Федеральное государственное автономное образовательное учреждение
высшего образования «Пермский национальный исследовательский
политехнический университет»

На правах рукописи

МИТЮКОВ Евгений Алексеевич

ПОВЫШЕНИЕ НАДЕЖНОСТИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМИ ОБЪЕКТАМИ ПУТЕМ СОВЕРШЕНСТВОВАНИЯ УРОВНЯ ИХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Специальность 2.3.3 - Автоматизация и управление технологическими процессами и производствами (технические науки)

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата технических наук

НАУЧНЫЙ РУКОВОДИТЕЛЬ
доктор технических наук, профессор
Затонский Андрей Владимирович

Пермь 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность исследования. Указ Президента Российской Федерации «О национальных целях развития РФ на период до 2030 года» предусматривает в качестве одной из задач обеспечение темпа роста валового внутреннего продукта страны выше среднемирового при сохранении макроэкономической стабильности. Одним из механизмов достижения подобного роста является дальнейшая автоматизация производств, в том числе не включающих объекты критической информационной инфраструктуры (КИИ). По отношению к объектам КИИ и иным объектам действуют нормы Совета Безопасности РФ и Федеральной службы по техническому и экспортному контролю (ФСТЭК России), регламентирующие мероприятия по информационной безопасности (ИБ) автоматизированных систем управления (АСУ), автоматизированных систем управления технологическими процессами (АСУТП), автоматизированных систем управления производствами (АСУП), автоматизированных систем управления технической подготовкой производства (АСТПП), которые в совокупности формируют единую автоматизированную систему управления промышленными объектами (АСУ промышленными объектами). АСУ промышленными объектами является большой системой в силу наличия в ней тысяч или десятков тысяч элементов на всех уровнях от полевых интеллектуальных датчиков до рабочих мест пользователей. Рост количества устройств автоматизации, технологической связи и наблюдения, средств информатизации деятельности и подобных им порождает рост потенциальных уязвимостей в системах их защиты. Повышение надежности АСУ промышленными объектами, на объектах не отнесенных к КИИ, в части рисков, возникающих от кибер-атак, зачастую реализуются дополнительными техническими мерами защиты. Применение той или иной меры определяется экспертно, а принятие решения о реализации лежит на плечах собственников предприятия. В то же время, во всем мире признается актуальность и практическая значимость проблемы противодействия атакам, констатируются участившиеся

случаи несанкционированного доступа со стороны к системам управления и оборудованию, которые могут привести к серьезным последствиям.

Методы обеспечения надежности через ИБ АСУ промышленными объектами плодотворно разрабатывались Африным А.Г., Беловым Е.Б., Васильевым В.И., Воробьевым А.А., Галимовым Р.Р., Герасименко В.А, Гузаировым М.Б., Дерябиным А.В., Лившицем И.О., Yoana A. I, Gunikhan S., Kuppusamy K.S, Gastellier-Prevost S., Granadillo G.G., Laurent M. Развитию теоретических положений ИБ АСУ промышленными объектами посвящены исследования Баранкова И.И., Глушкова В.М., Гуляева С.А., Котельникова В.А., Кузякова О.Н., Лебедева Ю.В., Машкина М.В., Михайлова У.В., Остапенко А.Г., Петрова Б.Н, Харкевича А.А., Чжо З.Е., Tewari A., Jain A.K., Gupta B.B., Fette I., Sadeh N., Tomasic A. Однако отдельные аспекты проблемы в работах перечисленных ученых не могут быть непосредственно применены в условиях крупного химико-технологического предприятия, обладающего рядом особенностей, перечисленных в диссертации, или же устарели в связи с быстрым развитием угроз ИБ. Поэтому модификация и дальнейшее развитие методов обеспечения надежности через ИБ АСУ промышленными объектами на этапах внедрения и эксплуатации представляются актуальными и практически значимыми.

В частности, обычные меры защиты против хорошо известных угроз в информационных системах (фишинга, фильтрации контента, определения необычной пользовательской активности и т.п.), подробно рассмотренные в работах Абрамова Е.С., Болодуриной И.П., Веревкина А.П., Гайдара М.Б., Дика Д.И., Жигулина Г.П., Каримова М.М., Квятковской И.Ю., Куликова Г.Г., Монахова М.Ю., Нестерука Ф.Г., Оголюка А.А., Поршнева С.В., Сердюка В.А., Сыпина А.А., Тимониной Е.Е., Ушакова Д.В., Хафизова А.Ф., Чопорова О.Н., Шелухина О.И., Cao Y., Han W., Le Y., Kang, J., Lee, D., Zhang Y., Hong J., Cranor L., Pan Y., Ding X. в АСУ промышленными объектами имеют свои особенности применения и даже оценки эффективности.

Целью диссертационного исследования является повышение надежности АСУ промышленными объектами крупного химико-технологического предприятия за счет совершенствования системы информационной безопасности.

Объект исследования – АСУ промышленными объектами химико-технологического предприятия на примере ПАО «Уралкалий» г. Березники Пермского края. **Предмет исследования** – методы и алгоритмы противодействия несанкционированным действиям пользователей и внешних злоумышленников.

Задачи исследования:

1. Изучение и анализ особенностей архитектуры АСУ промышленными объектами предприятий Пермского края, на предмет проблем и рисков ИБ;
2. Модификация существующих алгоритмов фильтрации трафика для применения в АСУ промышленными объектами;
3. Разработка метода алгоритмических проверок интернет-адресов с учетом особенностей их использования в АСУ промышленными объектами;
4. Разработка модели оценки опасности ресурсов, на основе различных индикаторов фишинговости;
5. Разработка методики повышения надежности АСУ промышленными объектами за счет обеспечения информационной безопасности;
6. Программная реализация полученных научных продуктов в виде системы повышения надежности АСУ промышленными объектами, а также для модельных исследований;
7. Провести имитационные исследования компонентов системы повышения надежности через ИБ АСУ промышленными объектами.

Методы исследования основаны на использовании теории автоматического управления, системного анализа, математического и имитационного моделирования, теории алгоритмов.

Научная новизна заключается в следующем:

1. Предложены к использованию в качестве первичных фильтров модифицированные алгоритмы фильтрации трафика в АСУ

промышленными объектами, позволяющие снизить риски несанкционированного изменения их информационной базы, повысить показатели производительности и точности обнаружения форм авторизации, отличающиеся применением механизма ротации записей в списке с учетом сохранения повторяемости посещения адресов, удалённым хранением списков и их шифрованием; применением n-граммного метода для поиска ключевых слов используемых для поиска форм авторизации.

2. Предложен метод алгоритмических проверок адресов в АСУ промышленными объектами на наличие фишинговых свойств, отличающийся комплексом обнаруживаемых свойств, а именно поиском использования более одного домена верхнего уровня в адресе; получением оценки ранжирования через несколько сервисов оценки; подсчетом частоты смены DNS-записей; сопоставлением контента ресурса с его доменным именем; поиском спец. символов в адресе с учетом их повторяемости; оценкой доступности ресурса основанной на его ошибках.
3. Разработана модель оценки опасности внешних ресурсов, посещаемых пользователями АСУ промышленными объектами, основанная на методе опорных векторов, отличающаяся минимизацией ошибок оценки за счет использования радиальной базисной функции в качестве функции ядра.

Достоверность результатов диссертационной работы обеспечена корректным использованием математического аппарата, результатами моделирования и тестирования алгоритмов и программного обеспечения.

Теоретическая значимость работы состоит в развитии теории обеспечения надежности АСУ промышленными объектами путем улучшения отдельных направлений информационной безопасности.

Практическая значимость работы заключается в разработке методики повышения надежности через ИБ АСУ промышленными объектами и её реализации в виде комплекса программного обеспечения для фильтрации трафика,

определения опасности внешних ресурсов, проверки адресов. Программный продукт реализован средствами объектно-ориентированного программирования. Внедрение разработанного программного продукта выполнено в ПАО «Уралкалий» и ЗАО «Бионт», а также в учебном процессе в Березниковском филиале ФГБОУ ВО Пермский национальный исследовательский университет, что подтверждается актами о внедрении.

Основные положения, выносимые на защиту:

1. Модифицированные алгоритмы фильтрации трафика, основанные на персональных белых списках и поиске форм авторизации;

2. Метод алгоритмических проверок адресов, состоящем из шестнадцати алгоритмов;

3. Модель оценки опасности внешних ресурсов, основанной на методе опорных векторов с радиальной базисной функцией в качестве функции ядра.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на Всероссийской научно-технической конференции "Автоматизированные системы управления и информационные технологии" (Пермь, 2018), Международной конференции студентов и молодых ученых "Молодежная наука в развитии регионов" (Березники, 2016), Всероссийской научно-практической конференции студентов и молодых ученых "Молодежная наука в развитии регионов" (Березники, 2017, 2018, 2019, 2020,2021), Всероссийской научно-практической конференции «Решение» (Березники, 2015, 2016, 2018, 2019, 2020), Международном семинаре «Advanced Technologies in Material Science, Mechanical and Automation Engineering» в рамках 24-й международной научной открытой конференции «Современные проблемы информатизации» (Красноярск, 2019).

Личный вклад автора. Содержание диссертации и основные положения, выносимые на защиту, отражают персональный вклад автора в постановку задачи, в разработку теоретических методов, в модельное исследование и внедрение результатов работу.

Публикации. По теме исследования опубликовано 19 работ, в том числе 1 статья в журнале Scopus, 5 статей в изданиях, рекомендованных ВАК РФ, 13 работ в материалах международных и всероссийских конференций.

Соответствие паспорту специальности. 9-Методы эффективной организации и ведения специализированного информационного и программного обеспечения АСУТП, АСУП, АСТПП и др., включая базы и банки данных и методы их оптимизации, 12-Методы контроля, обеспечения достоверности, защиты и резервирования информационного и программного обеспечения АСУТП, АСУП, АСТПП и др., 13-Теоретические основы и прикладные методы анализа и повышения эффективности, надежности и живучести АСУ на этапах их разработки, внедрения и эксплуатации, паспорта научной специальности 05.13.06 - Автоматизация и управление технологическими процессами и производствами (промышленность).

Структура и объем диссертации. Работа состоит из введения, четырех глав, заключения и списка литературы. Основное содержание диссертации изложено на 211 страницах, в т.ч. 21 таблица, 54 рисунка. Список литературы содержит 123 источника.

Во введении обоснованы актуальность темы диссертационного исследования, формулируются цель и задачи, определяется новизна и практическая значимость.

В **первой главе** выполнен анализ задачи повышения надежности АСУ промышленными объектами через использование новых методов защиты и диагностирования их информационного и программного обеспечения и проведен обзор предыдущих исследований в этом направлении. С учетом особенностей классической трехуровневой архитектуры АСУ промышленными объектами и статистики успешно реализованных атак, в качестве основной угрозы для исследования определен фишинг. Рассмотрены и обоснованно выбраны для целей настоящего исследования конкретные метрики оценки методик и алгоритмов повышения надежности через ИБ АСУ промышленными объектами.

Вторая глава посвящена развитию компонентов методики повышения надежности АСУ промышленными объектами с использованием комбинированного подхода, основанного на эвристических методах защиты и диагностирования их информационного и программного обеспечения. Представлены основные компоненты методики: модифицированные алгоритмы фильтрации: алгоритм, основанный на «белом листе» и алгоритм, основанный на поиске форм авторизации; метод алгоритмических проверок, состоящий из 16 алгоритмов; модель оценки опасности ресурсов, посещаемых пользователями АСУ промышленными объектами, основанная на методе опорных векторов. Для каждого из алгоритмов приведены соответствующая им блок-схема и описание.

В **третьей главе** представлено описание методики повышения надёжности АСУ промышленными объектами через ИБ. С целью её раскрытия разработаны трехкомпонентная архитектура и программная реализация системы повышения надежности АСУ промышленными объектами.

В **четвертой главе** произведено исследование эффективности использования разработанных научных продуктов для повышения надежности АСУ промышленными объектами. Для анализа эффективности системы в целом и отдельных алгоритмов проведены имитационные исследования. Так же представлены статистические данные по результатам опытной эксплуатации на действующей АСУ промышленными объектами ПАО «Уралкалий».

