$

Вторая группа состоит из показателей оперативности каждого из этапов, которые могут быть выражены через сокращение времени относительно максимально затрачиваемого за счет применения подхода-альтернативы:

$$\begin{cases} T_i = T_i^{Max}(1 - Q_i^T \times a_i) \\ Q_i^O \in [0 \dots 1] \\ K_i^O \in [0 \dots 1] \end{cases}, \tag{38}$$

где T_i^{Max} – условно максимальное время i-го этапа (*от англ.* Maximum); Q_i^T – доля времени выполнения i-го этапа, в рамках которого может происходить его ускорение за счет подхода деэволюции (*от англ.* Quote); a_i – коэффициент (или степень) сокращения времени выполнения i-го этапа относительно возможных рамок (*от англ.* Acceleration). Таким образом, для наиболее неэффективного с позиции оперативности подхода ($a_i = 0$) время этапа не сократится – $T_i = T_i^{Max}$; а для наиболее оперативно-эффективного похода ($a_i = 1$) время этапа будет минимально достижимым – $T_i = T_i^{Max}(1 - Q_i^T)$. Также, чем шире рамки возможного ускорения, тем этап может занять меньшее время – при максимальных $Q_i^T = 1$ и $a_i = 1$ оно будет теоретически снижено до 0.

Для получения оценочных значений показателя оперативности, подходящего для непосредственного сравнения альтернатив, можно сделать следующие разумные упрощения.

Во-первых, поскольку деэволюция является одним из ключевых звеньев во всем процессе нейтрализации уязвимостей и имеет качественное влияние на оперативность всех его этапов, то доля времени их выполнения для возможного ускорения, может считаться одинаковой:

$$\forall i: Q_i^T = Q^T, \tag{39}$$

где Q^T – единая доля времени выполнения этапов для ускорения, за счет подхода-альтернативы.

И, во-вторых, несмотря на различия в назначениях, принципах и вариациях решений для каждого из этапов, существуют общие соотношения времен их выполнения, что позволяет перейти от максимально затрачиваемого времени на каждом этапе к времени всего процесса нейтрализации (T^{Max}) путем использования весов:

$$\begin{cases} T^{Max} = \sum_i T_i^{Max} \\ T_i^{Max} = T^{Max} \times W_i^T \\ \sum_i W_i^T = 1 \\ W^T = |W_1^T, W_2^T, W_3^T, W_4^T| \end{cases}, \tag{40}$$

где W_i^T – доля или вес времени i-го этапа относительно общего времени нейтрализации (*он англ.* Weight); W^T – вектор таких весов для 4-х этапов.

Исходя из выше сделанных аналитических записей, оперативность i-го этапа приобретает следующий вид:

$$O = \frac{V_0}{T^{Max} \times (1 - Q^T \times \sum_i W_i^T \times a_i)}, \tag{41}$$

в котором элементы суммы в знаменателе вычислимы для каждого этапа – веса W_i^T могут быть получены экспертно, а коэффициенты a_i определяются через элементы нижнего уровня частной модели.

Таким образом, сравнение оперативности подходов может быть осуществлено исходя из суммарной доли сокращения времени выполнения всего процесса нейтрализации уязвимостей:

$$A = \sum_i A_i = \sum_i W_i^T \times a_i, \tag{42}$$

где A_i – взвешенный коэффициент сокращения времени выполнения i-го этапа.

Следовательно, альтернатива с более высоким A приводит к сокращению времени выполнения процесса и, следовательно, к повышению его оперативности. Таким образом, данный 3-й уровень частной модели в части второй группы содержит 4 дополнительных элемента.

Третья группа состоит из показателей машинной ресурсоэкономности каждого из этапов; при этом единый показатель может быть вычислен усредненно по всем этапам, поскольку подход-альтернатива предъявляет собственные требования к аппаратным средствам каждого из них:

$$\begin{cases} Funct^{RM} \coloneqq Avg \\ RM = Avg(RM_1, RM_2, RM_3, RM_4) \end{cases}, \tag{43}$$

где $Avg(\dots)$ – оператор вычисления среднего значения из набора.

Четвертая группа состоит из показателей «человеческой» ресурсоэкономности каждого из этапов процесса, а единый показатель может быть вычислен аналогичным третьей группе образом:

$$\begin{cases} Funct^{RH} \coloneqq Avg \\ RH = Avg(RH_1, RH_2, RH_3, RH_4) \end{cases}. \tag{44}$$

Четвертый уровень частной модели, элементы которого необходимы для вычисления элементов третьего, состоит из параметров деэволюции представлений с помощью подходов-

альтернатив, отображающих их особенности с позиции эффективности нейтрализации уязвимостей. При этом параметры разбиты на группы, логически (но не аналитически) связанные с каждым из основных показателей – P, O, RM и RH – через соответствующие группы вспомогательных показателей. В каждой группе присутствует по 3 параметра, полученных экспертным методом и наиболее существенно отражающих специфику альтернатив. Таким образом, общее количество элементов на данном уровне $3 \times 4 = 12$. Название каждого параметра имеет форму «VX_Y», где V – аббревиатура от слова «параметр» (*от англ.* Variable), X – аббревиатура связанного основного показателя, Y – аббревиатура названия параметра.

К первой группе параметров относятся те, которые напрямую связывают процесс деэволюции представлений с результативностью нейтрализации уязвимостей, а именно:

1) VP_{PR} – релевантность полученного представления к конечному, как степень возможности получения всех представлений программы, в которых могли возникнуть уязвимости;

2) VP_{FI} – функциональная тождественность полученного представления к конечному, как степень соответствия функциональности восстановленных представлений программы такой же в конечном представлении;

3) VP_{NI} – нотационная инвариантность полученного представления к конечному, как степень независимости преобразования представлений к их нотациям.

Ко второй группе – с оперативностью:

1) VO_{AC} – степень непрерывности выполнения действий в процессе деэволюции;

2) VO_{CS} – быстрота выполнения процесса деэволюции представлений;

3) VO_{NA} – степень отсутствия необходимости в корректировке процесса деэволюции.

К третьей группе – с машинной ресурсоэкономностью:

1) VRM_{CS} – степень простоты вычислений, выполняемых средством в процессе деэволюции;

2) VRM_{NI} – степень без-итеративности алгоритмов процесса деэволюции;

3) VRM_{DH} – степень однородности (или гомогенности) обрабатываемых в процессе деэволюции данных.

К четвертой группе – с «человеческой» ресурсоэкономностью:

1) VRH_{ID} – степень отсутствия высоких требований к квалификации эксперта, подготавливающего входные данные для деэволюции представлений;

2) VRH_{PE} – степень отсутствия высоких требований к квалификации эксперта, проводящего процесс деэволюции;

3) VRH_{OD} – степень отсутствия высоких требований к квалификации эксперта, подготавливающего выходные данные после деэволюции представлений.

Параметры первой и второй групп были получены на основе экспертного мнения специалистов по реверс-инжинирингу программ, третьей и четвертой – специалистов по программному инжинирингу.

Несмотря на очевидную связь каждой из групп параметров со своими основными показателями эффективности (через группы вспомогательных), тем не менее существует их косвенное влияние и на другие группы. Так, например, хотя от релевантности полученного представления (VP_{PR}) зависит результативность обнаружения уязвимостей (P_2) за счет отображения в каждом представлении уязвимости своего структурного уровня, тем не менее, оно незначительно снижает ресурсопотребление соответствующими средствами автоматического поиска уязвимостей на 2-ом этапе процесса нейтрализации уязвимости (RM_2) и еще более понижает требования к уровню эксперта на нем; и т.д.

Связь параметров альтернатив со вспомогательными показателя имеет форму «все со всеми» (т.е. каждый параметр влияет на все показатели) и может быть отображена с помощью взвешенной суммы, где каждый показатель вычисляется как сумма параметров с некоторыми заданными коэффициентами:

$$\begin{cases} K_n = N \times \sum_m W_{n,m}^V \times V_m \\ n \in \{1 \dots 16\} \\ m \in \{1 \dots 12\} \\ W_{n,m}^V \in [0,1] \end{cases}, \quad (45)$$

где K_n – вычисленное значение n-го вспомогательного показателя, т.е. $K_1 \dots K_4$ соответствуют $P_1 \dots P_4$, $K_5 \dots K_8$ соответствуют $A_1 \dots A_4$ (которые потом будут пересчитаны в $O_1 \dots O_4$), $K_9 \dots K_{12}$ соответствуют $RM_1 \dots RM_4$ и $K_{13} \dots K_{16}$ соответствуют $RH_1 \dots RH_4$; V_m – заданное значение m-го параметра, соответствующему одному из указанных ранее в каждой группе ($V_1 = VP_{RP} \dots, V_4 = VO_{AC} \dots, V_7 = VRM_{CS} \dots, V_{10} = VRH_{ID} \dots$); $W_{n,m}^V$ – вес участия каждого из m-ых параметров в увеличении n-го вспомогательного показателя; N – нормировочный коэффициент, который будет рассмотрен далее.

Для удобства составления экспертных оценок параметры могут задаваться на единичной балльной шкале – $V_m \in [0,1]$ с интерпретацией значений, как степень наличия соответствующих свойств у альтернатив: от 0 (отсутствие) до 1,0 (полное присутствие) с шагом 0,25.

Так, например, значение $VP_{FI} = 1$ означает, что альтернатива обеспечивает полную функциональную тождественность получаемых представлений к конечному, а $VRM_{NI} = 0$ – что альтернатива при деэволюции выполняет огромное количество циклических действий.

Тогда нормировка значений вспомогательных показателей к 1 осуществляется за счет нормировочного коэффициента $N = 1/12$, поскольку в этом случае при максимальном значении весов $W_{n,m}^V$ и параметров V_m для всех m значение каждого показателя будет $K_n = 1$.

Графическая схема частной модели в виде иерархической связи ее элементов в которой для упрощения визуализации множества связей «все со всеми» между параметрами альтернатив и вспомогательными показателями использован объект в виде горизонтально вытянутого шестиугольника, представлена на рисунке 14.

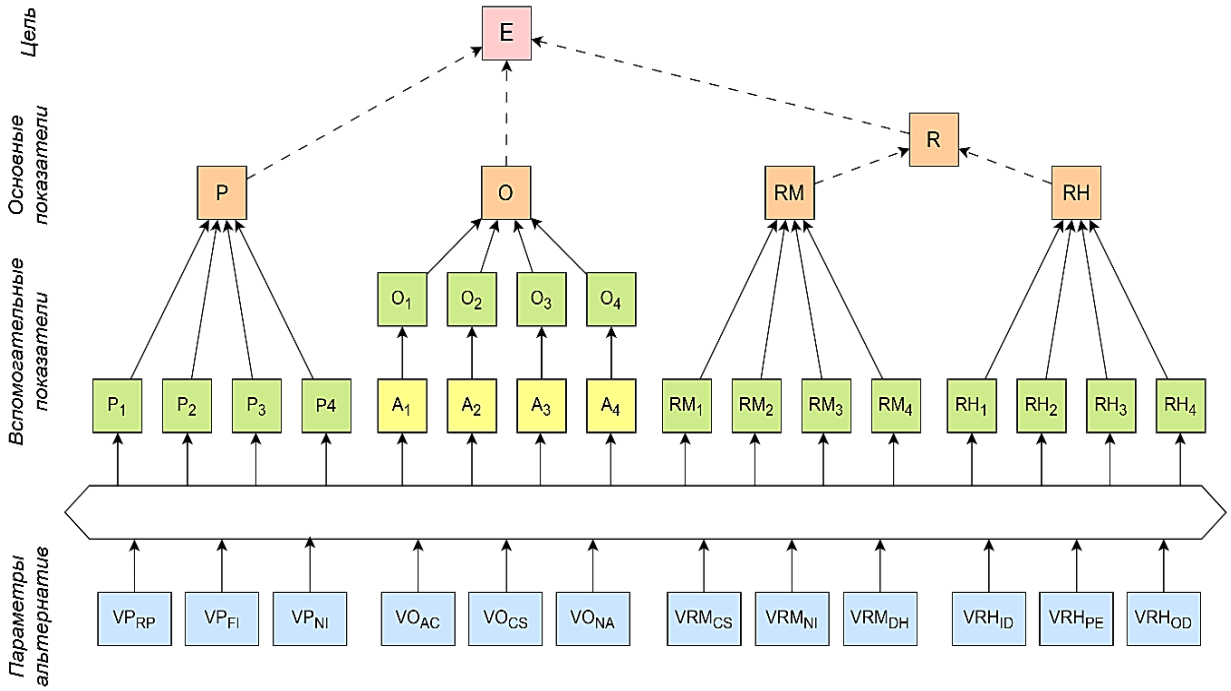


Рисунок 14 – Иерархическая частная модель эффективности нейтрализации уязвимостей в программе

Согласно схеме частной модели вычисление всех параметров альтернатив позволит произвести свертку и получить вспомогательные показатели, аналогичная свертка которых получит каждый из основных показателей, обеспечив тем самым определение итоговой эффективности процесса нейтрализации, а также качественное сравнение альтернатив и выбор наилучшей из них.

На основании экспертного мнения специалистов по ИБ, обладающих большим практическим опытом проведения реверс-инжиниринга программ, поиска в них уязвимостей и их исправлением, были получены необходимые оценки всех параметров и промежуточных весов частной модели.

Соответствующие значения приведены в таблице 5.

Таблица 5 – Экспертные значения весов частной модели ($W^V_{n,m}$)

Вспомогательные показатели (для индекса n)	Параметры (для индекса m)											
	VP_{PR}	VP_{FI}	VP_{NI}	VO_{AC}	VO_{CS}	VO_{NA}	VRM_{CS}	VRM_{NI}	VRM_{DH}	VRH_{ID}	VRH_{PE}	VRH_{OD}
P_1			0.25									
P_2	1	0.25	1									
P_3	0.25		0.5									
P_4		1	1									
A_1				0.5	1	0.75						
A_2	0.25	0.25	0.25			0.25						
A_3	0.25											
A_4		0.25	0.25									
RM							1	0.5	0.25			
RM_2	0.25		0.25									
RM_3												
RM_4		0.25	0.25									
RH_1			0.25	0.25		0.25				0.25	0.75	0.5
RH_2	0.5		0.25									0.25
RH_3	0.25		0.25									
RH_4		0.5	0.25									

В таблице используются следующие обозначения: сплошными линиями выделены блоки, делящие параметры и показатели на группы; пустая клетка соответствует значению «0». Доля времени каждого из этапов относительно всего времени нейтрализации уязвимостей была определена, как $W^T = |0{,}3;\ \ 0{,}4;\ 0{,}1;\ \ 0{,}2|$.

Выделенные ранее группы параметров альтернатив хотя и имеют основное влияние на соответствующие группы вспомогательных показателей и связанные с ними основные показатели, но и косвенно способны повышать другие показатели (см. верхний горизонтальный блок в Таблице 5).

Проведенная экспертная оценка параметров всех альтернатив, где используются следующие обозначения: «Альт.» – *сокр. от* Альтернатива; сплошными линиями выделены блоки, делящие параметры на группы; пустая клетка соответствует значению «0», представлена в таблице 6.

Таблица 6 – Значения параметров альтернатив

		Параметры (V_m)											
		VP_{PR}	VP_{FI}	VP_{NI}	VO_{AC}	VO_{CS}	VO_{NA}	VRM_{CS}	VRM_{NI}	VRM_{DH}	VRH_{ID}	VRH_{PE}	VRH_{OD}
Альтернативы	Альт. 1	0.5	0.25				0.25	0.75	0.25	0.25	0.25		0.25
	Альт. 2	0.25	0.5	0.25	0.75	1	0.75	0.25	1	0.25	0.75	0.75	0.75
	Альт. 3	0.5	0.25	0.25	0.75	0.5	0.25	0.25	0.75	0.25	0.5	0.5	0.5
	Альт. 4	0.75	1	0.75	1		0.75	0.5			1	1	1
	Альт. 5	1	1	1	1	0.25	0.5	0.5	0.5	0.5	1	1	1

Используя приведенные выше формулы, были получены значения основных показателей эффективности, представленные в таблице 7; для наглядности, наибольшие значения по каждому показателю отмечены зеленым, а наименьшие – синим.