В заключении сформулированы основные результаты научно-квалификационной работы.

В приложениях представлены дополнительные материалы и копии актов об использовании результатов работы.

Глава 1. Системный анализ задачи повышения надежности через ИБ автоматизированных систем управления промышленными объектами

1.1 Место, роль и постановка задачи повышения надежности через ИБ автоматизированных систем управления промышленными объектами

С 2000 года начался значительный рост атак, направленных на проникновение в промышленные сети. С появлением необходимости подключения промышленных сетей к сети Интернет и концепции промышленного Интернета вещей, проведение атак на АСУ промышленными объектами стало возможным из любой точки мира. С дальнейшим развитием информационных технологий атаки стали принимать различные формы. Защититься от малоизученных, совершенно новых атак практически невозможно, в то время как защита от остальных атак - вопрос обеспечения эффективной инфраструктуры и физической безопасности в целом [11,12,35,59,59,83]. Понимание особенностей [4] и возможных уязвимостей инфраструктуры, а также векторов атак и методов защиты от них имеют важное значение. Это крайне необходимо для повышения надежности АСУ промышленными объектами [1,33,34], в совокупности со смягчением потенциального ущерба, который может быть нанесен в результате проникновения в промышленную сеть нарушителем [10].

Наибольший интерес к уровню защищенности АСУ промышленными объектами появился после случаев, возникших с червями Stuxnet, Duqu, Flame, при помощи которых, злоумышленники атаковали ряд государственных учреждений, промышленных объектов, в том числе ядерных, различных стран. На смену этим червям пришли более сложные, активно ретирующиеся и многоступенчатые атаки. Так, для распространения вируса Havex в 2014 году злоумышленники взламывали сайты производителей ПО для управления промышленными предприятиями и заражали официальные дистрибутивы Supervisory Control And Data Acquisition

(SCADA)-систем, которые затем устанавливались на предприятиях, что позволило нарушителям получить контроль над АСУ промышленными объектами в нескольких европейских странах. Так же, нужно отметить, что любой длительный отказ в крупной промышленной организации, вне зависимости от причины его появления, непосредственно ведет к огромным финансовым потерям. Вышеописанные факторы являются лишь некоторыми причинами появления [39,67] приказа ФСТЭК России от 14 марта 2014 г. N 31 [70], с которого началось активное регулирование требований к системам защиты АСУ КИИ.

Для большинства АСУ промышленными объектами свойственны уязвимости, характерные и для всех остальных информационных систем. Следующие основные тенденции были выявлены в процессе работ по исследованию ИБ АСУ промышленными объектами[38,107,108]:

- Открытость. Множество АСУ промышленными объектами в РФ и других странах имеют доступ в сеть Интернет. Подобные системы можно найти при помощи поисковых машин с использованием специальных ключевых слов. В том числе и отдельные компоненты АСУ промышленными объектами могут быть доступны из сети Интернет, при этом их владельцы могут не подозревать об этом;
- Один ключ для всех замков. Продолжается монополизация производства контроллеров, SCADA-систем управляющих производством в различных отраслях. В данном случае продукт одного и того же вендора может быть использован на большом количестве предприятий. Следовательно, уязвимость, найденная в данных продуктах, будет применима абсолютно ко всем, кто их использует;
- Угрозы всегда на шаг впереди. Особенности АСУ промышленными объектами, сложность их организации и необходимость использования в непрерывном производстве ведут к тому, что основные элементы АСУ промышленными объектами отживают свой век, при этом работает принцип – «работает не трогай». Т.е. о своевременном обновлении программной или аппаратной части речь не идет, следовательно, риски использования

злоумышленниками ранее известных уязвимостей АСУ промышленными объектами увеличиваются в значительной степени.

- «Mad house». Дешевизна и малые габариты промышленных устройств привели к активному внедрению в обыденную жизнь людей устройств управления системами жизнеобеспечения зданий/сооружений, мониторингом и распределением электроэнергии, видеонаблюдением. При этом внимание к ИБ этих устройств на уровне конечных пользователей оставляет желать лучшего.

Исходя из выделенных основных тенденций, можно обозначить основные типы рисков ИБ, возникающих в отношении к АСУ промышленными объектами [24,39,42,54,66,108]:

1. Риски получения злоумышленниками несанкционированного доступа (НСД) к узлам промышленных и корпоративных сетей путём целевых фишинговых атак, направленных на пользователей АСУ промышленными объектами.
2. Риски получения злоумышленниками НСД к узлам и данным из-за "слабой" парольной защиты. Использование словарных идентификаторов и паролей является одной из самых распространенных уязвимостей, она была обнаружена на сетевом периметре в 87% исследованных систем, причем в 67% компаний простые пароли использовались и для привилегированных учетных записей. В каждой второй организации (53% от общего числа) для доступа к публичным WEB-приложениям используются словарные учетные данные.
3. Риски применения недостаточно криптостойкого шифрования посредством старых (слабых) криптографических алгоритмов и систем управления ключами.
4. Риски, вызванные ошибками конфигурации (ошибки настройки сетевого оборудования и служб ОС, ошибки при разграничении прав доступа и полномочий на доступ к ресурсам, применение заводских (по умолчанию) шаблонов безопасности, упрощающих управление системой) [79].

5. Риски, возникающие вследствие отсутствия обновлений безопасности для платформенных ОС и ПО, либо несвоевременная их установка. При этом в АСУ промышленными объектами недопустимо автоматическое обновление, т.к. все обновления производятся только при плановых остановочных работах, исключительно в "ручном" режиме.
6. Риски, связанные с эксплуатацией уязвимостей программно-аппаратных компонент АСУ промышленными объектами (позволяющие использовать Denial of Service (DoS)-атаки для автоматического выполнения аварийных протоколов и пр.).
7. Риски эксплуатации уязвимостей мобильных клиентов на базе Android/iOS-приложений, взаимодействие мобильных устройств с инфраструктурой, включая решения Programmable Logic Controller (PLC), Open Platform Communications (OPC), Manufacturing Execution System (MES) для управления SCADA-системами - незащищенные или недостаточно защищенные методы передачи и хранения данных (в том числе, некорректное использование Secure Sockets Layer (SSL) или «самодельные» криптоалгоритмы), удаленная атака типа «отказ в обслуживании» на клиент и сервер, SQL-инъекции, использование не доверенных входных данных в качестве параметров настройки техпроцесса и др.
8. Риски недостаточного разделения между сегментами промышленных и корпоративных сетей. В данном случае, большое количество точек "входа", обусловлено особенностями архитектуры построения АСУ промышленными объектами сетей, а также тем, что в коммутационном шкафу может находиться оборудование, к которому, при обслуживании, персонал может иметь неограниченный доступ (в частности, к периметровым).
9. Высокие риски вирусных заражений (отсутствие антивирусной защиты на любом из узлов сети).
10. Риски, связанные с отсутствием мер по обеспечению физической безопасности на объектах. Отсутствие блокировки автоматизированных рабочих мест (АРМ) пользователей (полная физическая блокировка АРМ

посредством железного ящика или отдельного помещения). Отсутствие Internet Protocol (IP)-камер и систем контроля и управления доступом (СКУД) на периметре объектов АСУ промышленными объектами и др.

Сегодня, увеличение количества уязвимостей, наблюдаемых в процессе развития АСУ промышленными объектами, значительно увеличивает количество возможных рисков и вероятность их реализации. Реализация данных рисков в случаях с АСУ промышленными объектами, может привести, как к материальным потерям, в случае поломки оборудования в результате изменения конфигурации злоумышленниками, так и к смерти сотрудников, в случае отказа систем жизнеобеспечения или иных систем злоумышленниками. Исходя из этого, список рисков будет постоянно пополняться, пока не иссякнет интерес злоумышленников к АСУ промышленными объектами. Все вышеописанное в значительной степени увеличивает актуальность задачи повышения надежности АСУ промышленными объектами.

Исходя из тенденций АСУ промышленными объектами и подключения их компонент к сети Интернет, активно развивается направление целевых фишинговых атак на пользователей АСУ промышленными объектами. Фишинг — это одно из направлений кибер-преступлений, которое активно развивается. Фишинг реализуется через фальсификации сайтов, форм авторизации, электронных писем и пр., в которых с использованием методов и средств социальной инженерии имитируется легитимный аналог, чтобы обмануть жертву и получить её конфиденциальную информацию с целью собственного использования. Такие как, номер кредитной карты, пароль от персональной учетной записи или учетной записи, используемой в АСУ промышленными объектами и др. Учитывая вышеописанное, фишинг выбран в качестве основной угрозы к исследованию.

Подтверждение активного развития направления фишинговых атак можно увидеть в большом количестве возникших кибер-инцидентов за последние годы. Например, в одном из недавних кибер-инцидентов, преступники сфабриковали фишинговые сайты 26 индийских банков, стремясь получить конфиденциальную

клиентскую информацию [92]. Злоумышленники воспользовались методами социальной инженерии для реализовали фиктивных писем. В содержимом писем присутствовала ссылка, открыв которую получатель перенаправлялся на фишинговый сайт. Для конечного пользователя содержимое письма и отправитель выглядели, вполне, правдоподобно. Международная коалиция APWG опубликовала отчет о количестве уникальных фишинговых сайтов согласно рисунка 1.1, зарегистрированных за период с октября 2017 года по март 2018 года, согласно которому наблюдался экспоненциальный рост количества фишинговых сайтов [50].

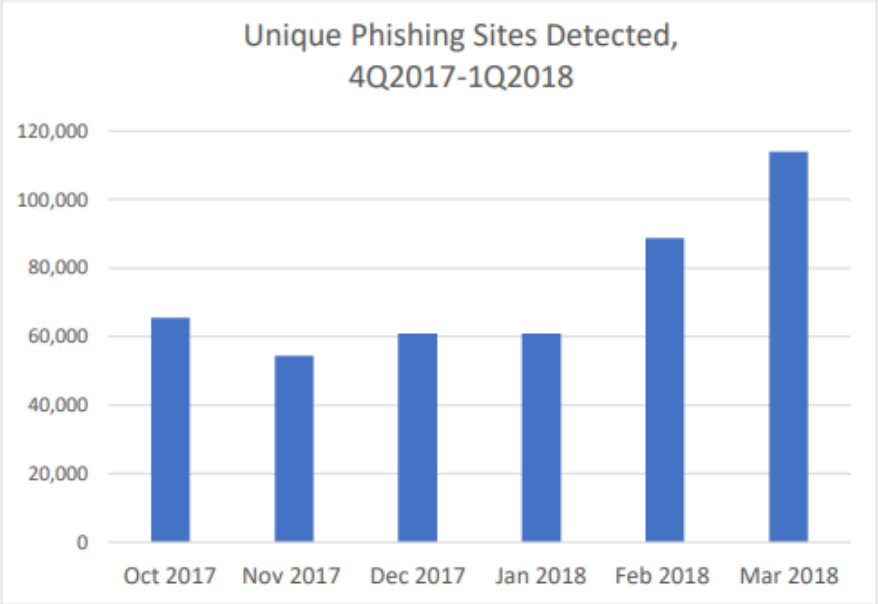


Рисунок 1.1 Отчет о количестве уникальных фишинговых сайтов.

Также наблюдается количества уязвимостей в компонентах АСУ промышленными объектами [47] согласно рисунка 1.2. Уязвимостям наиболее подтверждены SCADA/HMI-интерфейсы/Распределенные системы управления, следом за ними промышленное сетевое оборудование, далее ПО, PLC/терминалы удалённого доступа и др. В свою очередь, статистика Kaspersky Lab ICS CERT с октября 2017 по июнь 2018 показывает, что таргетированные фишинговые письма с вредоносным вложением активно рассылались и продолжают рассылаться

крупным промышленным компаниям[63,88]. Экспертами лаборатории определено, что в России успешно подверглись, одной из фишинговых атак, не менее 400 различных промышленных компаний. Промышленный сектор становится приоритетной целью для мошенников [63,88]. Атакуя промышленную компанию в целом, злоумышленники атакуют, в том числе, пользователей АСУ промышленными объектами, которые могут иметь различные роли: от оператора, до обслуживающего персонала АСУ промышленными объектами, в рамках отдельных объектов или всего предприятия.

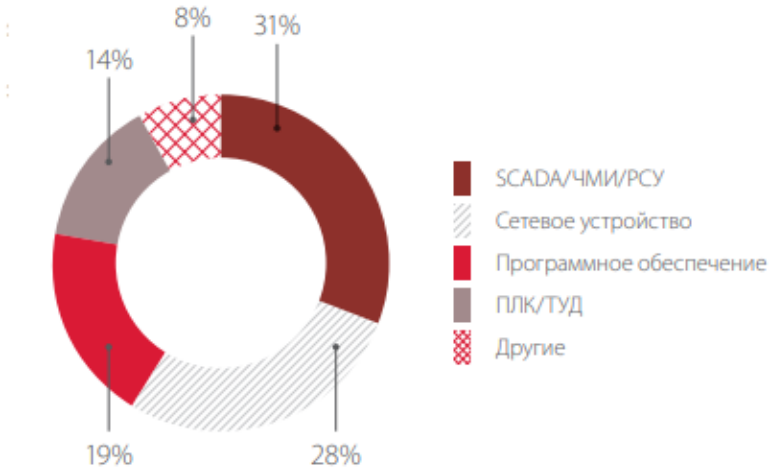


Рисунок 1.2 Основные цели промышленных секторов.

АСУ промышленными объектами уязвимы для фишинга по ряду причин:

- АСУ промышленными объектами не достигли уровня, при котором можно обойтись без человеческого вмешательства, а люди – серьёзная уязвимость;
- сложность применения организационных или технических средств защиты, субоптимальный их выбор;
- преуменьшение общего уровня угрозы или недостаточное её понимание;
- обеспечение непрерывности производства – основная задача обслуживающего персонала АСУ промышленными объектами, ИБ для них менее важна;

- отсутствие своевременных обновлений как программного и аппаратного обеспечения АСУ промышленными объектами, в том числе платформенных операционных систем (ОС); как следствие высокий риск заражений зловредами или использование уязвимостей злоумышленниками;
- наличие мобильных клиентов, WEB-интерфейсов и др.;
- разработано множество различных решений и методов борьбы с фишингом [52], но не существует решения, которое может гарантировать полную защиту АСУ промышленными объектами [7,8,41].

Принимая во внимание, что объектом исследования является АСУ промышленными объектами, как объект управления ее можно охарактеризовать набором входных и выходных переменных, управляющих и возмущающих воздействий (Рис. 1.3).

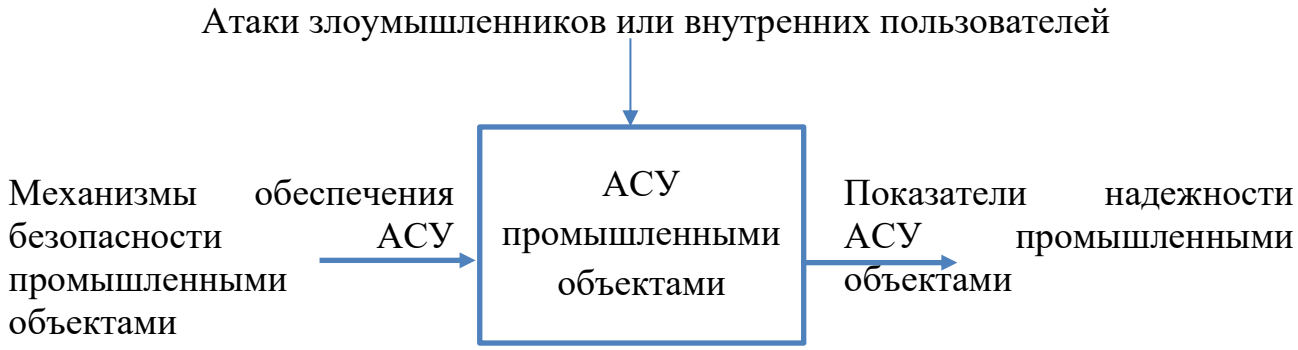


Рисунок 1.3 Объект управления

Управляющие воздействия: алгоритмы, методы, модели обеспечения безопасности АСУ промышленными объектами.

Возмущающие воздействия: атаки злоумышленников или внутренних пользователей на АСУ промышленными объектами.

Выходные величины: средняя наработка на отказ, среднее время восстановления, коэффициент готовности АСУ промышленными объектами.

Задача управления формулируется следующим образом: с учетом воздействия возмущений найти управляющие воздействия, при которых улучшаются показатели надежности.

Таким образом, одна из ролей задачи повышения надежности АСУ промышленными объектами - минимизация рисков ИБ АСУ промышленными объектами, возникающих на этапах их внедрения и эксплуатации.

Нужно отметить, что обеспечение должного уровня защищенности [32] компонентов АСУ промышленными объектами – одна из основных задач в РФ, это подтверждает в том числе и Федеральный закон от 12 июля 2017 года № 187-ФЗ [78]. А также, пристальное внимание стран, направленное на усиление защищенности АСУ промышленными объектами, говорит о появлении кибер-инцидентов в данной области. Что свидетельствует о значительном интересе злоумышленников к АСУ промышленными объектами. Но разрабатываемые странами меры противодействия, не успевают за прогрессом злоумышленников. А реализация предлагаемых организационных и технических мер, не может в полной степени минимизировать возникающие при этом риски и гарантировать максимальный уровень защищенности в отношении современных угроз.

1.2 Особенности архитектуры компонент автоматизированных систем управления промышленными объектами

На сегодняшний день вследствие увеличения значимости информационных активов в современном мире в целом, компоненты АСУ промышленными объектами всё чаще начинают занимать существенные позиции в инфраструктурах различных объектов РФ. Очень важно не оставлять их систему ИБ без внимания.

Рассмотрим типичную архитектуру распределённой АСУ ТП, как компоненты АСУ промышленными объектами в соответствии с рисунком 1.4:

- Нижний уровень ввода/вывода (полевой) производит сбор информации о технологическом процессе, состояниях оборудования, осуществляет исполнительные воздействия на процесс, определенные регулятором. К основному оборудованию этого уровня относят кабельные линии связи,

контрольно-измерительные приборы, исполнительные элементы, модульные станции ввода-вывода, силовые реле, конечные выключатели, частотные преобразователи и другие средства автоматизации [51].

- Средний уровень автоматического управления решает задачи останова и пуска оборудования, логико-командного управления, автоматического регулирования и управления [26], автоматической работы защиты при авариях и сбоях оборудования. Зачастую все это организуется на базе PLC, с соответствующим количеством модулей ввода/вывода, барьеров искрозащиты и специального коммуникационного оборудования. PLC осуществляют обмен данными между всеми компонентами системы при помощи различных протоколов (семейств PROFIBUS, Modbus и др.) [51].
- Верхний, операторский, уровень решает задачи динамического планирования или диспетчеризации процессов, оптимизации режимов, расчета технико-экономических показателей, записи трендов и протоколов событий, визуализации и архивирования данных процесса, диагностики и коррекции программного обеспечения (ПО) системы. Уровень включает в себя SCADA/HMI системы. Основным оборудованием данного уровня чаще всего являются средства визуализации, это система компьютеров и специализированных мониторов. Количество оборудования и информационных параметров, а также частота их изменения программируются в зависимости от потребностей производства.

Верхний "операторский" уровень архитектуры.

SCADA/HMI-системы состоят из ПО и оборудования, реализующего информационные (зачастую оповещательные) функции. На этот уровень поступает информация не только о текущем состоянии производственных процессов, но и оповещения (Alarm), возникающие в результате внешних вмешательств в рабочий процесс, и срабатывания автоматики безопасности. Так же ПО, эксплуатируемое на данном уровне, позволяет удалённо управлять оборудованием и конфигурировать систему управления.

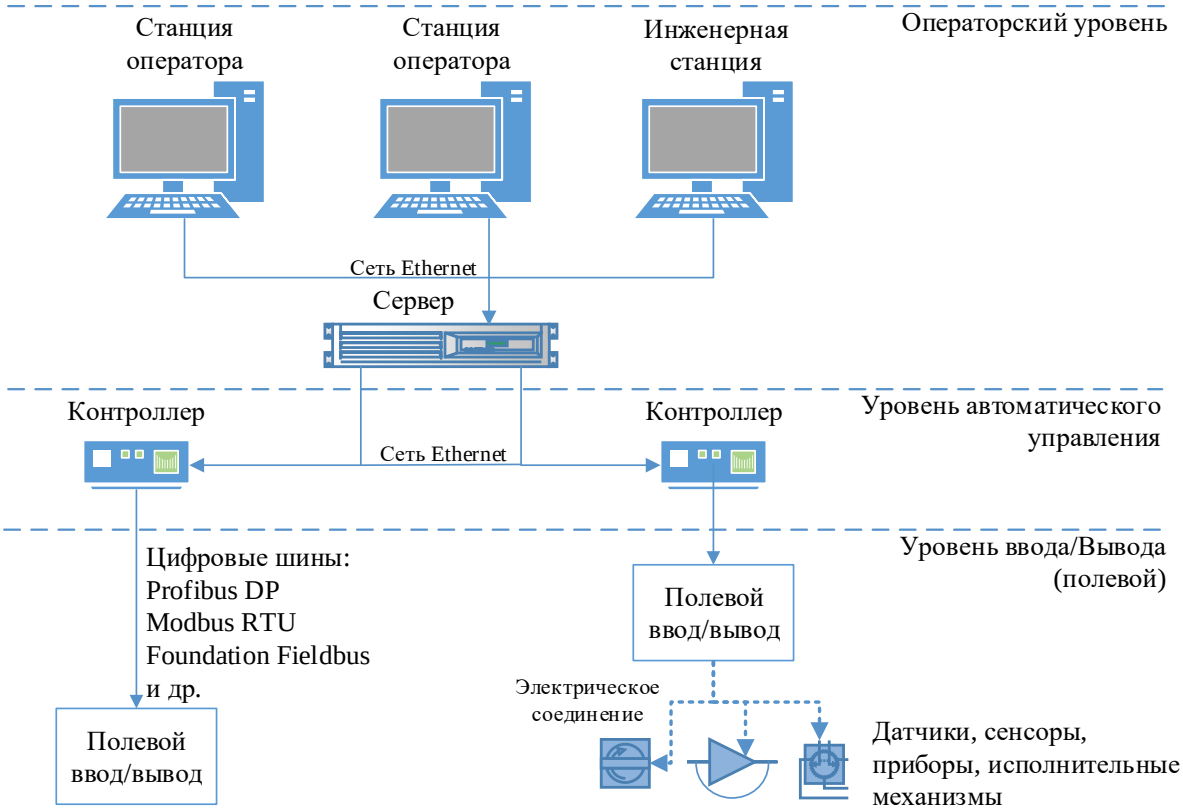


Рисунок 1.4 Пример схемы, распределенной АСУ ТП

Так как человек — это потребитель информации и субъект её обработки, однозначно существуют риски, связанные с ошибками принятия решений. Например, некорректная настройка исполнительных механизмов, ПО и др. В целом, для любого человека свойственны ошибки и возможности любого могут быть ограничены при принятии тех или иных решений. Такие явления, которые возникают при любых систем (в том числе, систем обработки информации) и людей часто называют «человеческий фактор». Ошибки, связанные с проявлением человеческого фактора, как правило, непреднамеренны: человек совершая ошибку, может считать свои действия правильными или подходящими к данной ситуации.

Ошибочные действия человека, можно сгруппировать по причинам их проявления [26]:

- отсутствие достаточного количества информационных материалов информационного (соответствующие специальные документы, материалы, инструкции для повышения осведомлённости персонала). В большей степени

такие ошибки возникают при недостаточном количестве времени для принятия решения;

- отсутствие достаточного количества требуемых ресурсов для реализации принятого решения;
- просчет при оценке рисков, преуменьшение влияния человеческого фактора на процессы;
- воздействие сторонних/внешних факторов (проблемы отвлечения внимания);
- физическое состояние человека (регулярный стресс и подавленная реакция на проблемы, на фоне длительной монотонной работы; эмоциональная усталость; неготовность выполнять физические и умственные нагрузки различного характера).