Таблица 7 – Значения вспомогательных показателей эффективности для альтернатив

		Вспомогательные показатели															
		P_1	P_2	P_3	P_4	A_1	A_2	A_3	A_4	RM_1	RM_2	RM_3	RM_4	RH_1	RH_2	RH_3	RH_4
Альтернативы	Альт. 1	0.0000	0.0352	0.0078	0.0156	0.0035	0.0063	0.0008	0.0008	0.0586	0.0078	0.0000	0.0039	0.0156	0.0195	0.0078	0.0078
	Альт. 2	0.0039	0.0391	0.0117	0.0469	0.0363	0.0109	0.0004	0.0023	0.0508	0.0078	0.0000	0.0117	0.0977	0.0234	0.0078	0.0195
	Альт. 3	0.0039	0.0508	0.0156	0.0313	0.0199	0.0078	0.0008	0.0016	0.0430	0.0117	0.0000	0.0078	0.0664	0.0273	0.0117	0.0117
	Альт. 4	0.0117	0.1094	0.0352	0.1094	0.0141	0.0188	0.0012	0.0055	0.0313	0.0234	0.0000	0.0273	0.1250	0.0508	0.0234	0.0430
	Альт. 5	0.0156	0.1406	0.0469	0.1250	0.0211	0.0219	0.0016	0.0063	0.0547	0.0313	0.0000	0.0313	0.1328	0.0625	0.0313	0.0469

Таким образом, итоговые значения основных показателей эффективности для каждой альтернативы являются следующими:

1) Ручной подход – $P = \langle 0{,}0000;\ 0{,}0352;\ 0{,}0078;\ 0{,}0156\rangle$, $A = 0.0113$, $RM = 0.0176$, $RH = 0.0127$;

2) Алгоритмический подход – $P = \langle 0.0039, 0.0391, 0.0117, 0.0469\rangle$, $A = 0.05$, $RM = 0.0176$, $RH = 0.0371$;

3) Интеллектуальный подход – $P = \langle 0.0039, 0.0508, 0.0156, 0.0313\rangle$, $A = 0.0301$, $RM = 0.0156$, $RH = 0{,}0293$;

4) Полный перебор – $P = \langle 0.0117, 0.1094, 0.0352, 0.1094\rangle$, $A = 0.0395$, $RM = 0.0205$, $RH = 0.0605$;

5) Генетический подход – $P = \langle 0.0156, 0.1406, 0.0469, 0.1250\rangle$, $A = 0.0508$, $RM = 0.0293$, $RH = 0.0684$.

Исходя из сравнительного анализа показателей эффективности выделим генетический подход, который превосходит по результативности этапов, выигрывает по общей оперативности, умеренно превышает по машинной и «человеческой» ресурсоэкономности. Несмотря на проигрыш по отдельным показателям (уступает на 1-ом этапе алгоритмическому подходу по взвешенному коэффициенту сокращения времени выполнения и ручному по машинной ресурсоэкономности), качественно повышает результативность нейтрализации уязвимостей в программе при одновременном неухудшении общей оперативности и ресурсоэкономности процесса. Следуя проведенной специализации общей модели эффективности процесса мониторинга и реагирования на инциденты ИБ к предметной области нейтрализации уязвимостей в программе, она показала свою работоспособность и может применяться, как

оценочный инструмент при осуществлении комбинаторной оптимизации по выбору рациональных способов.

3. Отчет о результатах апробации научно обоснованных рациональных способов мониторинга и реагирования на типовые инциденты информационной безопасности в ЦИИ

Цель апробации способов реагирования на типовые инциденты ИБ для выполнения регламентированных действий специалистов в области защиты информации заключается в практическом подтверждении реализуемости и применимости научно обоснованных рациональных способов реагирования на возможные инциденты ИБ в ЦИИ.

Задачи апробации: во-первых, проверить возможность выполнения регламентированных действий операторами и администраторами безопасности информационных систем, входящих в состав ЦИИ, в рамках мероприятий по мониторингу ИБ в ЦИИ; во-вторых, проверить возможность выполнения регламентированных действий операторами и администраторами безопасности информационных систем, входящих в состав ЦИИ, в рамках мероприятий по управлению компьютерными инцидентами (КИ) ИБ в ЦИИ; в-третьих, сформировать организационно-технические предложения по реализации рациональных способов мониторинга и реагирования на возможные инциденты в ЦИИ и возможность комплексирования предлагаемых решений с корпоративными информационными системами с учетом уровня развития информационных систем.

Для организации апробации были разработаны и утверждены Программа и методики апробации (проверок), которые проводились в Санкт-Петербургском университете ГПС МЧС России, в том числе с применением авторских моделирующих инфраструктур и специализированных программных средств обработки экспериментальных данных.

Объем проверок (апробации) способов реагирования на типовые инциденты информационной безопасности для выполнения регламентированных действий специалистов в области информационной безопасности и защиты информации определялся требованиями методик, включенных в программу, в части мониторинга и реагирования на типовые инциденты, а также дополнительными проверками - частными методиками. Результаты проверок (апробации) оформлены Протоколами проверок и сведены в Акт о результатах проверок (апробации).

Апробация (проверки) антропометрического подхода к выявлению эффектов инфраструктурного деструктивизма

Частная методика проверки (апробации) распределенной системы ситуационного мониторинга (РССМ) на базе сетевой инфраструктуры основана на процедурах, которые описываются следующими шагами [37].

Шаг 1 – Построение сетевой инфраструктуры для заданного количества Веб-серверов. Развертывается и конфигурируется сетевая инфраструктура с определенным количеством, состоящая из трех сетевых сегментов, в каждом из которых располагалось по 36 клиентов. Запросы от них поступают через сеть Интернет, последующий балансировщик нагрузки, маршрутизатор к переменному (в зависимости от сценария) количеству Веб-серверов. Обеспечивается стабильная работа всех сетевых клиентов с Веб-серверами.

Шаг 2 – Имитация работы распределенной системы ситуационного мониторинга. Производится запуск РССМ в штатном режиме с имитацией деятельности пользователей АРМ по получению информации с Веб-серверов. Своевременность доставки информации напрямую зависит от количества Веб-серверов и их взаимодействия в аспекте обслуживания пользовательских запросов.

Шаг 3 – Сбор данных о работе сетевой инфраструктуры. На данном шаге происходит сбор различной информации (например, логов работы Веб-серверов) при обслуживании пользовательских запросов в штатном режиме работы РССМ. Шаг выполняется параллельно с Шагом 2.

Шаг 4 – Обработка данных о работе сетевой инфраструктуры. Процедура выполняется после завершения имитации работы РССМ и предназначена для обработки информации, собранной в процессе с Веб-серверов.

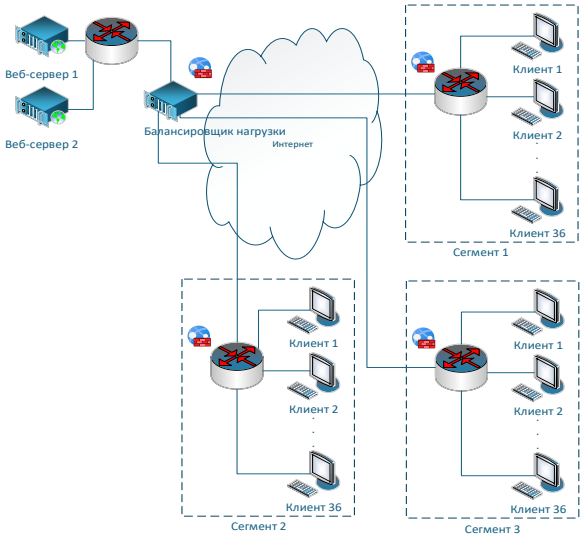
Шаг 5 – Выявление эффектов ИД. Применяется алгоритмы выявления ИД к информации, собранной на Шаге 3 и обработанной на Шаге 4. Например, если на некотором сервере за счет кэширования последующие запросы были выполнены более оперативно, то это будет считаться положительным эффектом; и, наоборот.

Примечание: В основе функционирования этой стадии лежит принцип, изложенный в источнике [39]. Характерные модели поведения запросов определяются через систему правил, имитирующих человеческое взаимодействие, с использованием продукционной модели знаний. Каждый запрос, извлекаемый из журналов событий, проходит анализ на соответствие пространственно-временным шаблонам, выстроенным в определенной последовательности.

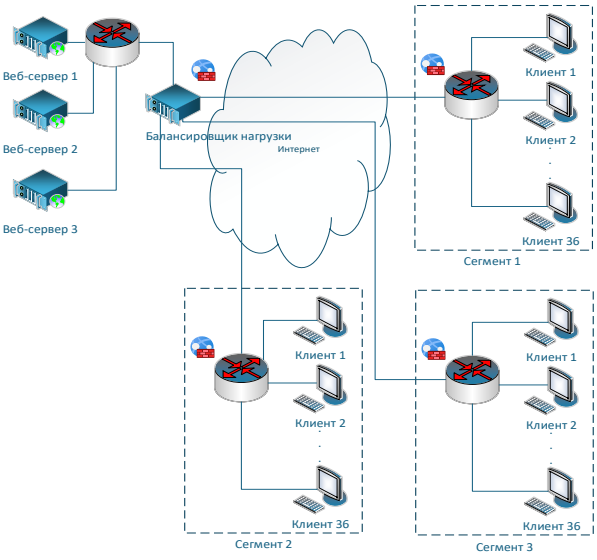
На старте алгоритма производится загрузка данных о запросах. Для каждого из них создается группа взаимодействующих запросов, подлежащих анализу, и проверяется наличие между ними причинно-следственных отношений. При обнаружении такой связи генерируется набор параметров для описания количественных показателей этого антропоморфного взаимодействия, что в итоге дает возможность классифицировать тип возникающего эффекта. Расчет количественных показателей базируется на реальной длительности обработки запросов. Для этого определяется доля времени, затрачиваемого на каждый тип взаимодействия, по сравнению с совокупным временем выполнения всех исследуемых запросов; результат выражается в процентах. Поскольку время обработки запросов значительно меньше общего времени работы РССМ, то возможно выявить и определить процентное соотношение каждого из типов эффектов ИД к общему времени. Реализация процедур частной методики осуществлялась по 3 сценариям (в зависимости от количества Веб-серверов).

Сценарий 1. Обработка данных в системе происходит с помощью 2 Веб-серверов; схема архитектуры советующей сетевой инфраструктуры РССМ приведена на рисунке 15а. Согласно предварительным экспертным оценкам, была спрогнозирована недостаточная производительность системы.

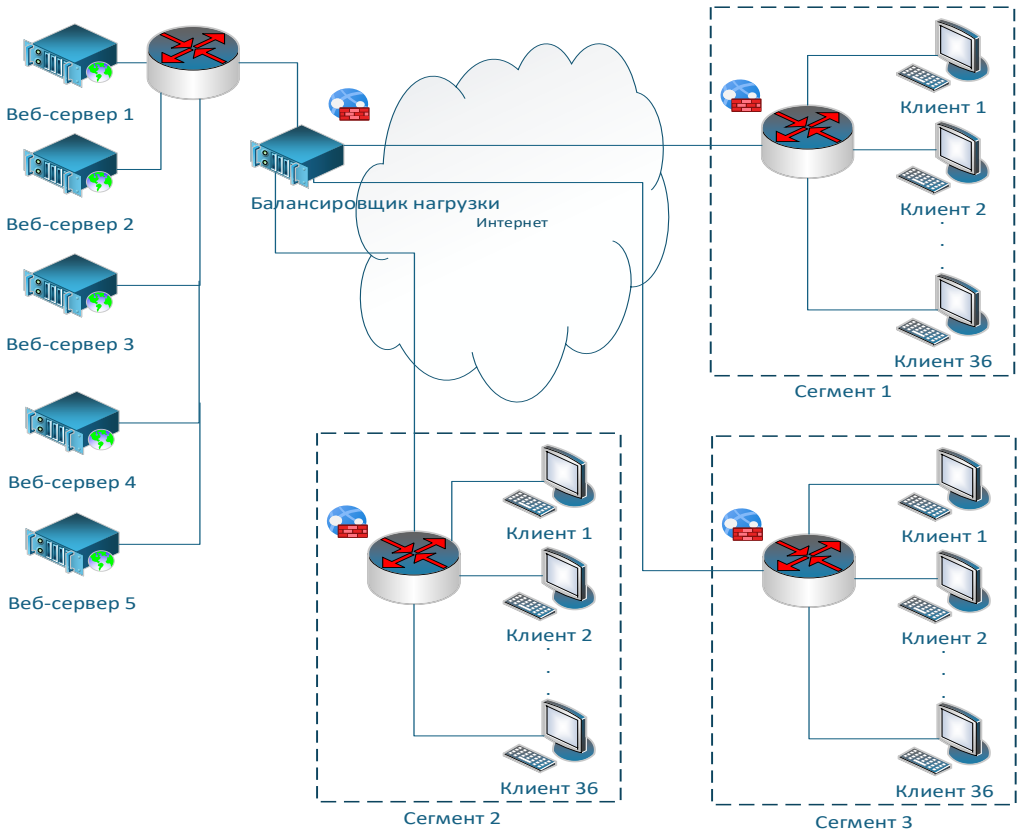
Сценарий 2. Обработка данных в системе происходит с помощью 3 веб-серверов; схема архитектуры советующей сетевой инфраструктуры РССМ приведена на рисунке 15б. Согласно экспертным оценкам, была спрогнозирована достаточная производительность системы без угрозы ДВ ИГ для Веб-серверов.