Угрозы физического характера, происходят непосредственно от действий нарушителя (человека), форс-мажора (стихийные бедствия) или от отказа оборудования и внутренних систем жизнеобеспечения. Применение запрета физического доступа к оборудованию и использование видеонаблюдения, это основные меры применимые для предотвращения физических угроз нарушителей.

Уязвимости ОС, под управлением которых находится SCADA-система, являются одной из ключевых проблем ИБ АСУ промышленными объектами верхнего уровня. Т.к. вместе с “условной” стабильностью работы, которую обеспечивает управляющая система, владелец системы получает все возможные уязвимости, типичные для этой ОС. Если вендор предоставляет свое ПО на собственной ОС, это значительно сокращает риски, возникающие при использовании данного продукта. Но, к сожалению, большинство компаний, используют в качестве управляющей ОС сторонние системы, зачастую предоставляемые компанией Microsoft. В таблице 1.1 приведены примеры систем, которые используются в производстве предприятиями Пермского края:

Таблица 1.1 SCADA-системы и их платформенные ОС

Операционная система	**SCADA-система**
Microsoft Windows	Experion PKS
	Experion TPS
	Emerson Delta-V
	Siemens SIMATIC
	HoneyWell
	Intouch
	YOKOGAWA CENTUM
	ABBSystem 800xA
Linux/Unix	Foxboro (несмотря на плюсы Unix, в большинстве случаев, для работы с данной системой используют виртуальную машину с эмуляцией Microsoft Windows)

До определённого момента специалистам, обслуживающим АСУ промышленными объектами, было не критично, под управлением какой ОС работает SCADA-система. Специалисты не задавались вопросом обновлений и версионности платформенной ОС. Главным критерием выбора ОС был большой аптайм машин, т.е. чем дольше ОС работала без перезагрузки вместе со SCADA-системой, тем больше была вероятность её выбора, немаловажным фактором являлась поддержка того или иного семейства ОС, самим производителем SCADA. На сегодняшний день, ввиду пристального внимания злоумышленников к АСУ промышленными объектами, стали формироваться критерии выбора не только SCADA-системы, но и платформенной ОС. С целью сравнения характеристик ОС, представленных в таблице 1.2, ниже представлены их архитектуры и возможные уязвимости при их использовании.

Итак, основным принципом работы ядра ОС семейства Microsoft (MS) Windows является принцип подсистем окружения. Приложения, запускаемые в системе, не взаимодействуют с сервисами ОС открыто. Вся работа происходит через подсистему окружения API, вызова функций ядра в открытом виде не происходит. За счет этой особенности ОС имеется возможным оставлять обратную

совместимость с приложениями, разработанными под ранее выпущенные версии ОС в соответствии с рисунком 1.5.

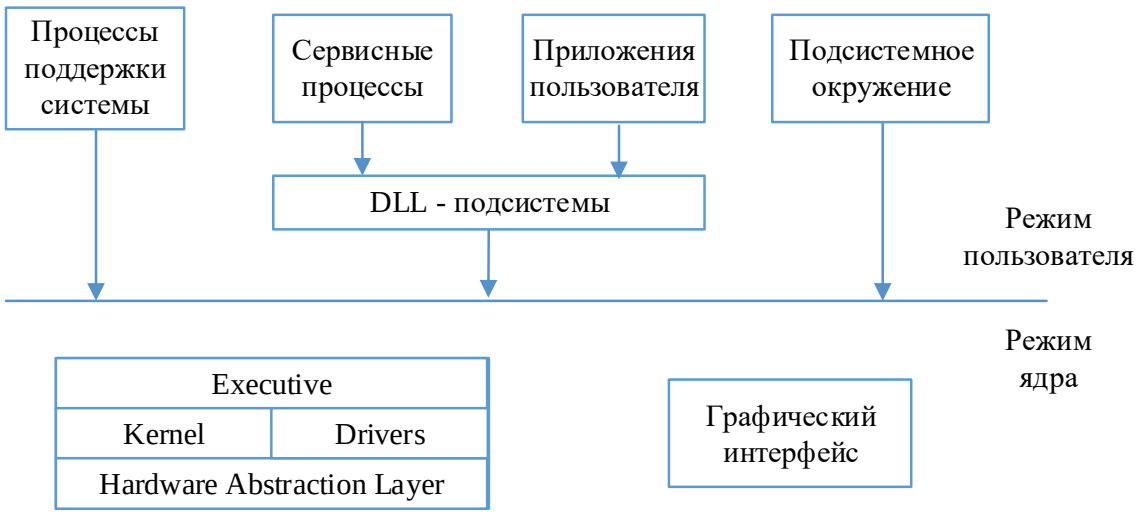


Рисунок 1.5 Схема архитектуры ядра Windows

При проектировании ядра сделан акцент на существующее и будущее аппаратное обеспечение (т.к. сложно спрогнозировать, какая аппаратная платформа будет использоваться через год, два, три), с возможностью смены типов платформ и т.д. Для решения данной проблемы, было принято решение добавить в архитектуру портабельное ядро с помощью слоя абстракции аппаратного обеспечения Hardware Abstraction Layer (HAL). Над этим слоем располагаются следующие два: Drivers, взаимодействуют с модулями при помощи HAL и дают возможность воспользоваться своими сервисами (device input/output); Kernel, основная часть ядра.

Затем слой Executive реализует основные сервисы ОС: управление памятью, процессами, потоками, безопасность, ввод/вывод, межпроцессорное взаимодействие. Остальные модули ядра обеспечивают работу графического интерфейса и оконную подсистему [86]. Для того чтобы не перезаписывать всё ядро системы при добавлении ресурсов в систему, используется унифицированная форма: наименование, совместное использование, учет. Любой раздельный ресурс системы — это объект. При создании объекта создается ссылка на объект (handle).

Эти ссылки ассоциированы с процессом, но могут быть отданы другим процессам. Ресурсы, используемые разными процессами, называют разделяемыми. Есть 2 класса объектов: executive, взаимодействуют с пользовательскими исполняемыми файлами и компонентами подсистемы; kernel, предоставляют базовые ресурсы (физические устройства) [86].

Режим пользователя состоит из подсистем, передающих запросы ввода-вывода советующему драйверу режима ядра посредством менеджера ввода-вывода. На уровне пользователя есть: подсистема окружения (запускает приложения, написанные для разных ОС) и интегрированная подсистема (управляет особыми системными функциями от имени подсистемы окружения). Режим ядра имеет полный доступ к аппаратной части и системным ресурсам компьютера. Он также предотвращает доступ к критическим зонам системы со стороны пользовательских служб и приложений [22].

Как было отмечено ранее, одна из основных проблем выбора ОС как платформы для SCADA-систем — это уязвимости, с которыми придется иметь дело в дальнейшем при работе со SCADA-системой. Возможные уязвимости:

- При выборе ОС семейства MS Windows нужно осознавать важность разграничения прав. Разграничение прав не сводится к защите от заражений и блокированием доступа к правам администраторским учетным записям. Семейство Windows очень уязвимо перед вирусами и червями — отчасти потому, что выполняет за пользователя многие задачи. В результате вредоносный код запускается автоматически при выполнении совершенно не связанных с ним задач. Например, при попытке открыть файл MS Word, являющийся, на самом деле, виртуозно замаскированной вредоносной программой, Windows с готовностью передает файл от Word другому процессу, необходимому для успешного запуска этой программы [22].
- Переполнение буфера. Если ранее выделенный размер буфера превышен или его недостаточно для помещения внутрь всех данных, то содержимое памяти, находящееся за буфером, будет замещено. В зависимости от ситуации за концом буфера могут находиться:

1. другой буфер и переменные программ;
2. служебные данные (адреса возврата функций);
3. исполняемый код;
4. незанятая/отсутствующая страница памяти.

- Рассмотренное семейство ОС потенциально допускает возможность переполнения буфера почти во всех вышеописанных ситуациях, кроме исполняемого кода [22].
- Ранее доступ к адресному пространству по умолчанию был разрешен всем пользователям, даже гостю (Guest), и если один из владельцев процесса не хочет, чтобы в его пространство проникали другие, то должен об этом явно заявить. В более современных ОС доступ получает группа администраторов и “debug users”. Имеется возможность отладки процессов, находящихся в работе, и новых процессов с наследованием привилегий процесса отладчика.

Рассмотрим архитектуру ОС семейства Linux/Unix в соответствии с рисунком 1.6, ее можно разделить на 5 основных компонентов:

- Ядро – виртуализирует общие аппаратные ресурсы сервера, с целью предоставления каждому процессу индивидуальных ресурсов. Что позволяет процессам работать изолированно друг от друга. Так же, управляет самими процессами и драйверами устройств. Все данные ядра хранятся в общем адресном пространстве, применение которого минимизирует переключения при запросах виртуальных ресурсов процессами. При необходимости расширения функционала ядро позволяет загружать, либо выгружать модули в память. Каждый модуль представляется в виде программного кода, который расширяет стандартные возможности ядра. Модульность позволяет гибко использовать функционал ОС, используя необходимые модули в соответствии с потребностями, минимизируя использование системных ресурсов. Большинство системных процессов работают в режиме ядра и изолированы от пользовательского пространства, но часть из них так же включена в раздел системных библиотек. Ядра бывают различных видов (Monolithic, Hybrid, Exo, Micro).

- Разделяемые системные библиотеки – стандартный набор функций, используемых для реализации функциональных возможностей ОС. Основная задача обеспечение взаимодействия между ядром и пользовательскими процессами;
- Оболочка – интерфейс взаимодействия с ядром, позволяющий отправлять команды ядру от пользователя;
- Аппаратный уровень – периферийные устройства, такие как оперативная память, жесткие диски, центральный процессор и др.;
- Системные утилиты – утилиты предоставляющие функции операционной системы пользователю.

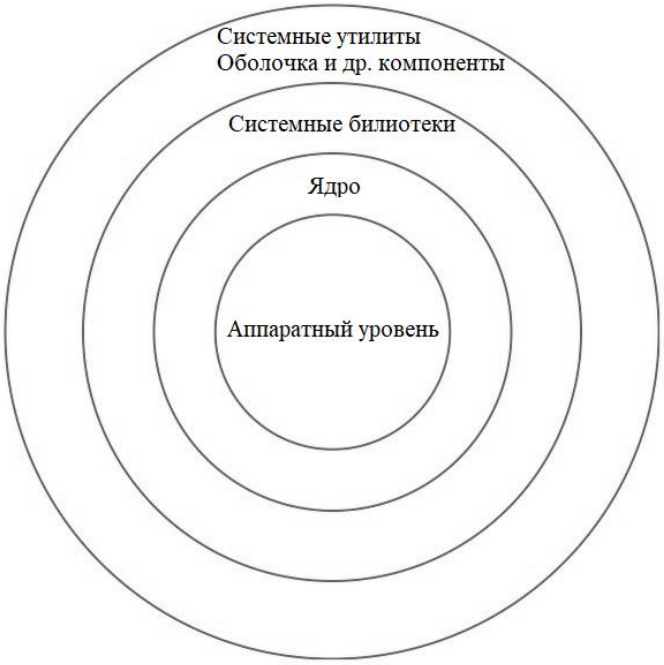


Рисунок 1.6 Компоненты ядра Linux/Unix

Для большей наглядности составлена таблица 1.2 сравнения основных характеристик описанных выше семейств ОС, относящихся к ИБ.

Таблица 1.2 Сравнение характеристик

Характеристика	**Windows**	**Unix**
Доступность исходных текстов	Недоступны	Доступны

Характеристика	Windows	Unix
Сложность анализа кода (при наличии)	Высокая	Высокая
Распространенность	Высокая, увеличивается шанс нахождения критичных уязвимостей, которые могут быть использованы злоумышленниками, на множестве устройств.	Умеренная
Поддержка удаленного администрирования	Поддерживается	Поддерживается
Механизмы аутентификации	Устойчив к перехвату паролей, при корректной конфигурации механизмов аутентификации	Устойчив к перехвату паролей, при корректной конфигурации механизмов аутентификации
Выполнение привилегированных операций	Выполняется ОС	Выполняется самим приложением с временным повышением привилегий
Модель пользователей	Иерархическая	Одноуровневая
Защита от переполнения буфера	Отсутствует, ОС написана на языке провоцирующим такие ошибки	Отсутствует, ОС написана на языке провоцирующим такие ошибки
Возможность доступа в адресное пространство чужого процесса	Имеется у группы администраторов и у системных процессов	Имеется только в случае root-привилегий
Возможность отладки процессов	Имеется, разрешена по умолчанию	Имеется, но связана с рядом ограничений

На сегодняшний день, наиболее острой проблемой верхнего уровня наряду с другими, является несанкционированный доступ к компонентам системы. Еще в 2011 году, опубликованная статья Siemens SIMATIC Remote, обход аутентификации (которого не существует) [28]. Отмеченная там уязвимость заключалась в использовании стандартных паролей для доступа к трём сервисам,

которые можно использовать после установки Siemens SIMATIC: WEB, VNC и telnet соответственно. При этом, для работы с WEB, администратору, необходимо использовать два параметра: учетную запись и пароль, а для работы с VNC, как пользователю, так и администратору - только пароль. Т.е. злоумышленнику получить доступ к VNC значительно легче. Нужно понимать, что политики ИБ устройства располагают отдельно и при изменении параметров аутентификации одного сервиса, например, VNC или Telnet, смена на другом не происходит (Web). Если пользователь меняет пароль на новый, который включает специальные символы, это будет воспринято системой, как ошибка. Система использует механизм восстановления конфигурации и пароль будет сброшен на тот, что использован в файле конфигурации по умолчанию, т.е. сброшен на стандартный. При удачной авторизации в WEB, возвращаются сессионные cookie’s, внутри которых содержатся данные о логине, пароле (не hash, а сам пароль) и номере сессии. Использование пары логина и пароля по умолчанию, дают возможность злоумышленнику управлять в полной мере как промышленными контроллерами, так самой SCADA системой в целом. Для выявления подобных проблем необходимо менять пароли от всех учетных записей, поставляемых по умолчанию и время от времени проводить внутренний IT-аудит компании. Это позволит сократить расходы на возможные простои оборудования и потерю конфиденциальной информации.

Наряду с этими проблемами аутентификации, возникают другие, не менее критичные. Дело в том, что получение доступа к WEB, VNC или telnet на контроллерах и рабочих станциях не имеет (по умолчанию) ограничения по количеству попыток ввода пароля. Данное ограничение отсутствует в большинстве случаев использования Siemens Simatic. При этом стандартные средства защиты такие как капча (Captcha), столь часто используемые в аутентификации Интернет-ресурсов, невозможно применить для промышленных систем. Виной тому использование стандартных форм авторизации (HTTP и Basic Auth) и с невозможность их изменения, т.к. они зашиты в прошивку. Разумеется, не все вендоры используют формы авторизации собственной разработки. Но даже в

случае возможности их изменения, возникает другая проблема: все компоненты АСУ промышленными объектами за исключением шлюзов, могут быть размешены в отдельном VLAN без доступа к сети Интернет, что значительно затрудняет использование капчи и поддержание её актуальных баз.

Средний уровень архитектуры.

Рассмотрим более подробно управляющее оборудование среднего уровня типичной АСУ промышленными объектами: контроллеры, многофункциональные устройства, состоящие из приборов на базе микропроцессоров, с различными функциональными возможностями. Первые контроллеры выпускались со статично зафиксированным функциональным набором, предназначенным под определённый тип задач. Современные контроллеры – универсальны, позволяют решать огромный спектр задач автоматизации. Как правило контроллеры выбираются под решения определённых задач, т.е. в соответствии с требуемым функционалом и характеристиками, например надежность, диапазоны рабочей температуры, наличие сертификатов и разрешений для использования на промышленных объектах и др. В процессе активного развития контроллеров наблюдаются следующие основные тенденции:

- расширение функциональных возможностей;
- увеличение количества поддерживаемых протоколов передачи данных;
- развитие направления "открытых систем";
- применение характерных технологических текстовых и графических языков программирования;
- минимизация размеров;
- минимизация стоимости;
- уменьшение аппаратных отличительных особенностей между компьютерами и контроллерами;
- наличие конструктивных особенностей для работы в тяжелых климатических и производственных условиях;

Типовая архитектура PLC включает в себя следующий набор основных компонентов и связей между ними в соответствии с рисунком 1.7.

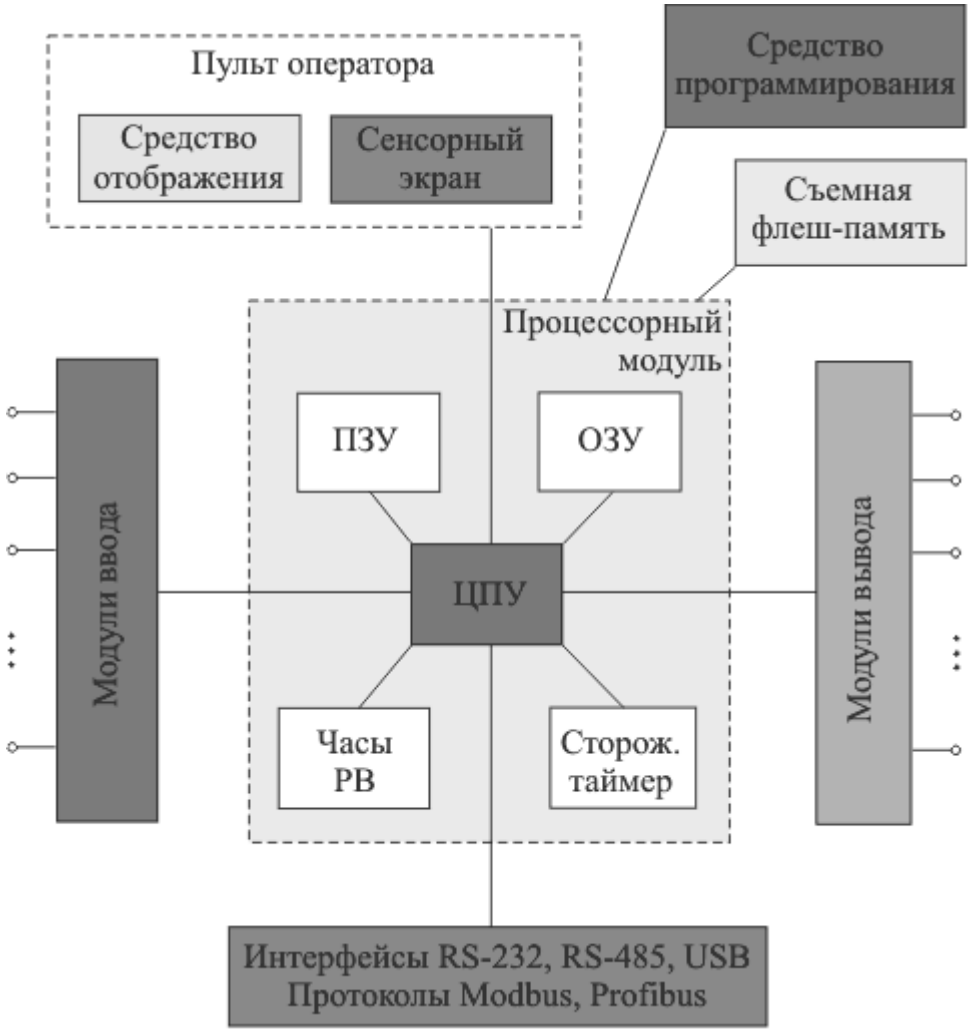


Рисунок 1.7 Типовая архитектура PLC

Основным показателем PLC является количество каналов ввода-вывода. По этому признаку PLC делятся на следующие группы [84]:

- нано-PLC (менее 16 каналов);
- микро-PLC (более 16, до 100 каналов);
- средние (более 100, до 500 каналов);
- большие (более 500 каналов).

Процессорный модуль состоит из микропроцессора, запоминающего устройства, часов реального времени и таймера. Основные характеристики микропроцессора: разрядность, тактовая частота, архитектура, наличие операций с

плавающей точкой, типов портов input/output, температурный диапазон и потребляемая мощность [84].

Ёмкость памяти определяет количество переменных, которые могут быть обработаны в процессе функционирования PLC. Задержка, возникающая при доступе к памяти - одна из главных характеристик, влияющих на быстродействие PLC в целом. Память подразделяется на несколько уровней, в зависимости от частоты использования в ней данных и быстродействия. Основными типами памяти являются ПЗУ, ОЗУ и набор регистров [84].

Сторожевой таймер (Watchdog Timer - WDT) представляет собой счетчик, который считает импульсы тактового генератора и, в нормальном режиме, периодически сбрасывается (перезапускается) работающим процессором. Если процессор "зависает", то сигналы сброса не поступают в счетчик, он продолжает считать и при достижении некоторого порога вырабатывает сигнал "Сброс" для перезапуска "зависшего" процессора [84].

Часы реального времени (РВ) представляют собой кварцевые часы, которые питаются от батарейки и поэтому продолжают идти при выключенном PLC. Часы РВ используются, например, для управления уличным освещением в зависимости от времени суток, в системах охраны объектов и других случаях, когда необходима привязка данных или событий к астрономическому времени [84].

С появлением на PLC дополнительных интерфейсов и расширением функционала (например, Web интерфейс для тестирования, реализованный в составе контроллера), появляется ряд проблем с безопасностью и надежностью функционирования АСУ промышленными объектами. Одна из них, это наследование уязвимостей (как в случае SCADA-систем) платформы PLC. При расширении функционала PLC, используют технологии, идентичные используемым в информационных системах. Соответственно, список уязвимостей растёт пропорционально добавленному функционалу и списку угроз, которым подвержены протоколы, на основе которых работают новые инструменты PLC. К проблемам можно также отнести отсутствие интегрированных средств защиты и

отсутствие использования безопасных протоколов при приеме/передаче данных внутри сети.

Нижний уровень архитектуры.

На этом уровне происходит сбор информации о текущем ТП, состоянии оборудования и осуществляются непосредственные исполнительные воздействия на процесс, определенные регулятором [2]. Именно с этого уровня начинается контроль состояния ТП, при этом большинство оборудования уровня не имеют средств визуализации. Контролируемая величина, считанная датчиком, преобразуется в стандартный сигнал, затем сигнал отправляется на средний уровень, на котором обрабатывается контроллером. Необходимая информация передается на верхний уровень, после чего происходит анализ полученных данных и, при необходимости, их визуализация. Выбор оборудования нижнего уровня (датчиков и других вторичных измерительных приборов) основан на особенностях их функциональных и конструктивных параметров по отношению к производству, на котором они будут использованы. К основным требованиям при выборе можно отнести следующее:

- однозначная зависимость класса точности выходного и входного сигналов;
- отношение цена/качество;
- стабильность работы во времени;
- возможность эксплуатации в требуемых условиях;
- наличие необходимых интерфейсов;
- наличие необходимых креплений;
- необходимая точность и чувствительность;
- небольшие размеры и общая масса;

Затрагивая тему ИБ нижнего уровня, нужно отметить, что ранее все векторы атак были направлены сверху вниз, т.е. атака начиналась всегда с операторского уровня. С развитием технологий появилась новая тенденция - атака в обратном направлении. Принцип атаки состоит в том, что через уязвимый датчик атакуются верхние части системы (средний и верхний уровни). Новая тенденция позволяет

при помощи датчика отправить легитимные данные вместе с XML-инъекцией, взаимодействующей с частью системы следующего уровня (контроллером). Так нарушитель отправляет данные в MES-систему или другие аналогичные системы. Затем происходит переход по указанной в инъекции ссылке, как правило, на внешнюю схему XML, и скачивание требуемого нарушителем файла. Тем самым, нарушитель может считывать данные с сервера и расширять функционал атаки. В определенных условиях (когда не используются VLAN), описанная атака может привести нарушителя в адресное пространство обычной корпоративной сети.