а) Сценарий 1 – 2 сервера



б) Сценарий 2 – 3 сервера



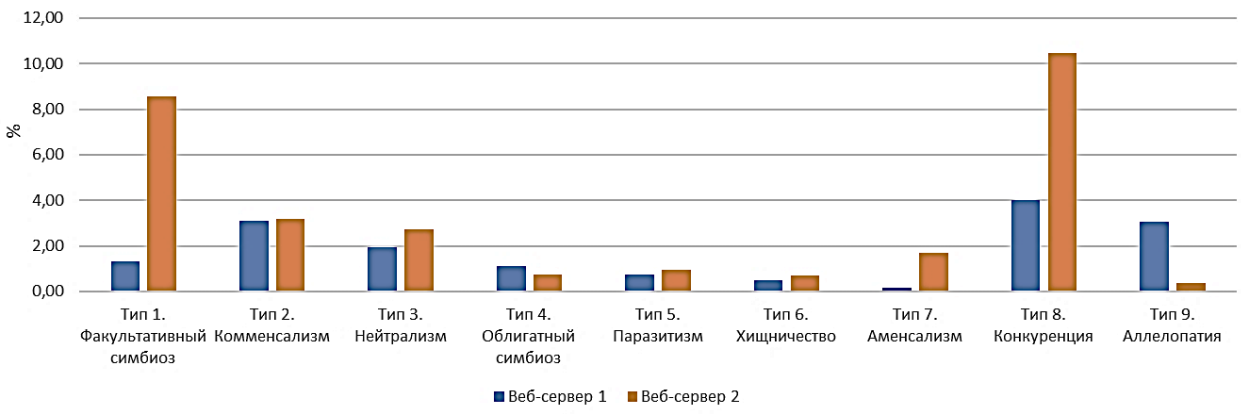
в) Сценарий 3 – 5 серверов

Рисунок 15 – Схема архитектуры РССМ с различным количеством Веб-серверов

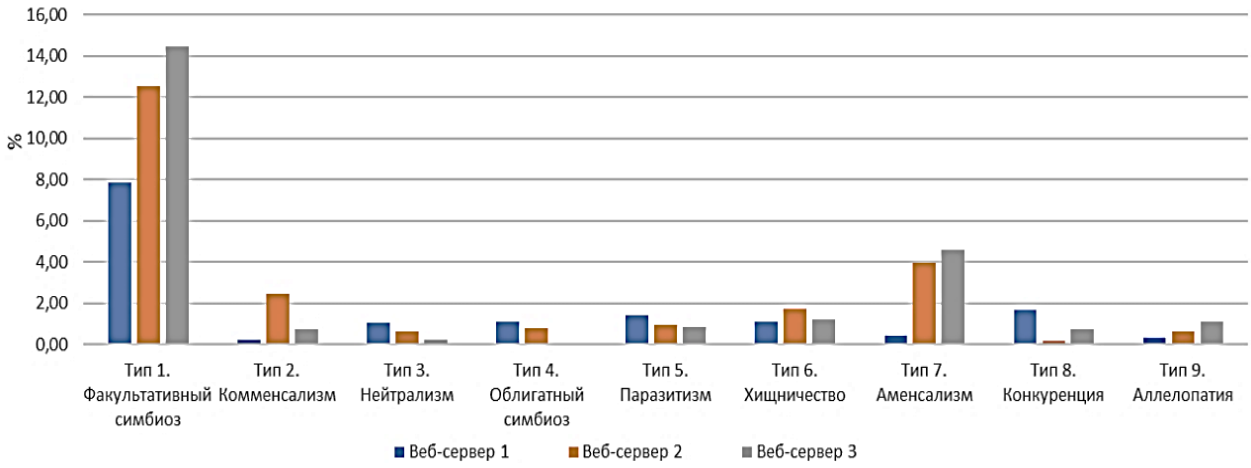
Сценарий 3. Обработка данных в системе происходит с помощью 3 Веб-серверов; схема архитектуры соответствующей сетевой инфраструктуры РССМ приведена на рисунке 15в. Согласно предварительным экспертным оценкам, была спрогнозирована достаточная производительность системы при угрозе ДВ ИГ для Веб-серверов. Как результат, РССМ гипотетически будет работать нестабильно (например, «зависание», большое время реагирования и др.).

Согласно частной методике проведена проверка способа выявления эффектов ИД по трем Сценариям, диаграммы которых представлены на рисунке 16.

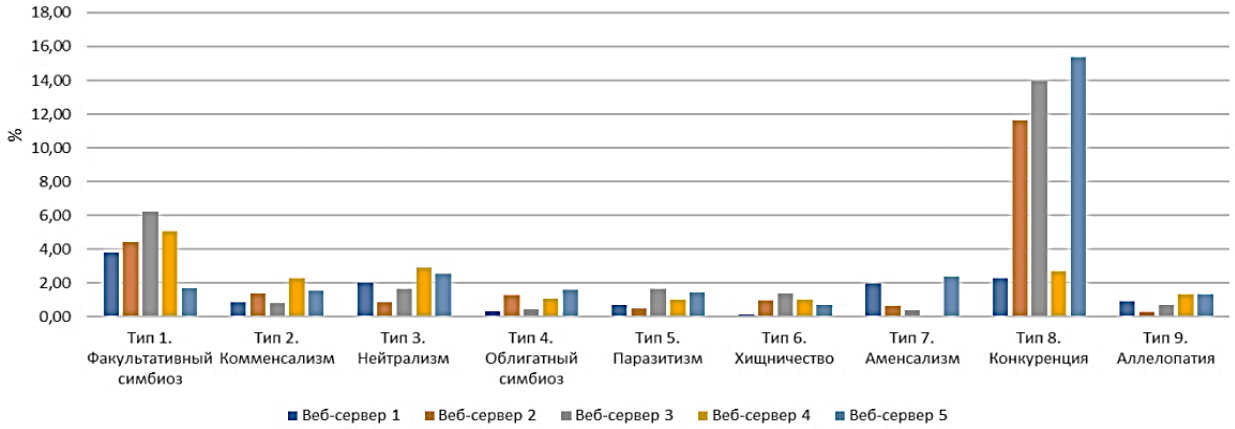
Выполнение всех шагов частной методики проверки (апробации) является однотипным, приводятся только итоговые результаты по каждому из сценариев.



а) Сценарий 1



б) Сценарий 2



в) Сценарий 3

Рисунок 16 – Эффекты ИД в РССМ

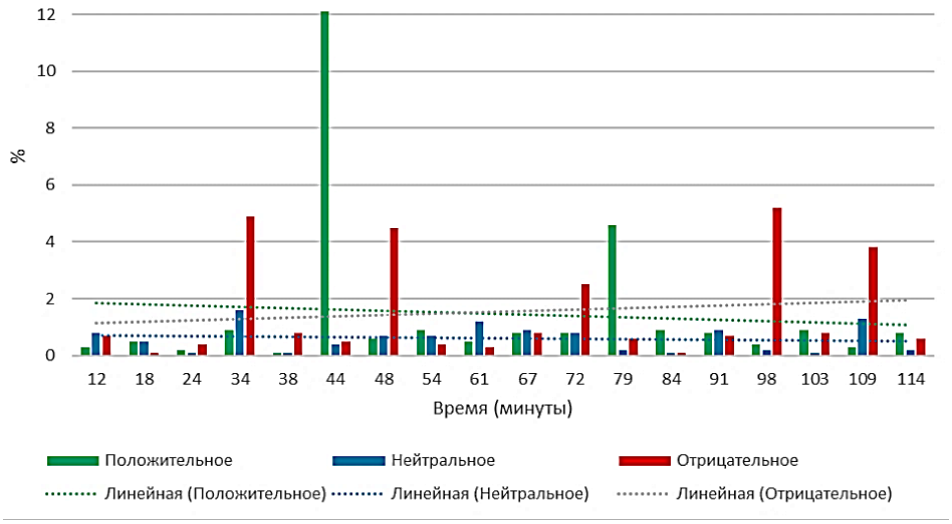
Сценарий 1. Диаграмма взаимного влияния Веб-серверов с указанием типа эффекта ИД и относительной длительности его проявления (в процентах к общему времени) приведена в виде диаграмм на рисунке 16а [40]. Сводная диаграмма итоговых «знаков» эффектов ИД, ассоциируемых с рисками их возникновения, на промежутках в 12 минут для данного сценария приведена на рисунке 17а [41].

Для удобства отображения результатов было проведено объединение типов антропоморфического взаимодействия процессов в группы – в результате удалось классифицировать динамику взаимного влияния сервисов [37, 42, 43]:

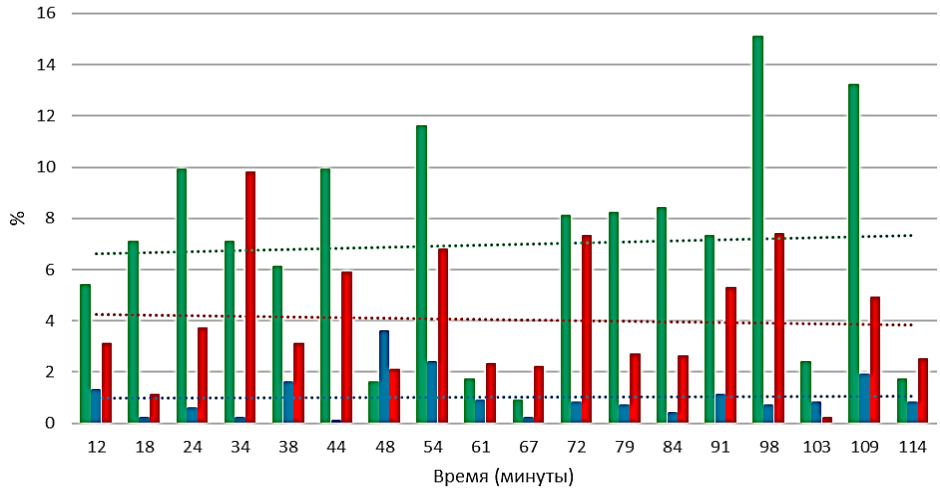
- положительный класс: «Тип 1. Облигатный симбиоз», «Тип 2. Факультативный симбиоз», «Тип 3. Комменсализм»;
- нейтральный класс: «Тип 4. Нейтрализм»;
- отрицательный класс: «Тип 5. Паразитизм», «Тип 6. Хищничество», «Тип 7. Аменсализм», «Тип 8. Аллелопатия», «Тип 9. Конкуренция».

Сценарий 2. Количество Веб-серверов: 3. Диаграмма взаимного влияния Веб-серверов с указанием типа эффекта ИД и относительной длительности его проявления (в процентах к общему времени) приведена в виде диаграмм на рисунке 16б [60]. Анализ диаграммы позволил установить, что взаимодействие процессов Веб-серверов описывается эффектами ИД, аналогичными характерным для Сценария 1, однако интенсивность взаимодействия имеет большее значение (с точки зрения временных оценок). Также установлено, что эффект ИД «Тип 1. Факультативный симбиоз» более выражен (высота его столбцов существенно превосходит аналогичные для «Тип 2. Конкуренция») – это обосновывается взаимопомощью Веб-серверов и, как результат, общему повышение работоспособности РССМ. Сводная диаграмма итоговых

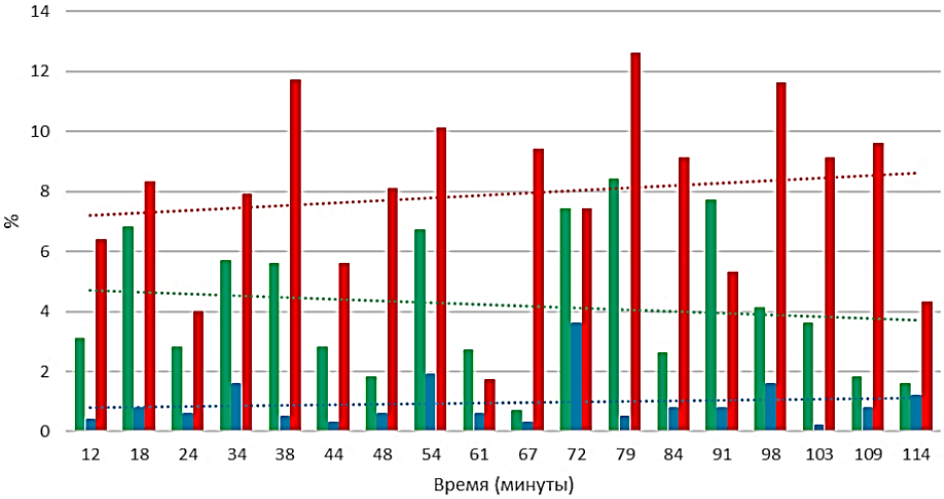
«знаков» эффектов ИД, ассоциируемых с рисками их возникновения, для данного сценария приведена на рисунке 17б [41].



а) Сценарий 1



б) Сценарий 2



в) Сценарий 3

Рисунок 17 – Диаграмма рисков возникновения эффектов ИД в РССМ

Сценарий 3. Количество Веб-серверов: 5. Диаграмма взаимного влияния Веб-серверов с указанием типа эффекта ИД и относительной длительности его проявления приведена в виде диаграмм на рисунке 16в [40]. Анализ диаграммы позволил установить, что эффект ИД «Тип 2. Конкуренция» существенно более выражен, чем «Тип 1. Факультативный симбиоз» (что наглядно видно по высоте столбцов для обоих эффектов) – это говорит о взаимно-негативном взаимодействии Веб-серверов и ведет к общему снижению работоспособности РССМ. Также установлено, что для данной системы угроза ИГ способна привести к полной блокировке работы ситуационного центра. Тем самым, для имеющейся РССМ удалось повысить точность оценки эффектов ИД и выполнить прогнозирование угроз ИГ. Сводная диаграмма итоговых «знаков» эффектов ИД, ассоциируемых с рисками их возникновения, для данного сценария приведена на рисунке 17в [41].

Анализ результатов проведенной проверки (апробации) антропометрического подхода к выявлению эффектов ИД на макро-уровне позволил сделать следующие предварительные выводы.

Во-первых, эффекты ИД возможно прогнозировать по динамике положительного, нейтрального и отрицательного взаимодействия и согласно их трендам.

Во-вторых, анализ поведенческой активности на основе антропоморфических типов позволяет по-новому оценивать характеристики распределенных систем.

В-третьих, на основе антропоморфических типов можно находить на ранних стадиях негативные взаимодействия серверов (и их сервисов), что позволит защитить распределенную системы от реализации тактики «T1499. Отказа в обслуживании» (согласно классификации MITRE ATT&CK), обусловленной в данном случае неверной настройкой системы [37].

В-четвертых, точка пересечения трендов будет являться бифуркационной, потенциально ведущей последнюю к необратимому саморазрушению.

Апробация (проверки) комплексных методов анализа и прогнозирования временных рядов инцидентов информационной безопасности в цифровой информационной инфраструктуре

В ходе апробации (проверки) были использованы статистические пакеты JASP, jamovi, а также язык R и интегрированная среда разработки Rstudio. Выбор указанных средств был обусловлен возможностью с их помощью автоматизировать большое количество задач прогнозирования, а в ряде случаев отказаться от разработки программных модулей. Реализована технология no-code. В частности, были применены следующие методы теории временных рядов: регрессионный анализ; экспоненциальное сглаживание; методы авторегрессии проинтегрированного скользящего среднего; байесовские методы пространства состояний; метод Prophet.

Такое значительное количество методов позволило произвести сравнительный анализ результатов исследования, выбрать лучшие модели временных рядов и решить задачи прогнозирования с их помощью, а также дать характеристику динамики кибератак на исследуемую информационную систему. Так как указанные методы разработаны на основе различных подходов, это позволяет учесть особенности анализируемых данных, реализовать модель Дж. Тьюки. Результаты проведенного графического анализа исследуемых временных рядов с помощью Rstudio, приведены на рисунке 18.

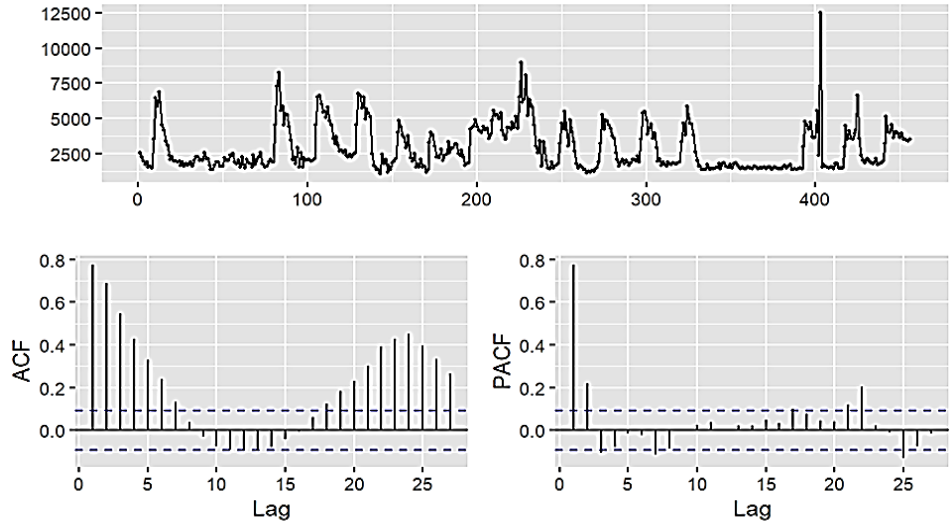
Выполненный анализ показал, что временные ряды имеют сезонные составляющие, выявлена автокорреляция уровней их. Также установлено, что имеется большая дисперсия случайных составляющих, что усложняет их исследование. Результаты описательной статистики данных временных рядов приведены в таблице 8.