1.3 Анализ методов и алгоритмов повышения надежности через ИБ автоматизированных систем управления промышленными объектами

В последние годы множество исследователей занимаются изучением проблемы фишинга в АСУ промышленными объектами. В работе исследователей Института электроники и телекоммуникаций Южной Кореи предложен общий подход, основанный на белых списках [103]. Подход запрещает доступ к явным фишинговым сайтам. При посещении пользователем любого WEB-сайта, URL-адрес и IP-адрес WEB-сайта передаются в механизм доступа Access Enforcement Facility (AEF), для проверки легитимности сайта. Если URL-адрес, переданный в AEF, соответствует записи в белом списке, проверяется сходство IP-адресов. Если IP-адреса совпадают, то сайт легитимный; в противном случае сайт считается фишинговыми. Данный алгоритм не учитывает временную метку посещения ресурса, что в результате масштабирования создаёт проблемы с производительностью из-за увеличения перечня записей в списке, а также не предполагает механизма очистки списка. В разделе 2.1, представлен схожий подход, с устранением вышеописанных недостатков алгоритма.

Исследователи Университета Фудан Китайской Народной Республики предложили подход «Automated individual white-list» (AIWL) [91]. При

использовании которого, пользователи самостоятельно поддерживают собственные «белые списки». Дополнительно, «белые списки» хранятся индивидуально у каждого пользователя на АРМ, что значительно ускоряет скорость проверки по «белому списку». Каждый «белый список» содержит различные параметры WEB-страниц, такие как URL-адреса виджетов загружаемых на страницу и др. В данном случае, узким местом является локальное хранение списка: в случае компрометации списка, пользователь становится уязвимым. Решить данную проблему предложено в разделе 2.1 хранением списков на удалённом сервере с шифрованием файловой системы.

В результате исследований фишинговых страниц, авторами была предложена модель с использованием авто-обновляемого белого списка легитимных сайтов и уведомлений, об отсутствии URL-адреса в списке, для пользователей [114]. Валидность WEB-страницы проверяется на основе двух компонентов: 1) сопоставление имени домена и его IP-адреса; 2) исследования особенностей гиперссылок из исходного кода исследуемых страниц. Ключевой особенностью модели является механизм, который в результате проверок URL-адреса, автоматически добавлял его в белый список. В данном случае, узким место является поддержание в актуальном состоянии белых списков, поскольку злоумышленники часто создают новые сайты и доменные имена, а старые отключают. Дополнительной проблемой, является увеличение требований к вычислительным мощностям, в случае увеличения списков. В разделе 2.1 представлен алгоритм, согласно которому URL-адрес попадает в белый список, только в случае прохождения всех проверок, но по истечении N дней с момента добавления ресурс проверяется повторно. Если при проверке доменное имя недоступно, либо отсутствует в списке, то ресурс исключается из списка, иначе обновляется дата посещения.

Исследователями предложен подход, основанный на функциях метода CANTINA, в котором использован алгоритм Term frequency and inverse document frequency (TF-IDF) для определения ключевых слов в содержимом WEB-страницы [122]. Найденные слова проверяются с помощью поискового движка компании

LLC «Google». Если в результирующих страницах поисковых запросов найдена информация с исследуемой страницы, то страница считается легитимной. Данный метод обладает сравнительно низкими показателем 89% TP и 1% FP, при получении показателей по 100 фишинговым URL-адресов и 100 легитимным URL-адресов. В результате развития метода CANTINA исследователями другими исследователями предложена новая версия алгоритма названная CANTINA+ [118]. Новая версия алгоритма определяет фишинговые ресурсы по уникальному набору данных со следующими метриками: TPR равен 92.54%, FPR равен 1.41%, F1-мера равна 95.92%. Основным недостатком данного метода является статический список ключевых слов. В разделе 2.2 алгоритм CANTINA+ адаптирован под особенности АСУ промышленными объектами и дополнен механизмом поиска форм авторизации при помощи ключевых слов собранных при помощи n-граммного метода, которые наиболее часто встречаются среди фишинговых ресурсов у пользователей АСУ промышленными объектами. Дополнительно, были рассмотрены условия определения фишинговых URL-адресов из CANTINA+. Условия трёхсегментности CANTINA+ [122], где каждый сегмент состоит из не менее двух символов (буква, цифра или символ подчеркивания). Применение сегментов метода генерирует множество ложных срабатываний. Один из примеров, страница http://w.daniel.org/company/contacts.page.html определяется методом CANTINA+ как подозрительная, хотя страница полностью валидна. Этот недостаток был учтен при разработке алгоритма, описанного в разделе 2.3.2, представленные условия трёхсегментности использованы не были.

В свою очередь, исследователи департамента компьютерных наук Стэнфордского университета Калифорнии разработали расширение браузера «Spoof-Guard» на основе серии эвристик, при помощи которых обнаруживаются фишинговые WEB-страницы [93]. Эти эвристики группируются по двум типам, с сохранением состояния и без. Метод с сохранением состояния подразумевает проверку, посещался ли ресурс пользователем ранее. Метод без сохранения состояния, подразумевает полную проверку ресурса по всем разработанным эвристикам. Несмотря на эффективность механизма получения URL-алреса, в

данном подходе, он не использован в данной работе, поскольку пользователям АСУ промышленными объектами, как правило, запрещено использовать сторонние расширения в браузерах. Кроме того, с учетом особенностей АСУ промышленными объектами, приведенных в разделе 1.3, не всегда имеется возможность использования новых версий браузеров. Механизм использования истории посещений пользователей [109] – переработан в персональный «белый список» и описан в разделе 2.1.

Метод обнаружения фишинговых электронных писем [97] основан на особенностях заголовков в письмах и их содержимом. Метод представлен десятью функциями идентификации фишинговых электронных писем, среди которых проверяется «возраст» доменных имен, обнаруженных в теле и заголовках электронного письма при помощи WHOIS-поиска. В зависимости от данного показателя срабатывает спам фильтр. WHOIS-поиск применен в алгоритмах представленных во 2 главе, для определения возраста доменных имен и возраста форм для исследуемого URL.

В другом исследовании был разработан тулбар для браузеров под названием «Phishark» [99]. Исследователи изучили и проанализировали множество фишинговых атак, в результате чего, были выбраны 20 алгоритмов для определения фишинговых WEB-страниц. Затем ими была проведена оценка эффективноси выбранных алгоритмов в части определения как фишинговых, так и легитимных страниц. В результате исследователи получили следующие значения метрик, которые описаны в разделе 1.3: TPR равен 97.60%, FPR равен 2.4%, Precision равен 97.6%, F1-мера равна 97.6%. Представленные исследователями алгоритмы имеют ряд недостатков: алгоритм «Проверка подлинности кода страны» в ряде случаев некорректно определял принадлежность домена к стране; алгоритм «Доменное имя в пути URL» не учитывал использование злоумышленниками дублей доменов верхнего уровня. Данные алгоритмы были доработаны с учетом недостатков, доработанная версия алгоритма «поиска дублей доменов верхнего уровня в URL» представлена в разделе 2.3.2, доработанная

версия алгоритма «сопоставления домена верхнего уровня с кодом страны его IP-адресом» представлена в разделе 2.3.9.

Метод, с помощью которого извлекаются свойства WEB-страниц (заголовок, текст, мета поля и т.д.) и http транзакции, был разработан для определения фишинговости WEB-страниц [111]. Извлеченные свойства используются для поиска аномалий специальным классификатором страниц. Основываясь на нескольких методах обнаружения фишинга, исследователями Университета Карнеги Меллон Пенсильвании был разработан гибридный метод обнаружения фишинговых страниц. В результате реализации метода исследователи получили следующие значения метрик, которые описаны в разделе 1.3: TPR равен 90.60%, FPR равен 1.95%. Сам же метод основан на двух подходах: Information Retrieval (IR) и Information Extraction (IE) [117]. Оба подхода использовались авторами после того, как страница отображалась у пользователя. В случае с АСУ промышленными объектами, это не представляется возможным. Возможные фишинговые атаки должны быть пресечены до отображения страницы, с целью минимизации рисков загрузки вредоносного содержимого. Поэтому во 2 главе диссертации разработаны алгоритмы, которые используются превентивно, т.е. до загрузки страницы пользователем.

Особенности фишинговых и легитимных URL-адресов, являются отдельным направлением исследований. Среди которых, множеством исследователей определены следующие особенности, которые в большей степени ассоциируются с фишинговыми URL-адресами: дополнительные символы в составе URL-адреса; ключевые слова в URL-адресе; применение сертификатов, не соответствующих имени ресурса; и д.р. Среди вышеописанных особенностей, были выбраны наиболее характерные для фишинговых URL-адресов, обнаруженных при атаках на АСУ промышленными объектами. В последствии данные особенности были применены для разработки алгоритмов определения фишинговых URL-адресов во 2 главе диссертации, такие как: двойные слеши (“//”), точки в содержимом URL-адреса, наличие нескольких доменов верхнего уровня и др.

Модель «PhishDMA», была разработана для людей с нарушением зрения [101]. Авторы использовали каскад фильтров, представленный в виде модели, и различные функции для определения фишинговых URL-адресов. В результате имитационных исследований модели, исследователи получили следующие значения метрик, которые описаны в разделе 1.3: TPR равен 90.54%, FPR равен 5.82%, TNR равен 94.18, FNR равен 9.46%. Функция «определение оценки доступности» модели «PhishDMA» доработана и представлена в разделе 2.3.15.

1.4 Метрики оценки методик и алгоритмов повышения надежности через ИБ автоматизированных систем управления промышленными объектами

При разработке анти-фишинговых методов и моделей необходимо оценивать их сравнительную эффективность. Для их оценки не существует естественной меры, поэтому используются различные способы. Рассмотрим наиболее популярные из них [44].

True Positive Rate (TPR) оценивает насколько эффективно определяются фишинговые URL-адреса (1.1):

$$TPR = \frac{TP}{(TP+FN)}, \quad (1.1)$$

где, TP - количество фишинговых URL-адресов, которые корректно определены как фишинговые, FN - количество фишинговых URL-адресов, которые определены как легитимные.

False Positive Rate (FPR) оценивает процент определения легитимных URL-адресов, которые определены как фишинговые (1.2):

$$FPR = \frac{FP}{(FP+TN)}, \quad (1.2)$$

где, FP - количество легитимных URL-адресов, которые определены как фишинговые, TN - количество легитимных URL-адресов, которые определены верно, как легитимные.

True negative rate (TNR): оценивает, насколько эффективно определяются легитимные URL-адреса (1.3):

$$TNR = \frac{TN}{(TN+FP)}. \quad (1.3)$$

False Negative Rate (FNR) оценивает отношение фишинговых URL-адресов, которые определены как легитимные (1.4):

$$FNR = \frac{FN}{(FN+TP)}. \quad (1.4)$$

Для оценки качества работы каждого алгоритма по отдельности использованы Precision (1.5) и Recall (1.6). Precision оценивает отношение правильно идентифицированных фишинговых URL-адресов, к общему количеству корректно идентифицированных. Recall определяется аналогично TPR (1).

$$Precision = \frac{TP}{(TP+FP)}, \quad (1.5)$$

$$Recall = \frac{TP}{(TP+FN)}. \quad (1.6)$$

Существует несколько способов объединения Precision и Recall в агрегированный критерий качества. В данном случае используется F-measure: это среднее гармоническое между precision и recall (1.7).

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (1.7)$$

Mathews Correlation Coefficient (MCC) - коэффициент корреляции, который учитывает отношение TP, TN, FP, FN вне зависимости от их значений. MCC варьируется от -1 до 1, что интерпретируется следующим образом: чем ближе значение MCC к 1, тем выше корректность определения фишинговых URL-адресов алгоритмом; чем ближе значение MCC к -1, тем больше ошибок при определении фишинговых URL-адресов алгоритмом; чем ближе значение MCC к 0, определение фишинговых URL-адресов алгоритмом сделано случайным образом.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}. \quad (1.8)$$

Balanced Error Rate (BER) - сбалансированный коэффициент ошибок, который определяет среднее между TP, TN, FP, FN или между FPR и FNR (1.9):

$$BER = 0.5 \cdot \left(\frac{FP}{(FP+TN)} + \frac{FN}{(FN+TP)}\right) = \frac{FPR+FNR}{2}. \quad (1.9)$$

Дополнительно, с целью оценки качества работы каждого из алгоритмов, в процессе классификации, была использована площадь под кривой ошибок (AUC) рассчитываемая формуле (1.10). Если AUC =1.0, то алгоритм отлично определяет фишинговые URL. После проведения анализа URL-адреса, результат работы каждого алгоритма сравнивается с меткой класса (для тестовых данных) или с результатом прогнозирования классификатора (для обычных данных). Затем, полученные результаты помечаются знаками “+” и “-“, для положительных и отрицательных результатов соответственно. Затем, полученные значения группируются: $\{Y_1 \dots Y_N\}$ - положительные результаты сравнений, а $\{X_1 \dots X_M\}$ - отрицательные результаты сравнений. В результате получается Z последовательность отсортированных в порядке убывания результатов.

$$AUC = \frac{1}{MN}\sum_{j=1}^{M}(S_j - J), \quad (1.10)$$

где, N – число положительных значений результатов; M – число отрицательных значений результатов; j – порядковый номер проверяемого объекта; “S_j – J” - число положительных значений перед j-м отрицанием в последовательности Z.

Accuracy – точность определения фишинговых ресурсов (1.11).

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}. \quad (1.11)$$

В данной работе использованы все вышеописанные метрики для оценки антифишинговых алгоритмов и модели сообразно каждой конкретной задаче оценки.

1.5 Выводы по главе 1

Выполнен анализ задачи повышения надежности АСУ промышленными объектами через использование новых методов защиты и диагностирования их информационного и программного обеспечения и проведен обзор предыдущих исследований в этом направлении.

Определены место и роль задачи повышения надежности через ИБ АСУ промышленными объектами. Проанализированы особенности архитектуры АСУ промышленными объектами, определены три основных её уровня в зависимости от оборудования, размещённого на каждом из них. Представлены проблемы ИБ на каждом из уровней архитектуры АСУ промышленными объектами.

Фишинг определен в качестве основной угрозы для исследования. Рассмотрены существующие методы повышения надежности через ИБ АСУ

промышленными объектами в части, связанной с противодействиями фишинговым угрозам. Показана невозможность их непосредственного применения на практике в отношении АСУ промышленными объектами и их отдельные недостатки. Представлены метрики качественной оценки методик и алгоритмов повышения надежности через ИБ АСУ промышленными объектами.

АСУ промышленными объектами, как объект управления охарактеризована набором входных и выходных переменных, управляющих и возмущающих воздействий. В интересах повышения надежности через ИБ АСУ промышленными объектами в части противодействия фишинговым угрозам, сформирована задача управления.

Глава 2. Модель, метод и алгоритмы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

2.1 Алгоритм фильтрации, основанный на «белом листе»

Алгоритмы фильтрации, основанные на списках разрешенных адресов, отлично зарекомендовали себя в обнаружении фишинговых ресурсов [91,103,114]. Алгоритм [109], описанный в разделе 1.3, доработан с учетом особенностей архитектуры АСУ промышленными объектами. В частности, уменьшены риски внесения злоумышленниками изменений в персональные списки (ранее хранимые локально на рабочей станции) пользователей АСУ промышленными объектами за счет хранения их на удалённом сервере и их шифрования на уровне файловой системы; расширены хранимые в списках параметры (время посещения ресурса и др.) используемые для принятия решения о необходимости дальнейшей проверки ресурса; с целью актуализации состава записей в белых списках разработан механизм их очистки. Блок схема алгоритма представлена на рисунке 2.1.

Описание работы алгоритма:

1. На вход алгоритма подаются 3 возможных набора данных:
   - метка “NEW” (Label), уникальный идентификатор пользователя (Username), URL-адрес (URL), дата и время обращения (date);
   - метка “ADD” (Label), уникальный идентификатор пользователя (Username), URL-адрес (URL), дата и время обращения(date);
   - метка “CHECK” (Label),
2. Алгоритм обрезает протокольную часть URL-адреса, часть, содержащую номер порта и полный путь до страницы для выделения только имени домена;
3. Если использована метка “NEW”, то переход на шаг 3.1;

   3.1 Используется функция сопоставления исследуемого доменного имени с именами из персонального белого списка пользователя. Имя списка

соответствует уникальному идентификатору пользователя, сам же список включает в себя доменное имя, IP-адрес хоста, где размещена web-страница, дату посещения ресурса;

3.2 Если исследуемый ресурс соответствует записи в списке, предполагается, что web-сайт легитимный, доступ к нему разрешается и обновляется дата посещения в персональном белом списке;

3.3 В противном случае статус ресурса неизвестен, и URL-адрес перенаправляется в алгоритм из раздела 2.2 для дальнейшей проверки.

3.4 Переход на шаг 6;

4. Если использована метка “ADD”, то переход на шаг 4.1;

4.1. После успешного прохождения исследуемым доменным именем всех алгоритмов либо в результате проверки с шага 5, используется функция проверки доменного имени и IP-адреса перед добавлением в персональный список. Формируются запросы посредством команды nslookup, к локальному Domain Name System (DNS)-серверу и к удаленным корневым DNS-серверам;

4.2. Результаты запросов сравниваются;

4.3. Если IP-адреса совпадают, то проверяем соответствие данной записи у пользователя в списке, предполагается, что подобная запись может быть добавлена ранее,

4.4. Если запись найдена, то обновляется дата посещения в списке и доступ к ресурсу разрешается;

4.5. Иначе, в список добавляется новая запись в список и доступ к ресурсу разрешается;

4.6. Если IP-адреса не совпадают, в список добавляется запись с IP-адресом от корневых DNS-серверов и дополнительно производится уведомление соответствующего сотрудника службы ИБ АСУ промышленными объектами о возможной атаке “dns poisoning” на локальный DNS-сервер;

4.7. Переход на шаг 6;

5. Если использована метка “CHECK”, то переход на шаг 5.1;

5.1. Формируем общий список записей из всех персональных списков, считаем общее количество записей N в этом списке, счётчик I = 1;

5.2. До тех пор, пока I <= N, идем построчно по общему списку записей, начиная с 1;

5.3. Считаем разницу между датой сегодняшней и датой из записи;

5.4. Если разница более 30 дней, то переход на шаг 5.5, в противном случае инкрементируется счетчик I и переход на следующую запись из общего списка на шаг 5.2;

5.5. Проверяем доступность URL-адреса командой wget, если URL-адрес доступен, то переход на шаг 5.6, в противном случае на шаг 5.7;

5.6. Формируем запрос с меткой ADD, и он отправляется на вход, инкрементируется счетчик I и переход на следующую запись из общего списка на шаг 5.2;

5.7. Удаляем запись из соответствующего персонального списка, I=I+1 инкрементируется счетчик I и переход на следующую запись из общего списка на шаг 5.2;

6. Выход.

Набор данных с меткой “NEW”, поступает при первичной проверке URL. Набор данных с меткой “ADD”, поступает при успешном прохождении URL-адресом всей методики, когда принято решение о его легитимности. Набор данных с меткой “CHECK”, поступает раз в сутки, а модификация списка производится с целью уменьшения использования вычислительных мощностей, используемых для выполнения всех алгоритмов, поскольку, если ресурс попадёт в белый список, то повторную его проверку делать не потребуется. В качестве регулирующего параметра N (количество дней хранения) выбрано значение 30, в соответствии с результатами имитационных исследований и анализа эффективности, представленными в 4 главе. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе.

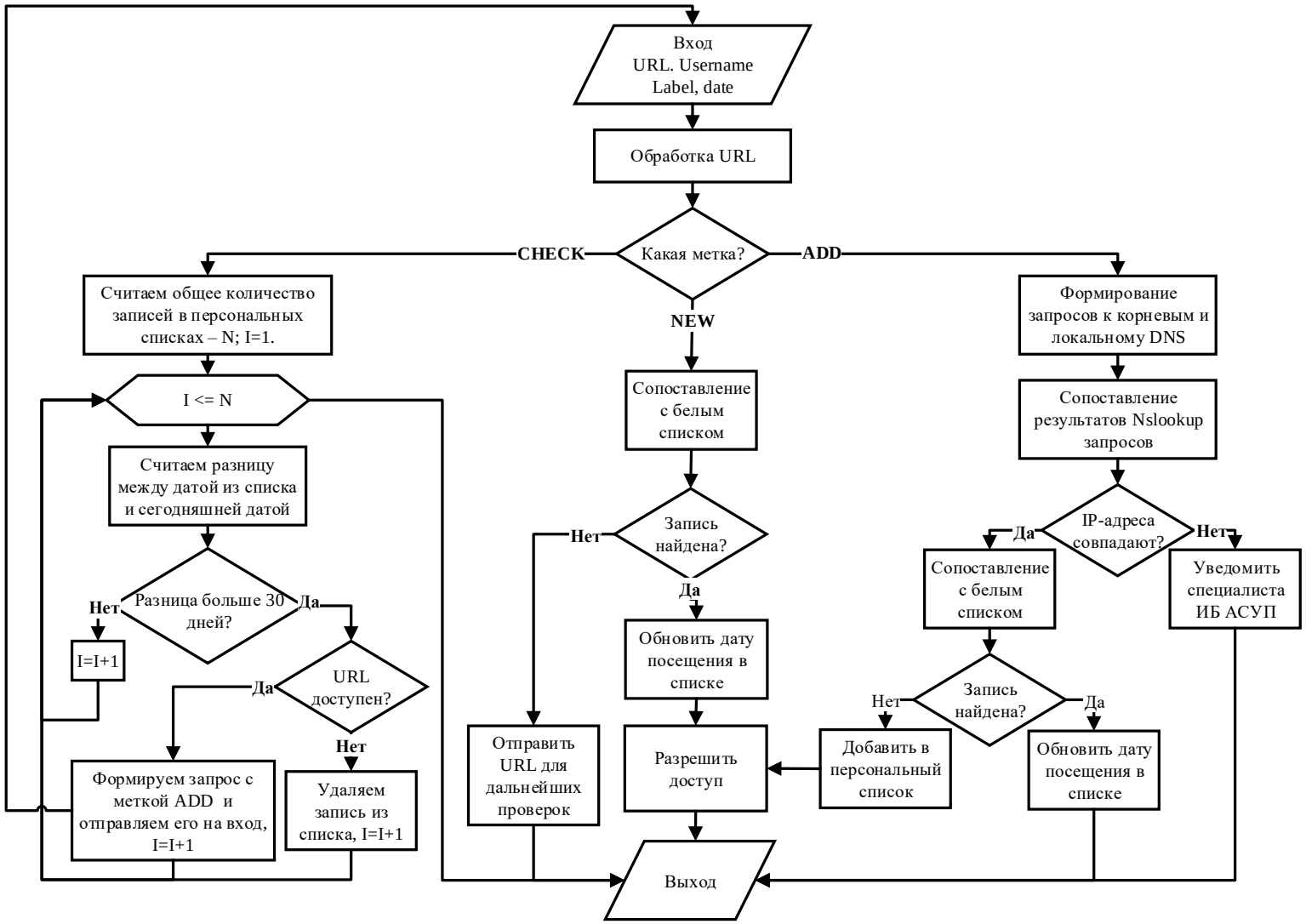


Рисунок 2.1 Алгоритм фильтрации, основанный на белом списке

2.2 Алгоритм фильтрации, основанный на поиске форм авторизации

В настоящее время на множестве ресурсов, размещенных в сети Интернет используются формы авторизации пользователей для обеспечения их авторизации. Такие формы широко применимы и на ресурсах, которые используют пользователи АСУ промышленными объектами, в том числе формы авторизации применимы и для компонентов самих АСУ промышленными объектами (например, HMI панели или системы самодиагностики Siemens Simatic S7). Часто, злоумышленники предпринимают попытки кражи регистрационных данных пользователей при помощи фиктивной формы авторизации. Для решения проблемы поиска форм авторизации разработан отдельный алгоритм [109], который позволяет проверять страницу на наличие хотя бы одной формы авторизации. Если нет форм регистрации/авторизации нет, то ресурс считается легитимным, поскольку пользователь никуда не вводит свои данные. Это улучшает производительность алгоритма в отношении ресурсов, на которых нет искомых форм. За основу взят алгоритм CANTINA+ и использованы ключевые слова связанные с аккаунтингом (например: login, user, username, pass, password и др.). Эти ключевые слова используются для определения, может ли страница, оказаться формой авторизации. Блок схема алгоритма представлена на рисунке 2.2.