Диаграммы, приведенные на рисунке 18, а также результаты описательной статистики позволяют сделать следующие выводы.

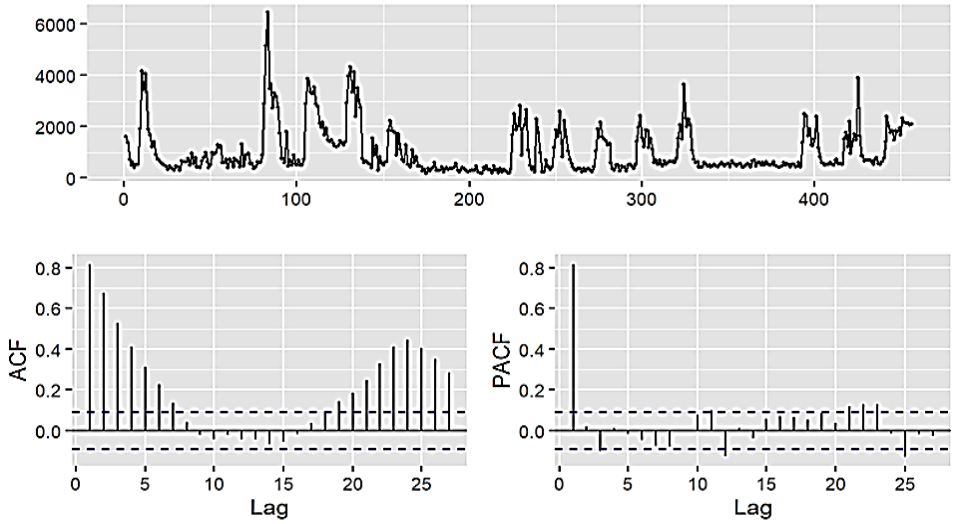
Во-первых, имеется большая вариативность уровней временных рядов. Для дальнейшего их исследования целесообразно решать задачи сглаживания или удалять аномальные наблюдения.

Во-вторых, коэффициент автокорреляции для каждого временного ряда значимо отличается от нуля. Следовательно, можно решать задачи прогнозирования.

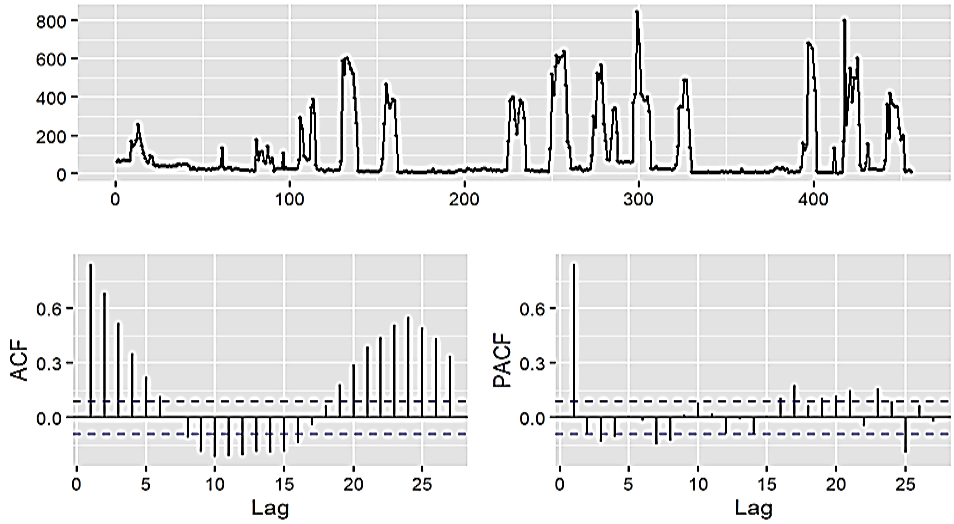
В-третьих, коррелограммы показывают, что существуют сезонные составляющие временных рядов с периодом сезонности, равным 24 часам. Наибольшее число атак приходится на период с 9 до 15 часов, т. е. на дневное время.



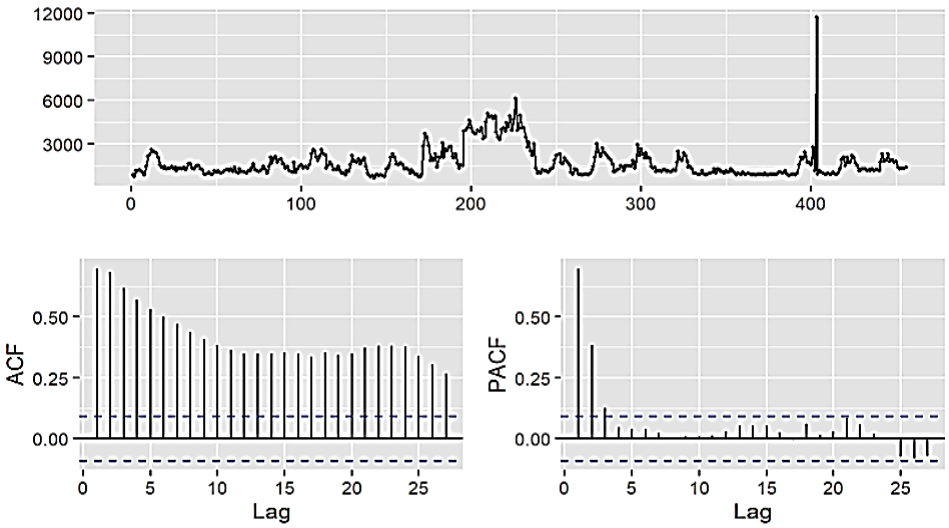
а) суммарное количество атак



б) количество предупреждений



в) критические атаки



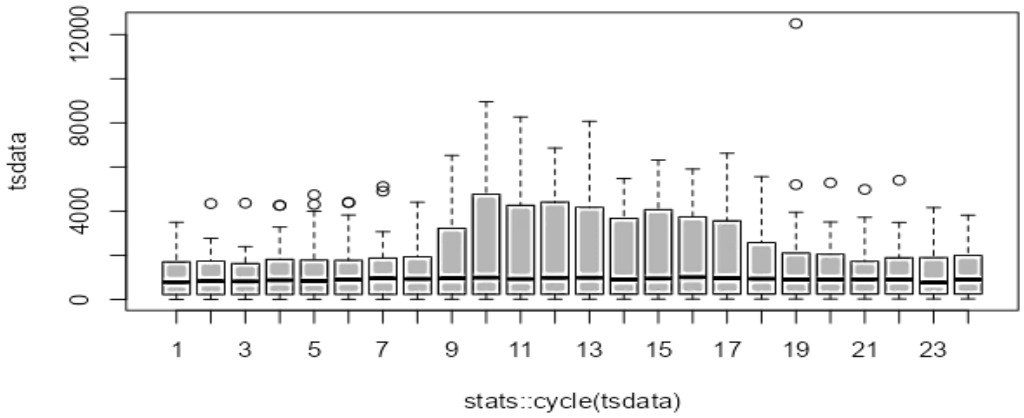
г) опасные атаки

Рисунок 18 – Точечная диаграмма и коррелограммы ACF и PACF функций

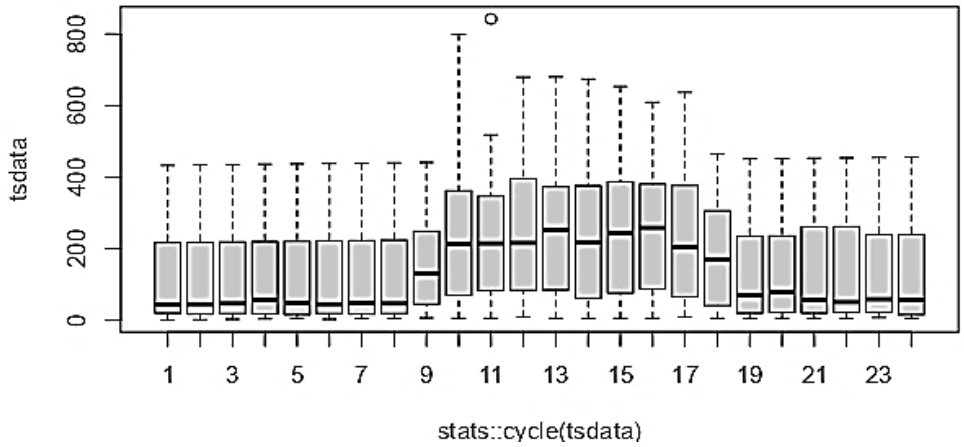
Таблица 8 – Описательная статистика

Показатели	Временные ряды			
	Критичные	Опасные	Предупреждающие	Всего
Существующие	456	456	456	456
Пропущенные	0	0	0	0
Среднее	110,746	1705,908	1046,156	2862,809
Стандартное отклонение	169,834	1032,702	918,354	1566,028
Дисперсия	28 843,557	1 066 000	843 374,923	2 452 000
Размах	839	11013	6277	11441
Минимум	3	664	178	1060
Максимум	842	11677	6455	12501
Start	63	898	1597	2558
End	10	1414	2105	3529
Автокорреляция первого порядка	0,844	0,698	0,82	0,776

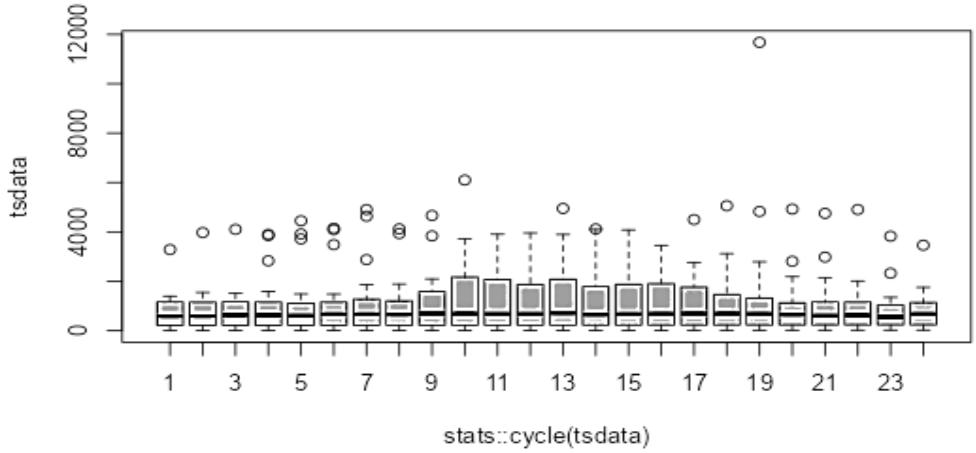
«Ящичные» диаграммы, которые были построены для исследуемых временных рядов, показаны на рисунке 19.



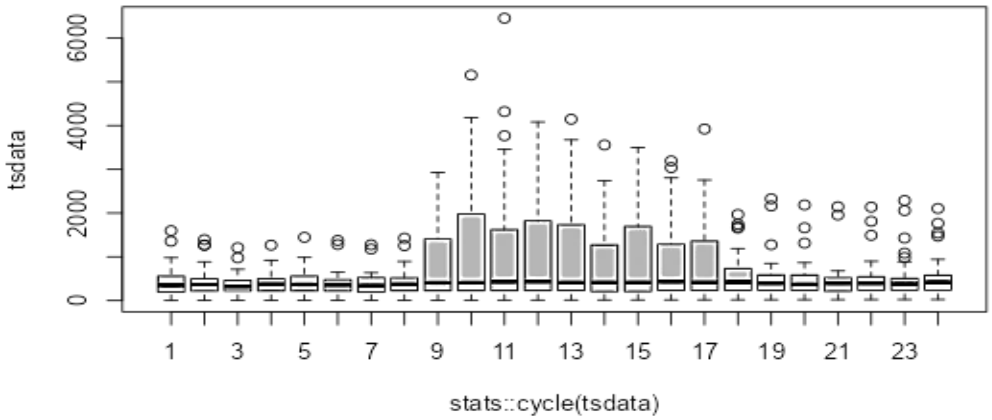
а) общего количества атак



б) критических атак



в) опасных атак



г) предупреждений

Рисунок 19 – «Ящичные» диаграммы количества атак

Наибольшие размеры «ящиков» также приходятся на дневное время. При этом характерно, что размах значений и длины верхних «усов» для анализируемого периода также максимальны. Диаграммы вновь подтверждают наличие выбросов, которые на диаграммах обозначены круглыми маркерами, расположенными над верхними «усами». Их число для разных рядов составляет от 19 до 33. Существуют и экстремальные значения, приходящиеся на 11 и 19 часов. Для построения моделей временных рядов потребовалось выполнить анализ их стационарности с помощью статистических критериев Дикки–Фуллера, KPSS [72] и Филиппа–Перона [73]. Результаты проверки данных статистических гипотез приведены в таблице 9.

Таблица 6 учитывает два вида стационарности временных рядов – DS- и TS-стационарность (*аббр. от англ.* Difference Stationary, разностно-стационарный и Trend Stationary, стационарный относительно тренда). В первом случае ряд является I(*k*)-интегрированным, например, случайным блужданием *I*(1).

Таблица 9 – Статистические критерии проверки стационарности временных рядов

Временной ряд	Критерий	Значение критерия	Значение лага	Уровень значимости (*p*-value)	Проверяемая гипотеза (*H*0): ряд ...
Предупреждения	Дики-Фуллера	–2,826	4	0,234	не стационарен
	Филлипса – Перона	–20,355	3	0,055	не стационарен
	KPSS, уровневая стационарность	0,182	4	0,100	уровнево-стационарен
	KPSS, стационарность тренда	0,135	4	0,071	стационарен по тренду
Опасные	Дики – Фуллера	–3,479	4	0,047	не стационарен
	Филлипса–Перона	–21,399	3	0,044	не стационарен
	Критерий KPSS, уровневая стационарность	0,160	4	0,100	уровнево-стационарен
	KPSS, стационарность тренда	0,147	4	0,050	стационарен по тренду
Критические	Дики – Фуллера	–3,092	4	0,124	не стационарен
	Филлипса–Перона	–32,156	3	0,010	не стационарен
	KPSS, уровневая стационарность	0,517	4	0,038	уровнево-стационарен
	KPSS, стационарность тренда	0,206	4	0,014	стационарен по тренду
Всех атак	Дики – Фуллера	–2,948	4	0,184	не стационарен
	Филлипса – Перона	–18,364	3	0,086	не стационарен
	KPSS, уровневая стационарность	0,158	4	0,100	уровнево-стационарен
	KPSS, стационарность тренда	0,143	4	0,055	стационарен по тренду

Приведение его к стационарному осуществляется с помощью нахождения разностного ряда *k*-го порядка, т. е. получения, так называемого, *I*(0)-стационарного процесса. При TS-стационарности, частным случаем которой является *I*(0)-ряд (уровнево-стационарный ряд), из наблюдаемых значений необходимо вычесть значения детерминированной функции, описывающей тренд.

Значение уровней значимости (*p*-value) для критериев Дики – Фуллера и Филлипса – Перрона больше, например 0,05, позволяют сделать вывод, что временной ряд предупреждений является DS-стационарным.

Чтобы его сделать уровнево-стационарным, необходимо построить ряд разностей. Итоговый временной ряд всех атак, а также временной ряд опасных атак не относятся к категории DS-рядов. С другой стороны, на уровне 0,055 они является уровнево-стационарными. Ряды не содержат тренда, и возможно, имеют ненулевое математическое ожидание своих уровней.

И наконец, анализ стационарности временного ряда критических атак с помощью четырех критериев приводит к противоречиям: первые два критерия на уровне 0,05 не позволяют сделать

вывод о DS-стационарности, а вторые два на этом же уровне значимости не отвечают на вопрос о стационарности по тренду или об уровневой стационарности.

Напомним, что в этом случае возможно использовать поправку Бонферрони, являющуюся методом противодействия проблеме множественных сравнений при применении семейства статистических гипотез. Необходимо продолжить исследование, например, с помощью моделей ARIMA.

Сравнительный анализ результатов построения моделей для одного из анализируемых временных рядов (всех атак) разными методами приведен в таблице 10.

Таблица 10 – Результаты оценки качества модели

Класс модели	CRPS	DSS	MAE	RMSE	R^2
Линейная регрессия	927,678	15,940	1405,956	1663,859	0,012
Байесовская модель BSTS	2901,963	19,174	1888,910	2454,455	0,108
Байесовская авторегрессионная модель	1235,123	16,621	1659,680	2127,404	0,115
Prophet	1426,080	19,452	1790,740	2154,794	0,381

Приведенные результаты показывают на низкое качество для различных классов моделей. Здесь в качестве критериев оценки их качества использованы:

- показатель ранжированной оценки вероятности (CRPS, Continuous Ranked Proba-bility Score) [74];
- показатель Дэвида – Себастьяни (DSS, Dawid-Sebastiani Score) [75];
- средняя абсолютная ошибка аппроксимации (MAE, Mean Absolute Error);
- квадратный корень из среднего квадрата ошибки аппроксимации (RMSE, Root Mean Squared Error);
- коэффициент детерминации (R^2).

Если последние три критерия применяются сравнительно часто, то первые два нуждаются в пояснении. Так, показатель CRPS – непрерывная ранжированная оценка вероятности, является обобщением показателя MAE для случая вероятностных прогнозов; его меньшему значению соответствует лучшая модель. Показатель DSS оценивает средние значения вектора отклонений наблюдаемых и прогнозных значений; здесь также меньшему значению критерия соответствует лучшая модель. Приведенные значения показывают, что нет лучшей по всем показателям модели, но по большинству показателей лучшей является линейная регрессионная модель, что довольно неожиданно. Однако ее построение и оценка качества такой модели также не позволяет сделать вывод о ее применимости.

Таким образом, без предварительной обработки с целью улучшения качества исходных данных задача прогнозирования не может быть решена. Поэтому дальнейшее исследование было проведено с учетом необходимости улучшения качества исходных данных за счет преобразования временных рядов. Известны различные методы таких преобразований, например, логарифмирование или извлечение квадратного корня наблюдаемых уровней. Их обобщением является преобразование Бокса – Кокса [76], при выполнении которого необходимо задать или найти значение параметра данного преобразования λ. Однако в этом случае затрудняется интерпретация полученных результатов, и возникает необходимость обратного преобразования.

В качестве альтернативы выберем методы фильтрации, в частности, метод ETS (Triple Exponential Smoothing, тройного экспоненциального сглаживания) [77]; система уравнений такого фильтра позволяет сгладить уровни временно́го ряда, тренд, а также сезонные составляющие. При этом модель задается трехзначным символьным кодом, первый знак которого определяет тип случайной составляющей «E», второй – тип тренда «T», третий – характеризует

сезонную составляющую «S». Такой код позволяет задать пятнадцать классов фильтров сглаживания. Будем использовать средства подгонки лучшего фильтра и оптимизации значений его параметров; их число зависит от выбора класса фильтра.

Для построения моделей временных рядов выполним композицию двух методов: экспоненциального сглаживания и авторегрессии ARIMA. Возможности применяемых программных средств позволяют использовать методологию autoML и подобрать с ее помощью нужные значения гиперпараметров, как для фильтров, так и для моделей ARIMA. Так как ее параметры подбираются автоматически, например, по значению информационных критериев, то при их определении возможно получение частных видов модели, например ARMA, AR и MA. В случае необходимости следует использовать расширения – SARIMA, ARIMAX, SARIMAX: их возможности позволят исследовать влияние рядов критических, опасных и атак-предупреждений на их общее число.

Лучшая модель фильтра позволяет получить сглаженные значения уровней временного ряда l_t.

По сглаженным значениям можно построить модель ARIMA, откликом для которой будет значение уровня ряда на момент времени t. Ее вид и значения гиперпараметров при этом определяются автоматически.

Первый временной ряд, содержащий все атаки, может быть представлен моделью ARIMA (1, 0, 2) с ненулевым математическим ожиданием:

$$l_t = 2858{,}52 + 0{,}82l_{t-1} + \varepsilon_t + 0{,}19\varepsilon_{t-2}, \varepsilon_t : N(0; 741). \quad (46)$$

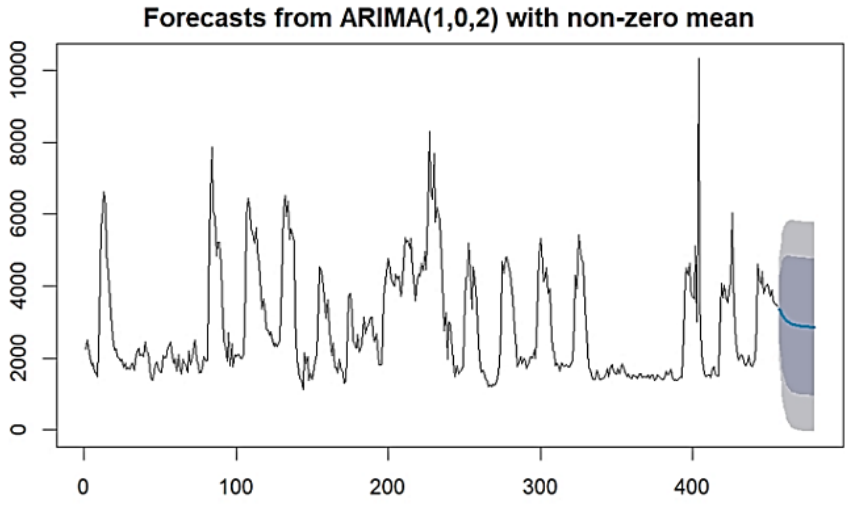
Фильтр сглаживания относится к классу фильтра с мультипликативной (M) ошибкой («E» = M), с отсутствием (N – None) тренда («T» = N) и сезонной составляющей («S» = N); его уравнение имеет вид:

$$l_t = 0{,}77y_t + 0{,}23l_{t-1}, l_{init} = 2246{,}2, \quad (47)$$

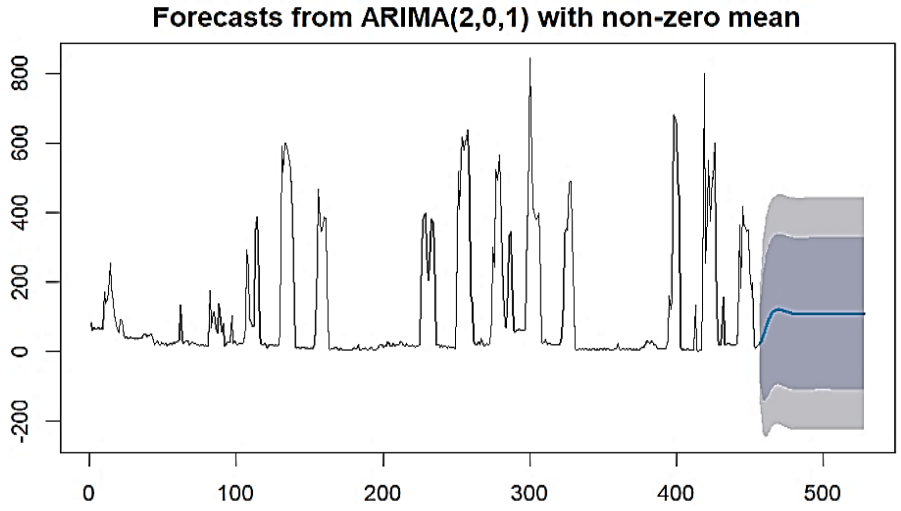
где l_{init} – начальное состояние фильтра.

Построенная модель временного ряда позволяет выполнить интервальную оценку условного математического ожидания прогноза.

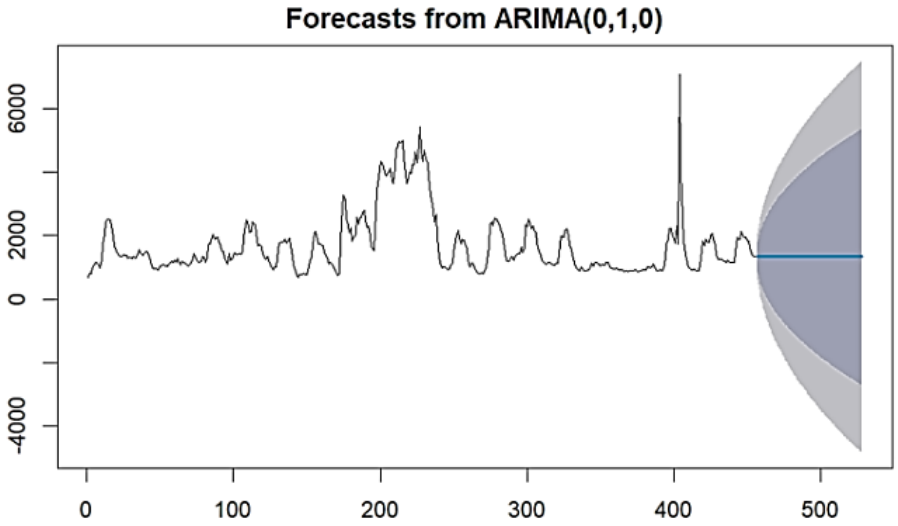
На рисунке 20 приведена диаграмма прогнозирования на три дня с построением верхней и нижней границ 80- и 95-процентных доверительных интервалов.



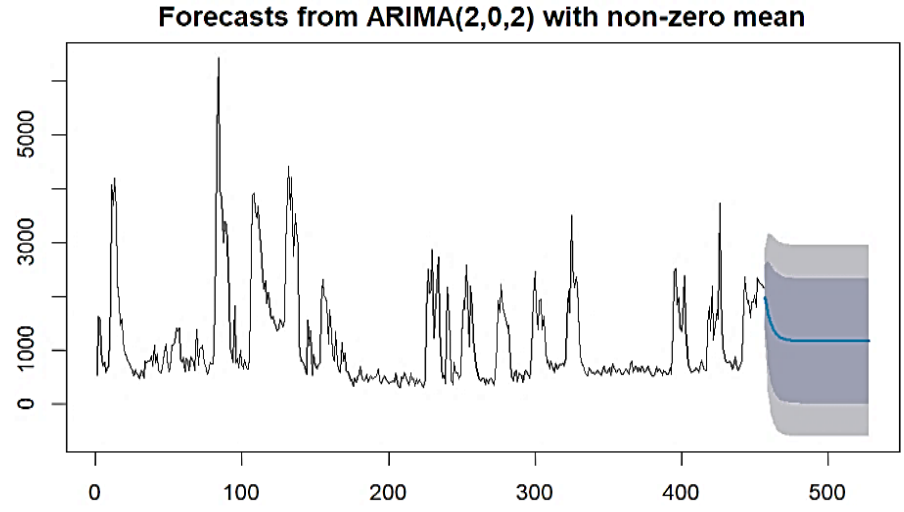
а) общего количества атак



б) критических атак



в) опасных атак



г) предупреждений

Рисунок 20 – Диаграмма прогнозирования уровней временного ряда

Сравнительно небольшая их ширина и медленный ее рост позволяют увеличить горизонт прогноза и с учетом знаков коэффициентов в уравнении предположить, что в среднем, в недалеком будущем общий поток атак не увеличится.

Аналогично решены задачи построения модели фильтра сглаживания, уравнения ARIMA и прогнозирования для других временны́х рядов. Так, временно́й ряд, содержащий критические атаки, может быть представлен моделью ARIMA (2, 0, 1) с ненулевым математическим ожиданием, имеющей вид:

$$l_t = 110{,}34 + 1{,}69l_{t-1} + 0{,}75l_{t-2} + \varepsilon_t - 0{,}77\varepsilon_{t-1}, \qquad \varepsilon_t : N(0{,}28; 8). \tag{48}$$

Сглаженные уровни временного ряда l_t были получены с помощью фильтра простого экспоненциального сглаживания с параметром $\alpha = 1$ и инициальным значением фильтра равным 83,0. К сожалению, данный фильтр не сглаживает уровни временного ряда, поэтому диаграмма, приведенная на рисунке 18б, показывает большую ширину обоих доверительных интервалов. Возможной причиной такой ситуации может быть большое число выбросов и большой разброс значений уровней временного ряда. Тем не менее, полученный на три для кратковременный прогноз также позволяет сделать вывод о стационарности временного ряда, т. е. подтвердить результаты ранее проверенных статистических гипотез по различным критериям.

Временной ряд, содержащий сведения об опасных атаках, описывается моделью авторегрессии-скользящего среднего ARIMA (0, 1, 0):

$$l_t = l_{t-1} + \varepsilon_t;\ \varepsilon_t : N(0; 370). \tag{49}$$

Для сглаживания его уровней с помощью autoML был определен аддитивный фильтр тройного экспоненциального сглаживания ETS с мультипликативной случайной составляющей (MAN).

Уравнения данного фильтра сглаживания содержат два оцененных параметра для каждого из них и имеют вид:

$$\begin{gathered} l_t = 0{,}5l_{t-1} + 0{,}49(l_{t-1} + b_{t-1}), \\ b_t = 0{,}02(l_t - l_{t-1}) = 0{,}98b_{t-1}, \\ l_{init} = 704{,}1, b_{init} = -3{,}53. \end{gathered} \tag{50}$$

Инициальные значения уровня ряда l_{init} и тренда b_{init} позволяют применять данный фильтр для решения задач сглаживания.

Построенная модель ARIMA позволяет сделать вывод, что данный временной ряд является DS-рядом, т. е. имеет стохастический тренд. С ростом времени прогнозирования растет ширина доверительного интервала прогноза, что не позволяет решать задачи долгосрочного прогнозирования уровней временного ряда. Это подтверждается колоколообразным видом доверительных интервалов прогноза, приведенных на рисунке 18в.

Последний временной ряд также может быть представлен моделью ARIMA с параметром авторегрессии $p = 2$ и параметров скользящего среднего $q = 2$ с ненулевым математическим ожиданием, имеющей следующий вид:

$$\begin{gathered} l_t = 116827 + 0{,}69l_{t-1} + 0{,}01l_{t-2} + \varepsilon_t + 0{,}24\varepsilon_{t-1} + \\ +0{,}13\varepsilon_{t-2}, \varepsilon_t : N(0; 467). \end{gathered} \tag{51}$$

Сглаженные значения данного временного ряда получены с помощью соотношений:

$$\begin{gathered} l_t = 0{,}88l_{t-1} + 0{,}12(l_{t-1} + b_{t-1}), \\ b_t = 0{,}001(l_t - l_{t-1}) + 0{,}999b_{t-1}, \\ l_{init} = 376{,}1, b_{init} = 155{,}59. \end{gathered} \tag{52}$$

Диаграмма прогнозирования уровней временного ряда на три дня также показывает, что доверительный интервал прогноза сравнительно невелик, поэтому можно увеличивать горизонт прогноза.

Полученные модели прогнозирования сглаженных уровней позволяют сделать вывод, что все анализируемые временные ряды, кроме опасных атак, относящегося к нестационарному ряду «случайное блуждание», являются TS-стационарными, а их случайные составляющие могут быть описаны авторегрессионными зависимостями. Отметим, что все коэффициенты, приведенные в уравнениях моделей ARIMA, значимо отличаются от нуля на сравнительно высоком уровне. Данный вывод сделал с помощью статистического критерия Стьюдента.

Дальнейшее исследование может быть направлено на анализ компонентов временных рядов, в частности тренда, сезонной и случайной составляющей, несмотря на то, что фильтры экспоненциального сглаживания не позволили выявить сезонные составляющие.

С этой целью целесообразно использовать метод Prophet, который основан на подгонке аддитивных регрессионных моделей, включающих тренд g_t, сезонные колебания s_t, эффекты праздников h_t, а также случайную составляющую ε_t.

Выбор основан на том, что он хорошо работает в условиях годовой, недельной и ежедневной сезонности, реализован в языках аналитики R, Python, а также в их графических приложениям.

В общем виде аддитивная регрессионная модель временного ряда, построенная с помощью данного метода, принимает вид:

$$y_t = g_t + s_t + h_t + \varepsilon_{t,} \quad (53)$$

где g_t – тренд, s_t – совокупность сезонных составляюших, h_t – составляющая, которая учитывает эффекты праздников и других влиятельных событий, ε_t – случайная компонента.

Для выявления сезонных составляющих s_t используется разложение в ряд Фурье. При определении тренда применяются кусочные линейная или логистическая модели с использованием точек излома.

Для построения модели и выявления ее компонент используем платформу jamovi (https://www.jamovi.org), которую можно рассматривать как графическую надстройку языка R.

Результаты декомпозиции на основе построения кусочной линейной регрессионной модели с годовыми (yearly), недельными (weekly) и дневными (daily) колебаниями для исследуемых временных рядов приведены на рисунках 21–25.

Декомпозиция показывает, что имеются точки излома тренда для двух временных рядов (общего числа атак и числа опасных атак). Ряды могут содержать различные участки монотонности. Следовательно, с учетом их разведывательного анализа может возникнуть необходимость создавать их слайсы, и для каждого слайса строить модель временного ряда. Вероятно, это сможет повысить качество модели.

Построенные модели сезонной декомпозиции показывают, что анализируемые ряды ведут себя по-разному. Так, например, для ряда с опасными атаками наибольшее число атак в среднем приходится на выходные. А для ряда, содержащего общее число атак и число критических атак – на среду. Эта информация является существенной для выбора и обоснования моментов повышенной готовности системы обнаружения и ликвидации последствий вторжений. Отметим, что для всех временных рядов, наибольшее число атак приходится на утренние часы, что также немаловажно для планирования работы системы обеспечения кибербезопасности.

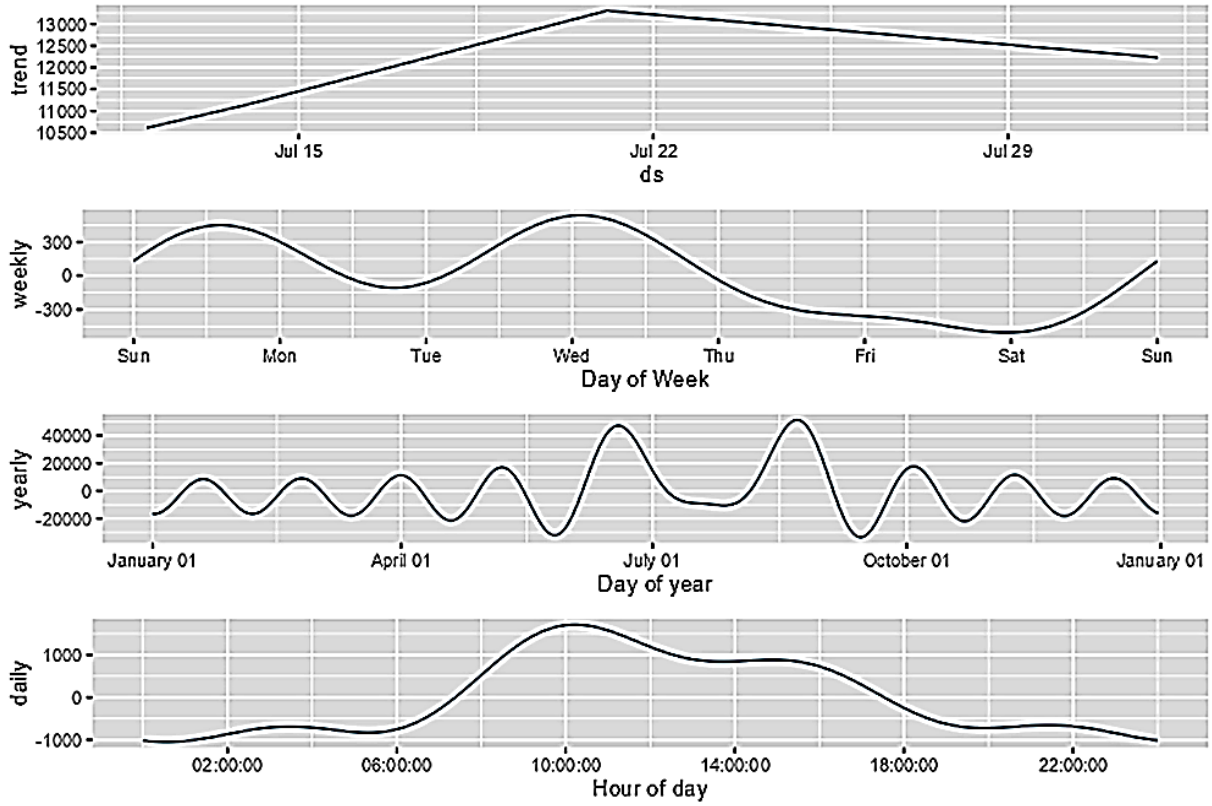


Рисунок 21 – Декомпозиция временного ряда общего числа атак

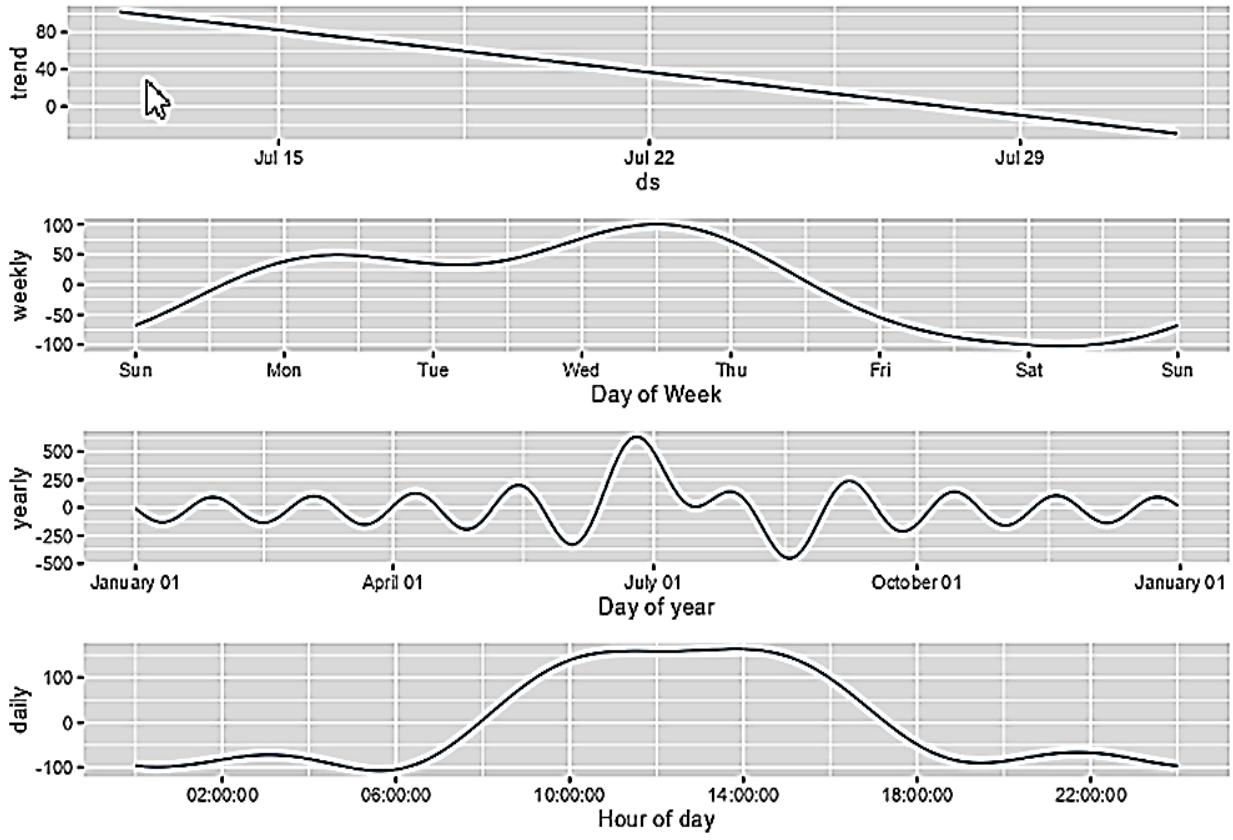


Рисунок 22 – Декомпозиция временного ряда числа критических атак

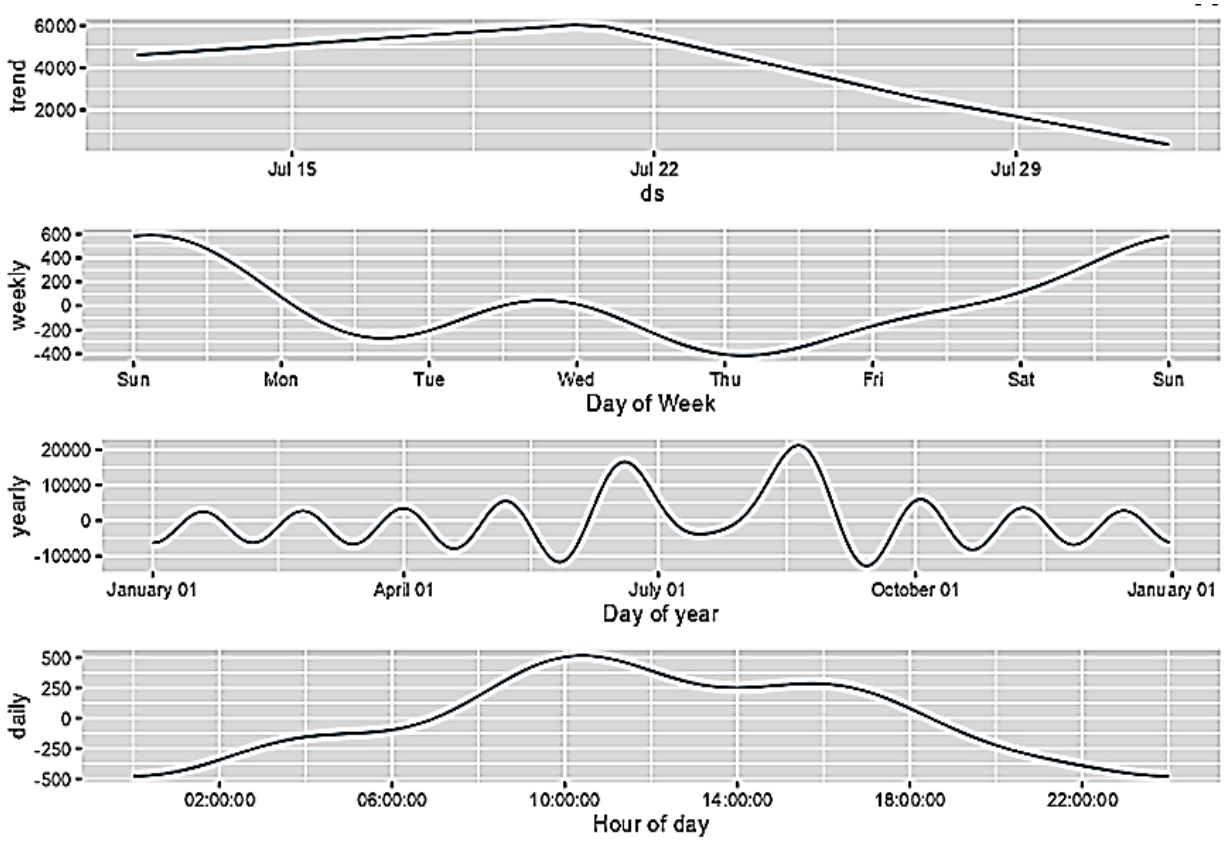


Рисунок 23 – Декомпозиция временного ряда числа опасных атак

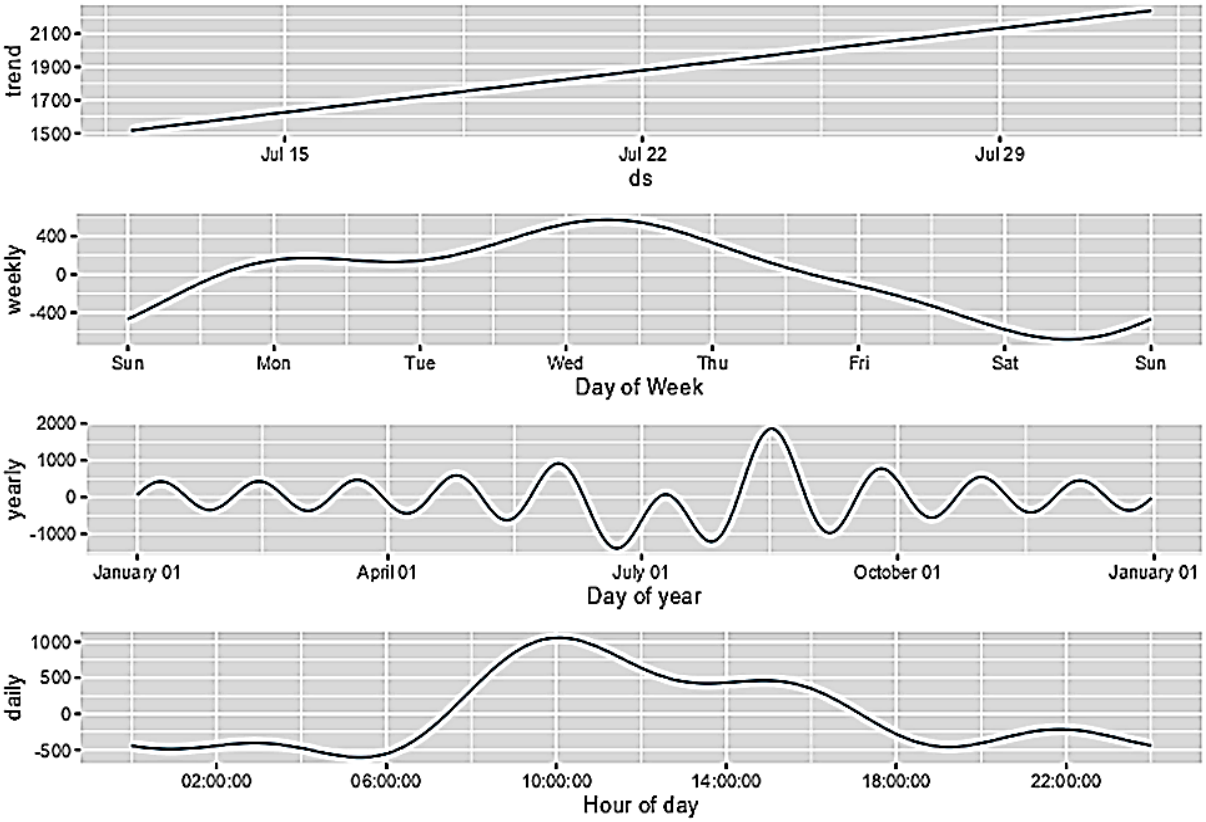


Рисунок 24 – Декомпозиция временного ряда числа предупреждений

Полученные результаты показывают, что при решении задач прогнозирования кибератак на объекты информационной инфраструктуры целесообразно использовать методы теории временных рядов. Ее разработанность, наличие большого количества методов и инструментальных средств позволяют реализовать комплексный подход, основанный на их последовательном применении, построении различных моделей, а затем их сравнительном анализе. Проведенное исследование четырех временных рядов на примере информационной системы ведомственного вуза показывает, что в силу большой вариативности, зашумленности измерений, наличия случайной составляющей с большой дисперсией, использование традиционных подходов к прогнозированию (например, методов регрессионного анализа), как правило, не эффективно. Так коэффициент детерминации построенной модели для одного из временных рядов составляет не более 0,01, а исправленное его значение становится даже отрицательным!

Попытка предобработать временные ряды с использованием преобразования Бокса – Кокса также не позволяет существенно улучшить качество решаемой задачи. Поэтому в исследовании использованы методы экспоненциального сглаживания, позволяющие уменьшить дисперсию временных рядов.

В дальнейшем результаты фильтрации могут быть использованы при решении задач прогнозирования.

К сожалению, даже это не позволяет существенно улучшить качество исходных данных. Поэтому решение задачи прогнозирования возможно только для небольшого горизонта прогноза, который в исследовании был задан равным трем суткам.

Вероятно, его можно увеличить, но предварительно следует исследовать характер анализируемого временного ряда и возможность такого подхода. В любом случае, большая зашумленность данных не позволяет решать задачи долгосрочного прогнозирования.

Апробация (проверки) способов мониторинга и реагирования на типовые инциденты информационной безопасности цифровой информационной инфраструктуры

Проведена апробация способов сбора данных о событиях безопасности и иных данных мониторинга от различных источников на примере средств ЕЦУ Dallas Lock.

Проверена апробация способов контроля, учета и анализа действий пользователей и администраторов на примере СЗИ DL.

Проведена апробация способов поэтапной проверки сбора и анализа данных о результатах контроля потоков информации сетевыми средствами защиты информации.

Проведена апробация способов выявления нарушений безопасности информации с применением типовых средств на примере Kaspersky Endpoint Security.

Проведена апробация способов разработки описаний выявленных уязвимостей на примере отечественного средства антивирусной защиты Kaspersky Endpoint Security.

Проведена апробация способов контроля установки обновлений безопасности программного обеспечения, включая сетевые и локальные средства защиты информации.

Проведена апробация способов контроля состава программно-технических средств, программного обеспечения и СЗИ (инвентаризация).

Проведена апробация способов контроля соответствия настроек программного обеспечения установленным требованиям к защите информации (политикам безопасности).

Проверена апробация способов контроля работоспособности (неотключения) программного обеспечения и СЗИ.

Проведена апробация способов формирования типовых перечней сведений, подлежащих сбору в ходе реагирования на КИ, включающих .

- определение перечня основных типов компьютерных инцидентов, встречающихся в организации;
- анализ существующих регламентов и стандартов по обработке инцидентов;

- установление требований к полноте и точности собираемых сведений, включая обязательные элементы каждого списка;
- составление шаблона типового перечня сведений, который должен заполняться при каждом инциденте;
- проверку контрольного заполнения с использованием заранее подготовленного сценария инцидентов.

Проведена апробация возможностей операторов в части блокировки сетевых атак с применением специализированных средств защиты информации.

Проведена апробация возможности отключения (изоляции, исключения) заражённого элемента сегмента в цифровой инфраструктуре.

Проведена апробация возможности выполнения регламентированных действий операторами и администраторами безопасности информационных систем, входящих в состав ЦИИ, в рамках мероприятий по управлению компьютерными инцидентами информационной безопасности в ЦИИ, в части изменения маршрутизации в ЦИИ.

Проведена апробация возможности выполнения регламентированных действий операторами и администраторами безопасности информационных систем, входящих в состав ЦИИ, в рамках мероприятий по управлению компьютерными инцидентами информационной безопасности в ЦИИ, в части локализации КИ.

Таким образом, результаты апробации научно обоснованных способов мониторинга и реагирования на инциденты информационной безопасности в цифровой информационной инфраструктуре подтвердили возможность выполнения регламентированных действий операторами и администраторами безопасности информационных систем, входящих в состав ЦИИ, в рамках мероприятий по мониторингу ИБ и управлению КИ ИБ в ЦИИ.

С целью реализации рациональных способов мониторинга и реагирования на возможные инциденты ИБ в ЦИИ состав объектовых средств мониторинга целесообразно расширить посредством сканеров уязвимостей.

С целью комплексирования решений по реализации рациональных способов мониторинга и реагирования на возможные инциденты в ЦИИ с внешними (корпоративными) информационными системами целесообразно регламентировать организацию информационно-логического взаимодействия с объектами, находящихся в поднадзорной деятельности организаций (подразделений) для учета состояния ИБ на поднадзорных объектах, например, при организации и выполнении функций государственной надзорной деятельности.

Заключение и выводы

Настоящие материалы обосновывают необходимость расширения классической парадигмы информационной безопасности за счет учета внутренних источников угроз, связанных с особенностями структуры и функционирования самой ЦИИ, а не только внешних нарушителей. Констатируется, что существующие нормативно-методические документы Регулятора хотя и допускают учет техногенных источников, но не предоставляют адекватного аппарата для идентификации и анализа угроз инфраструктурного генезиса.

На формальных схемах и практических примерах показано, что введение такого понятия, как *взаимодействие (программных) уязвимостей*, является полностью оправданным, а особенности различных типов взаимодействий говорят о том, что последние могут быть достаточно сложными и иметь непредсказуемые эффекты. Применение антропоморфизма для понимания взаимодействий оказалось достаточно удачным – как для типизации, так и для их интерпретации человеком. Предложенная формализация показала свою жизнеспособность. Все приведенные взаимодействия могут быть формализованы и учтены при расчете метрики уязвимостей программы (эффекта каждой из уязвимостей), что было показано при вычислении поправки к последней для гипотетического примера. Это позволяет перейти от субъективного оценивания к более формализованному, и как следствие, повысить эффективность автоматизированного в отдельных экземплярах ПО. Может найти применение при проверке

метрик уязвимостей программных обновлений, устанавливаемых в информационных системах ЦИИ в защищенном исполнении.

Введено и детально раскрыто понятие «деструктивное воздействие инфраструктурного генезиса» (ДВ ИГ), которое принципиально отличается от классических киберугроз. Его источником являются не внешние злоумышленники, а имманентные свойства самой цифровой инфраструктуры (ЦИИ), возникающие в процессе интеграции и взаимодействия ее компонентов. Обосновано существование феномена «инфраструктурный деструктивизм» как нового класса угроз, для которого характерны синергизм, кумулятивность и динамичность, способные привести к саморазрушению системы без внешней атаки. Для выявления и идентификации деструктивных взаимосвязей в ЦИИ также предлагается использовать антропоморфический подход, который позволяет с высокой чувствительностью классифицировать межобъектные и межсубъектные связи, выявить среди них деструктивно-образующие (формы антибиоза) и оценить их потенциал.

Проведена проверка (апробация) методики выявления эффектов инфраструктурного деструктивизма (ИД) в распределенной системе ситуационного мониторинга (РССМ); проведены эксперименты для 2, 3 и 5 веб-серверов РССМ. Исследованы межсервисные взаимодействия, которые классифицированы по антропоморфическим типам: симбиоз, паразитизм, конкуренции и др. Данные межсервисные взаимодействия объединены в группы: положительные, отрицательные и нейтральные взаимодействия. Анализ трендов отрицательных межсервисных взаимодействий позволил определить, что что с увеличением числа Веб-серверов возникают как положительные, так и негативные эффекты ИД. При 2 Веб-серверах система работает с ограниченной производительностью, при 3 – с достаточной, а при 5 – с угрозой блокировки из-за конкуренции и других негативных эффектов межсервисного взаимодействия. Анализ выявил возможность прогнозирования ИД по динамике межсервисных взаимодействий, что позволяет на ранних стадиях предотвращать нарушения работы системы, в том числе «отказ в обслуживании». Ранняя идентификация негативных эффектов ИД позволяет своевременно предпринимать меры для защиты системы от критических сбоев и атак, повышая её надежность и безопасность.

Установлено, что информационно-техническое взаимодействие (ИТВ) между подсистемами защиты информации при их интеграции является самостоятельным источником деструктивных воздействий инфраструктурного генезиса. Предложена концептуальная схема, описывающая процесс возникновения угроз в интегрированных системах защиты информации (СЗИ), где активным источником угроз выступает само ИТВ между модулями, пассивными источниками являются уязвимости, проявляющиеся в результате интеграции. Формально определены требования к протоколу ИТВ в ИСЗИ, построенной по модульному принципу.

На условных данных межсетевого экрана ведомственного вуза за 3 месяца построены временные ряды количества атак с часовым интервалом. Композиция методов анализа временных рядов ETS и ARIMA позволила выявлять важные закономерности в динамике кибератак и строить краткосрочные прогнозы, что может быть эффективно использовано для повышения оперативности реагирования на инциденты информационной безопасности. В частности, выявлены различные паттерны сезонности (для опасных атак пик приходится на выходные, для общих и критических – на среду), установлено время максимальной активности (утренние часы, что важно для планирования мер защиты).

Математический и методический аппарат, разработанный для обоснования рациональных способов мониторинга и реагирования на инциденты ИБ в ЦИИ, представляет собой комбинаторную оптимизационную модель, которая: учитывает множественные ограничения (совместимость методов и средств, компетенции персонала, выполнение всех необходимых задач), оценивает эффективность по трем критериям (результативность, оперативность и ресурсоэкономность – человеческая и машинная), имеет многоуровневую структуру (от параметров конкретных методов до интегральной эффективности всего процесса) и прошла апробацию на примере выбора подходов к реверс-инжинирингу для нейтрализации уязвимостей, где показала работоспособность. Как результат – модель позволяет обоснованно выбирать оптимальные комбинации методов и средств мониторинга и реагирования, максимизируя общую

эффективность процесса с учетом организационных и технических ограничений ведомственной цифровой информационной инфраструктуры.

Список литературы

1. Отчет о НИР шифр «Модель» [Разработка методологии и специальной технологии построения учебных дисциплин для освоения программ переподготовки и повышения квалификации должностных лиц МЧС России, осуществляющих профессиональную деятельность в области обеспечения информационной безопасности и защиты информации : отчет о НИР / руководитель А. В. Шестаков ; исполнители : А. Н. Метельков, М. В. Буйневич [и др.]; Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, Санкт-Петербургский университет ГПС МЧС России. – Санкт-Петербург, 2023 – 409 с. – Рег. № НИОКТР 123030100009-7.]
2. Отчет о НИР шифр «Гармония» [Разработка принципов, методологии и элементов технологии решения прикладных задач гармонизации нормативной правовой базы в части требований информационной и кибербезопасности в интересах МЧС России : отчет о НИР / руководитель А. В. Шестаков ; исполнители : М. В. Буйневич, А. В. Матвеев, А. В. Максимов [и др.]; Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий, Санкт-Петербургский университет ГПС МЧС России. – Санкт-Петербург, 2024 – 402 с. – Рег. № НИОКТР 124120300001-6.]
3. Методика оценки угроз безопасности информации: методический документ / Утвержден ФСТЭК России 05.02.2021 г. – URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-5-fevralya-2021-g (дата обращения 15.03.2025)
4. ГОСТ Р 22.0.05-2020 Безопасность в чрезвычайных ситуациях. Техногенные чрезвычайные ситуации. Термины и определения – М., 2020. – 12 с.
5. ГОСТ Р 51901.1-2002 Менеджмент риска. Анализ риска технологических систем (с Поправкой) – М., 2002. – 27 с.
6. *Буйневич М.В., Израилов К.Е.* Авторская метрика оценки близости программ: приложение для поиска уязвимостей с помощью генетической деэволюции // Программные продукты и системы. 2025. № 1. С. 89-99.
7. *Буйневич М.В., Израилов К.Е.* Сигнатурный поиск уязвимостей в машинном коде на базе генетической декомпиляции // Защита информации. Инсайд. 2025. № 2 (122). С. 8-17.
8. *Леонов Н.В., Буйневич М.В.* Проблемные вопросы поиска уязвимостей в программном обеспечении промышленных ИТ-устройств // Автоматизация в промышленности. 2023. № 12. С. 59-63.
9. *Леонов Н.В., Буйневич М.В.* Машинное обучение vs поиск уязвимостей в программном обеспечении: анализ применимости и синтез концептуальной системы // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 83-94.
10. *Buinevich M., Izrailov K., Vladyko A.* The life cycle of vulnerabilities in the representations of software for telecommunication devices // 18th International Conference On Advanced Communications Technology (ICACT-2016). 2016. PP. 430-435.
11. *Израилов К.Е.* Система критериев оценки способов поиска уязвимостей и метрика понятности представления программного кода // Информатизация и связь. 2017. № 3. С. 111-118.
12. *Patel J., Lee R., Kim H.* Architectural View in Software Development Life-Cycle Practices // 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007). 2007. PP. 194 - 199
13. *Skramstad T., Khan M.* A redefined software life cycle model for improved maintenance // Proceedings Conference on Software Maintenance. 1992. PP. 193 - 197.
14. *Havlice Z., Kunstar J., Adamuscinova I., Plocica O.* Knowledge in software life cycle // 7th International Symposium on Applied Machine Intelligence and Informatics. 2009. PP. 153-157.
15. *Buinevich M., Izrailov K.* Method and utility for recovering code algorithms of telecommunication devices for vulnerability search // 16th International Conference on Advanced Communication Technology (ICACT-2014). 2014. PP. 172-176.
16. *Buinevich M., Izrailov K., Vladyko A.* Method for partial recovering source code of telecommunication devices for vulnerability search // 17th International Conference On Advanced Communications Technology (ICACT-2015). 2015. PP. 76-80.

17. *Buinevich M., Izrailov K., Vladyko A.* Method and prototype of utility for partial recovering source code for low-level and medium-level vulnerability search // 18th International Conference on Advanced Communication Technology (ICACT-2016). 2016. PP. 700-707.

18. *Buinevich M., Izrailov K., Vladyko A.* Testing of Utilities for Finding Vulnerabilities in the Machine Code of Telecommunication Devices // 19th International Conference on Advanced Communication Technology (ICACT-2017). 2017. PP. 408-414.

19. *Mobley K.* Reverse Engineering for Software Performance Engineering // 14th Working Conference on Reverse Engineering (WCRE 2007). 2007. PP. 302-304

20. *Agarwal S., Aggarwal A.* Model driven reverse engineering of user interface — A comparative study of static and dynamic model generation tools // 2014 International Conference on Parallel, Distributed and Grid Computing. 2014. PP. 268-273.

21. *Буйневич М.В., Израилов К.Е.* Метод алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. 2012. № 12. С. 2-6.

22. *Буйневич М.В., Израилов К.Е.* Автоматизированное средство алгоритмизации машинного кода телекоммуникационных устройств // Телекоммуникации. 2013. № 6. С. 2-9.

23. *Буйневич М.В., Израилов К.Е., Щербаков О.В.* Структурная модель машинного кода, специализированная для поиска уязвимостей в программном обеспечении автоматизированных систем управления // Проблемы управления рисками в техносфере. 2014. № 3(31). С. 68-74.

24. *Буйневич М.В., Израилов К.Е., Щербаков О.В.* Модель машинного кода, специализированная для поиска уязвимостей // Вестник Воронежского института ГПС МЧС России. 2014. № 2(11). С. 46-51.

25. *Израилов К.Е.* Расширение языка «С» для описания алгоритмов кода телекоммуникационных устройств [Электронный ресурс] // Информационные технологии и телекоммуникации. 2013. № 2(2). С. 21-31.

26. *Покусов В.В.* Синергетические эффекты взаимодействия модулей системы обеспечения информационной безопасности // Информатизация и связь. 2018. № 3. С 61-67.

27. *Буйневич М.В., Покусов В.В., Израилов К.Е.* Эффекты взаимодействия обеспечивающих служб предприятия информационного сервиса (на примере службы пожарной безопасности) // Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2018. № 4. С. 48-54.

28. *Покусов В.В.* Особенности взаимодействия служб обеспечения функционирования информационной системы // Информатизация и связь. 2018. № 5. С. 51-56.

29. *Коробко И.* Централизованно меняем пароли локального системного администратора // Системный администратор. 2006. № 6 (43). С. 30-32.

30. *Taneski V., Heričko M., Brumen B.* Impact of security education on password change // 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2015. PP. 1350-1355.

31. Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах: Приказ ФСТЭК от 11 февраля 2013 г. № 17.

32. *Максимова Е.А.* Модели и методы оценки информационной безопасности субъектов критической информационной инфраструктуры при деструктивных воздействиях инфраструктурного генеза: дис. …. д-ра техн. наук: 2.3.6 / Максимова Елена Александровна. Санкт-Петербург, 2022. 448 с.

33. *Буйневич М.В., Израилов К.Е.* Антропоморфический подход к описанию взаимодействия уязвимостей в программном коде. Часть 1: Типы взаимодействий // Защита информации. Инсайд. 2019. № 5(89). С. 78-85.

34. *Буйневич М.В., Израилов К.Е.* Антропоморфический подход к описанию взаимодействия уязвимостей в программном коде. Часть 2: Метрика уязвимостей // Защита информации. Инсайд. 2019. № 6(90). С. 61-65.

35. *Максимова Е.А., Буйневич М.В.* Метод оценки инфраструктурной устойчивости субъектов критической информационной инфраструктуры // Вестник УрФО. Безопасность в информационной сфере. 2022. № 1 (43). С. 50-63.

36. *Максимова Е.А., Буйневич М.В., Шестаков А.В.* Проактивное управление информационной безопасностью субъектов критической информационной инфраструктуры как сложных организационных систем с динамически изменяющейся структурой // Вестник Воронежского института МВД России. 2023. № 2. С. 49-59.

37. *Русаков А.М., Максимова Е.А.* Проактивная оценка динамики рисков инфраструктурного деструктивизма для распределенной системы распознавания лиц // Защита информации. Инсайд. 2025. № 4(124). С. 66-71. EDN UMBYME.

38. *Русаков А.М.* Комплекс антропоморфических моделей поведенческого анализа процессов для обнаружения эффектов инфраструктурного деструктивизма // Инженерный вестник Дона. 2024. № 11(119). С. 391-404. EDN DLHUZQ.

39. *Русаков А.М.* Алгоритмическая реализация модели оценки эффектов инфраструктурного деструктивизма информационно-технологической инфраструктуры // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2025. № 1. С. 121-128. DOI 10.37882/2223-2966.2025.01.35.

40. Свидетельство о государственной регистрации программы для ЭВМ № 2022685869 Российская Федерация. Программное обеспечение системы моделирования межобъектных системных связей инфраструктурного характера в информационных системах: № 2022685248: заявл. 15.12.2022: опубл. 28.12.2022 / А. М. Русаков. EDN BYQWMV.

41. Свидетельство о государственной регистрации программы для ЭВМ № 2025614582 Российская Федерация. Интеллектуальная система поведенческого анализа процессов в информационно-технологической инфраструктуре на основе антропоморфических типов взаимодействия: заявл. 12.02.2025: опубл. 24.02.2025 / А. М. Русаков. EDN GRIKOF.

42. *Русаков А.М.* Прогнозирование рисков инфраструктурного деструктивизма с помощью антропоморфического подхода для сервисной архитектуры // Защита информации. Инсайд. 2025. № 2(122). С. 32-37.

43. *Русаков А.М.* Прогнозирование рисков инфраструктурного деструктивизма на основе анализа журналов событий облачной платформы Openstack // Актуальные проблемы прикладной математики, информатики и механики: Сборник трудов Международной научной конференции, Воронеж, 02–04 декабря 2024 года. Воронеж: Научно-исследовательские публикации, 2025. С. 955-960. EDN JFAPXX.

44. *Буйневич Д.В., Матвеев А.В., Покусов В.В.* Способ оценки информационно-технического взаимодействия. Часть 1. Модели информационных систем // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2020. № 3. С. 108-116.

45. *Буйневич Д.В., Матвеев А.В., Покусов В.В.* Способ оценки информационно-технического взаимодействия. Часть 2. Метрика безопасности // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2020. № 4. С. 95-102.

46. *Покусов В.В.* Синергетические эффекты взаимодействия модулей системы обеспечения информационной безопасности // Информатизация и связь. 2018. № 3. С. 61-67.

47. *Покусов В.В.* Оценка эффективности системы обеспечения ИБ. Часть 1. Показатели и модели представления // Защита информации. Инсайд. 2019. № 2 (86). С. 54-60.

48. *Покусов В.В.* Оценка эффективности системы обеспечения ИБ. Часть 2. Методика и результаты // Защита информации. Инсайд. 2019. № 3 (87). С. 64-72.

49. *Слукин С.В., Сметанкина Г.И.* Принципы формирования рациональных вариантов информационно-технического взаимодействия в АСУ МЧС России // Пожарная безопасность: проблемы и перспективы. 2018. Т. 1. № 9. С. 826-829.

50. *Левчунец И.В., Асхадеев А.И.* Абстрактная модель информационной безопасности при информационно-техническом взаимодействии автоматизированных систем // Совершенствование Единой государственной системы предупреждения и ликвидации чрезвычайных ситуаций и гражданской обороны Российской Федерации на современном этапе: сборник трудов XXX Международной научно-практической конференции. 2020. С. 71-75.

51. *Покусов В.В.* Формат протокола универсального информационно-технического взаимодействия в системе обеспечения информационной безопасности «УИТВ-СОИБ» // Телекоммуникации. 2019. № 9. С. 33-40.

52. *Израилов К.Е., Покусов В.В., Столярова Е.С.* Информационные объекты в системе обеспечения информационной безопасности // Теоретические и прикладные вопросы комплексной безопасности: материалы I Международной научно-практической конференции. Петровская академия наук и искусств. 2018. С. 166-169.

53. *Покусов В.В.* Формализация и определение корректности протокола информационно-технического взаимодействия (на примере интегрированной системы защиты информации) // Информатизация и связь. 2021. № 2. С. 55-68.

54. *Глухов Н.И., Наседкин П.Н.* Онтологические модели в процессе управления информационными рисками и информационной безопасности хозяйствующих субъектов // Информационные технологии и математическое моделирование в управлении сложными системами. 2020. № 2 (7). С. 24-31.

55. *Ниязова Р.С., Буданова Н.* Онтологическая модель процесса обеспечения информационной безопасности // Открытые семантические технологии проектирования интеллектуальных систем. 2015. № 5. С. 169-172.

56. *Буйневич М.В., Израилов К.Е., Покусов В.В*. Модель угроз информационно-технического взаимодействия в интегрированной системе защиты информации // Информатизация и связь. 2021. № 4. С. 66-73.

57. *Курта П.А.* Взаимодействие пользователя с информационной системой. Часть 2. Алгоритмы обнаружения недостатков // Известия СПбГЭТУ ЛЭТИ. 2020. № 10. С. 34-44.

58. *Курта П.А.* Взаимодействие пользователя с информационной системой. Часть 1. Схема взаимодействия и классификация недостатков // Известия СПбГЭТУ ЛЭТИ. 2020. № 8-9. С. 35-45.

59. *Сидоров Г.В.* Исследование модели угроз для частной сети IP-Телефонии // Наука в современном обществе: закономерности и тенденции развития: сборник статей по итогам Международной научно-практической конференции. 2018. С. 55-57.

60. *Короченцев Д.А., Кухтинов В.Н.* Программное средство построения частной модели угроз безопасности информационной системы персональных данных // Colloquium-journal. 2019. № 12-2 (36). С. 92-94.

61. *Воронин В.В., Сухоруков Я.П.* Аспекты разработки частной модели угроз безопасности информации в типовых информационных системах // Вестник Приамурского государственного университета им. Шолом-Алейхема. 2020. № 1 (38). С. 24-33.

62. *Шакирова Р.А., Заколдаев Д.А.* Разработка алгоритма процесса проектирования частной модели угроз безопасности персональных данных // Новый взгляд. Международный научный вестник. 2015. № 8. С. 165-168.

63. *Тищенко Е.Н., Шкаранда Е.Ю.* Алгоритмизация процесса формирования частной модели угроз безопасности персональных данных // Известия ЮФУ. Технические науки. 2013. № 12 (149). С. 180-187.

64. *Васильева О.В., Жигулин Г.П.* Способ формирования модели угроз и нарушителей информационной безопасности частной облачной инфраструктуры // Научно-технический вестник Поволжья. 2014. № 1. С. 78-80.

65. *Глазьев С.Ю.* Теория долгосрочного технико-экономического развития. М.: ВлаДар, 1993. EDN:YSXIUV

66. *Нильсен Э.* Практический анализ временных рядов. Прогнозирование со статистикой и машинное обучение. СПб.: Диалектика, 2021 544 с.

67. *Хайндман Р., Атанасопулос Дж.* Прогнозирование: принципы и практика. Пер. с англ. М.: ДМК Пресс, 2023. 458 с.

68. *Наумов В.Н., Буйневич М.В., Синещук М.Ю., Тукмачева М.А.* Анализ и прогнозирование временных рядов кибератак на информационную систему ведомственного вуза: возможности и ограничения методов // Труды учебных заведений связи. 2025. Т. 11. № 1. С. 118–131. DOI:10.31854/1813-324X-2025-11-1-118-131. EDN:OOPJJR

69. *Исаев С.В., Кононов Д.Д.* Исследование динамики и классификация атак на веб-сервисы корпоративной сети // Сибирский аэрокосмический журнал. 2022. Т. 23. № 4. С. 593–601. DOI:10.31772/2712-8970-2022-23-4-593-601. EDN:RUSJWB

70. *Zuzčák M., Bujok P.* Using honeynet data and a time series to predict the number of cyber attacks // Computer Science and Information Systems. 2021. Vol. 18. Iss. 4. PP. 1197–1217. DOI:10.2298/CSIS200715040Z

71. *Ларионов К.О.* Прогнозирование статистических данных атак на прикладное программное обеспечение // Проблемы современной науки и образования. 2021. № 6(163). С. 57–63. DOI:10.24411/2304-2338-2021-10606. EDN:PGVALC

72. *Hobijn B., Franses P.H., Ooms M.* Generalization of the KPSS-test for stationarity // Statistica Neerlandica. 2004. Vol. 58. Iss. 4. PP. 482–502. DOI:10.1111/j.1467-9574.2004.00272.x

73. *Phillips P.C.B., Perron P.* Testing for a Unit Root in Time Series Regression // Biometrika. 1988. Vol. 75. Iss. 2. PP. 335–346. DOI:10.1093/biomet/75.2.335. EDN:ILNEET

74. *Hersbach H.* Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems // Weather and Forecast. 2000. Vol. 15. Iss. 5. PP. 559–570. DOI:10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2

75. *Dawid A.P., Sebastiani P.* Coherent Dispersion Criteria for Optimal Experimental Design // Annals of Statistics. 1999. Vol. 27. Iss. 1. PP. 65–81.

76. *Bickel P.J., Doksum K.A.* An Analysis of Transformations // Journal of the American Statistical Association. 1981. Vol. 76. Iss. 374. PP. 296–311. DOI:10.2307/2287831

77. *Hyndman R.J., Koehler A.B., Snyder R.D., Grose S.* A state space framework for automatic forecasting using exponential smoothing methods // International Journal Forecasting. 2002. Vol. 18. Iss. 3. PP. 439–454.

78. *Cleveland R.B., Cleveland W.S., McRae J.E., Terpenning I.J.* STL: A Seasonal-Trend Decomposition Procedure Based on Loess // Journal of Official Statistics. 1990. Vol. 6. Iss. 1. PP. 3–33.

79. *Scott S., Varian H.R.* Predicting the Present with Bayesian Structural Time Series // SSRN Electronic Journal. 2014. Vol. 5. Iss. 1/2. PP. 4–23. DOI:10.1504/IJMMNO.2014.059942

80. *Мастицкий С.Э.* Анализ временных рядов с помощью R. 2020. URL: https://ranalytics.github.io/tsa-with-r (дата обращения 13.10.2025)

81. *Абдуллин Т.И., Баев В.Д., Буйневич М.В., Бурзунов Д.Д., Васильева И.Н., Галиуллина Э.Ф. и др.* Цифровые технологии и проблемы информационной безопасности: монография. СПб: СПГЭУ 2021. 163 с.

82. *Леонов Н.В.* Противодействие уязвимостям программного обеспечения. Часть 1. Онтологическая модель // Вопросы кибербезопасности. 2024. № 2 (60). С. 87-92. DOI: 10.21681/2311-3456-2024-2-87-92

83. *Леонов Н.В.* Противодействие уязвимостям программного обеспечения. Часть 2. Аналитическая модель и концептуальные решения // Вопросы кибербезопасности. 2024. № 3 (61). С. 90-95. DOI: 10.21681/2311-3456-2024-3-90-95

84. *He Y., Zhou H., Zhang S., Lu S., Yan Y., Li R., Gao Y.* Research on Evaluation Model and Algorithm of Information System Health State Based on Realtime Operation Data and Analytic Hierarchy Process // The proceedings of International Conference on Networking, Communications and Information Technology (Manchester, United Kingdom, 26-27 December 2021). 2020. PP. 353-356. DOI: 10.1109/NetCIT54147.2021.00077

85. *Шимчик Н.В., Игнатьев В.Н., Белеванцев А.А.* IRBIS: Статический анализатор помеченных данных для поиска уязвимостей в программах на C/C++ // Труды Института системного программирования РАН. 2022. Т. 34. № 6. С. 51-66. DOI: 10.15514/ISPRAS-2022-34(6)-4.

86. *Бровко Е.В., Зайцев В.В.* Метод поиска уязвимостей в программном коде с использованием свёрточной нейронной сети // Методы и технические средства обеспечения безопасности информации. 2024. № 33. С. 3-5.

87. *Jones A., Omar M.* Harnessing the Efficiency of Reformers to Detect Software Vulnerabilities // The proceedings of Congress in Computer Science, Computer Engineering, & Applied Computing (Las Vegas, NV, USA, 24-27 July 2023). 2023. PP. 2259-2264. DOI: 10.1109/CSCE60160.2023.00368.

88. *Кулагин И.И., Падарян В.А., Кошкин В.А.* О методах извлечения алгоритмов из бинарного кода // Труды Института системного программирования РАН. 2024. Т. 36. № 3. С. 139-160. DOI: 10.15514/ISPRAS-2024-36(3)-10.

89. *Гавшин Д.А.* Использование инструментов Ghidra для анализа файлов UEFI // Вестник молодых ученых Санкт-Петербургского государственного университета технологии и дизайна. 2021. № 4. С. 39-43.

90. *Xia B., Ge Y., Yang R., Yin J., Pang J., Tang C.* BContext2Name: Naming Functions in Stripped Binaries with Multi-Label Learning and Neural Networks // The proceedings of IEEE 10th International Conference on Cyber Security and Cloud Computing / 2023 IEEE 9th International Conference on Edge Computing and Scalable Cloud (Xiangtan, Hunan, China, 01-03 July 2023). 2023. PP. 167-172. DOI: 10.1109/CSCloud-EdgeCom58631.2023.00037

91. *Израилов К.Е.* Концепция генетической деэволюции представлений программы. Часть 1 // Вопросы кибербезопасности. 2024. № 1 (59). С. 61-66. DOI: 10.21681/2311-3456-2024-1-61-66.

92. *Израилов К.Е.* Концепция генетической деэволюции представлений программы. Часть 2 // Вопросы кибербезопасности. 2024. № 2 (60). С. 81-86. DOI: 10.21681/2311-3456-2024-2-81-86.