Описание работы алгоритма:

1. На вход модуля подаётся URL-страницы, она загружается и всё дерево Document Object Model (DOM) страницы проверяется на наличие тэгов ввода и форм;
2. В случае обнаружения и тэгов ввода и тэгов форм автоиризации:
   2.1. В той же области, где найдены тэги, производится поиск, по ключевым словам, (search, searchbox, watch и др.), для проверки,

является ли страница поисковой, если слова есть, продолжаем шаг 2.1, если слова не найдены переход на шаг 2.2.

2.2. Ищем ключевые слова, связанные с аккаунтингом, для проверки, может ли форма использована для авторизации. Если является, прекратить поиск и вернуть результат «1», в противном случае переходим на шаг 2.3.

2.3. Дополнительно, проводим поиск ключевых слов в узлах DOM исследуемой страницы до максимальной глубины. Если слова найдены, прекратить поиск и вернуть результат «1», в противном случае переходим на шаг 2.4.

2.4. Поскольку злоумышленники могут использовать изображения вместо форм авторизации, дополнительно производится поиск изображений. Если они обнаружены, тогда поиск прекращается и возвращается результат «1», в противном случае переходим на шаг 1.1, для продолжения поиска. Если не найдено ни одной страницы с формой авторизации, переходим на шаг 4.

3. В случае обнаружения только тэгов ввода:

3.1 Ищем ключевые слова, связанные с аккаунтингом в DOM. Если слова найдены, прекратить поиск и вернуть результат «1», в противном случае переходим на шаг 3.2

3.2 Если DOM состоит только из изображений, вернуть результат «1», в противном случает вернуть результат «0».

4. Если страница ни имеет тэгов формы авторизации, тэгов ввода и изображений тогда вернуть 0.

Если в результате выполнения, алгоритм возвращает 1, то форма авторизации присутствует; 0 - возвращается, когда формы авторизации нет. Одним из важных подготовительных этапов для реализации данного алгоритма является отбор ключевых слов. Релевантный набор ключевых слов, позволяет увеличить точность поиска форм авторизации, что в свою очередь, в значительной степени влияет на эффективность работы алгоритма в целом.

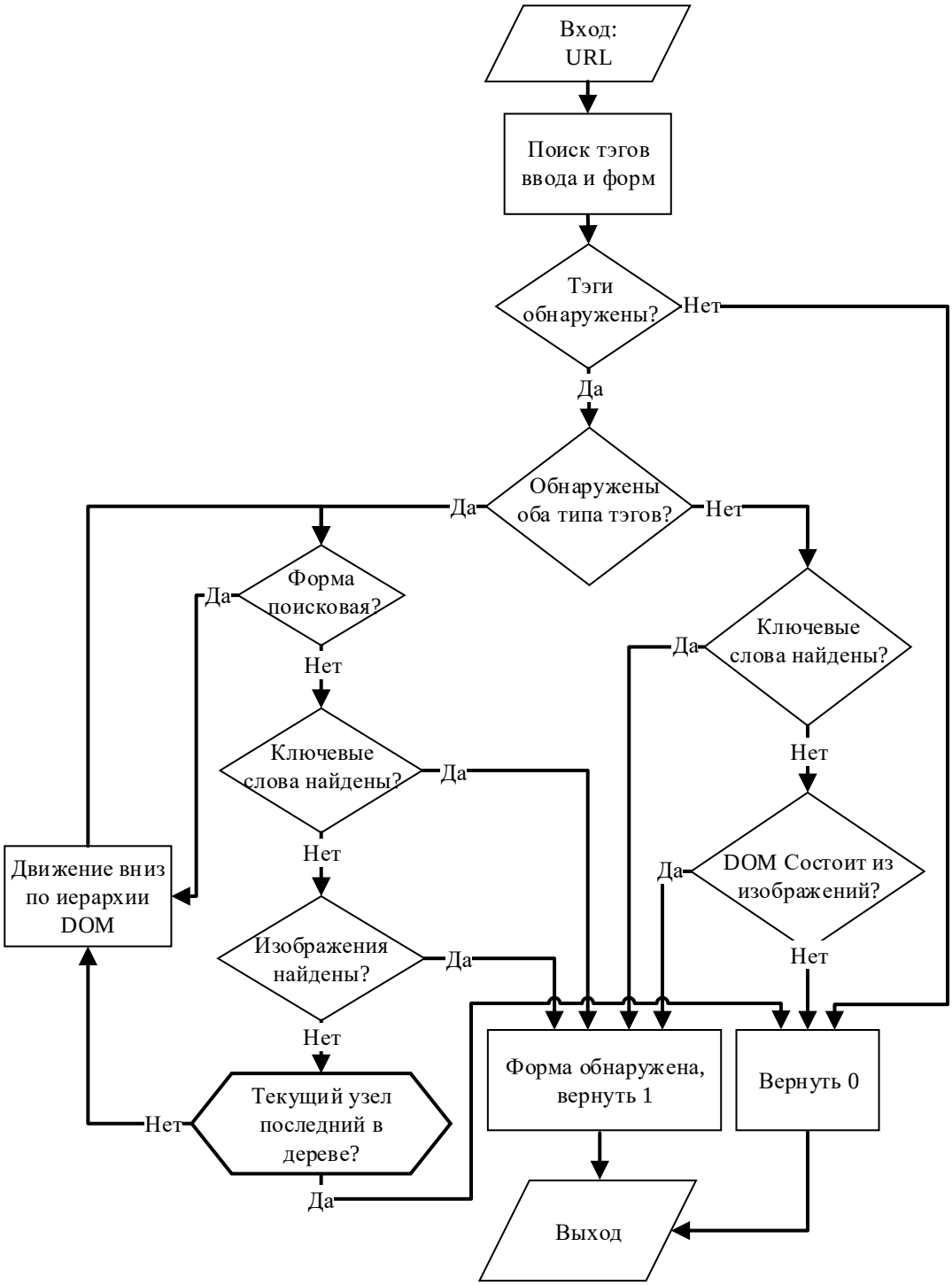


Рисунок 2.2 Алгоритм фильтрации, основанный на поиске формы авторизации

Для решения задачи поиска ключевых был извлечен текст из тэгов DOM из набора данных фишинговых и легитимных ресурсов, собранных в 4 главе. В результате анализа полученного набора данных стало ясно, что исследуемый набор состоит из многокомпонентных слов, что в свою очередь привело к необходимости их. Несмотря на существенное количество исследований,

автоматическое извлечение многокомпонентных ключевых слов, до сих пор представляет собой проблему [82,113]. Исходя из этого, было необходимо выбрать эффективный метод выделения ключевых слов и метод оценки их релевантности, относительно исследуемого набора данных.

Для выделения ключевых слов использовался n-грамный метод. Все строки из исследуемого набора данных разбивались на n-граммы, представляющие из себя подстроку из n-символов. Затем каждая подстрока проверялась по английскому словарю и словарю сокращений, на предмет корректности подстроки. Все несоответствующие данному критерию подстроки, т.е. подстроки содержащие «мусор», исключались из выборки. В результате применения метода удалось выявить артефактные подстроки со специальными символами, которые были использованы в исследуемом наборе для дальнейшего отбора наиболее релевантных ключевых слов.

Затем в отношении исследуемого набора для улучшения отбора ключевых слов был применен WoE (Weight of Evidence) анализ. Для оценки, была использована метрика IV (Information Value) помогающая ранжировать подстроки на основе их важности. Для расчёта метрики IV, предварительно требуется посчитать коэффициент WoE по формуле (2.1)

$$\mathrm{WoE_k} = \ln \frac{\left(\frac{\mathrm{N_k}}{\mathrm{N}}\right)}{\left(\frac{\mathrm{P_k}}{\mathrm{P}}\right)} = \ln \frac{\mathrm{F^-}}{\mathrm{F^+}}, \qquad (2.1)$$

где, k – порядковый номер подстроки, $\mathrm{N_k}$ – число фишинговых URL-адресов без k-подстроки, N – общее число фишинговых URL-адресов не содержащих ключевых слов из исходного набора, $\mathrm{P_k}$ – число фишинговых URL-адресов с k-подстрокой, P – общее фишинговых URL-адресов содержащих все ключевые слова в исходном наборе.

Метрика IV, рассчитывается по формуле (2.2) и всегда является положительной величиной. На ее основе определяется значимость подстроки на основе следующих условий:

- IV < 0.02 – значимость подстроки отсутствует;
- 0.02 ≤ IV < 0.1 – значимость подстроки низкая;
- 0.1 ≤ IV < 0.3 – значимость подстроки средняя;
- IV > 0.3 – значимость подстроки высокая.

$$IV = \left(\frac{N_k}{N} - \frac{Pk}{P}\right) \cdot WoE_k. \quad (2.2)$$

Использовав n-граммный метод к набору данных фишинговых и легитимных ресурсов, собранных в 4 главе, был получен перечень ключевых слов. К каждому ключевому слову из данного перечня была рассчитана метрика IV, затем перечень был отсортирован по метрике IV в порядке убывания. Первые 10 ключевых слов представленные в таблице 2.1, использованы для поиска форм авторизации.

Таблица 2.1 Ключевые слова с высоким показателем метрики IV

№	Ключевое слово	WoE	IV
1	login	-2.53968	0.600526
2	authorize	-6.41088	0.358479
3	login/	-3.16829	0.318726
4	auth	-2.53968	0.297533
5	signin	-2.45182	0.256221
6	account	-3.24786	0.244051
7	auth?	-3.12461	0.244001
8	sign-in	-4.0336	0.226542
9	#login	-3.23283	0.195667
10	logon	-3.27205	0.191007

Алгоритм фильтрации, основанный на поиске форм авторизации, позволяет увеличить точность определения фишинговых ресурсов, а так же, уменьшить использование вычислительных мощностей, отсутствие формы авторизации, трактуется, как отсутствие индикаторов компрометации, а

значит исследуемый URL-адрес – легитимный. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности и имитационные исследования алгоритма представлены в 4 главе.

2.3 Метод алгоритмических проверок

По мере развития фишинговые ресурсы становятся все более изощрёнными и подготовленными для реализации атак. С целью более точной их идентификации, в соответствии с особенностями ресурсов, которые чаще всего посещают пользователи АСУ промышленными объектами, разработан метод алгоритмических проверок [45]. Он получает на вход URL-адрес и анализирует его на наличие индикаторов фишинговости при помощи следующих шестнадцати алгоритмов:

1. Поиска IP-адреса в URL-адресе;
2. Поиска дублей доменов верхнего уровня в URL-адресе;
3. Определения нестандартного номера порта в URL-адресе;
4. Валидации доменных имён;
5. Определения возраста доменного имени;
6. Определения возраста формы авторизации;
7. Сопоставления контента страницы с доменным именем;
8. Анализа истории DNS записей домена;
9. Сопоставления домена верхнего уровня и с кодом страны его IP-адреса;
10. Поиска ключевых слов URL-адресе;
11. Валидации SSL/TLS сертификата;
12. Определения длины URL-адреса;

13. Подсчёта точек в URL-адресе;
14. Поиска специального символа @ в URL-адресе;
15. Поиска специальных символов “Слеши, протокол и порт” в URL-адресе;
16. Оценки доступности URL-адреса.

Все алгоритмы сгруппированы по функциям. Каждый алгоритм выполняется параллельно и независимо от других. В результате анализа формируется n-мерный вектор, который отправляется для анализа далее.

2.3.1 Алгоритм поиска IP-адреса в URL-адресе

Злоумышленники могут использовать IP-адреса для перенаправления пользователей АСУ промышленными объектами на фишинговые страницы. Использование IP-адресов, в первую очередь, с тем, что доменные имена платные. Во-вторых, часто злоумышленники размешают фишинговые сайты на «взломанных» серверах, доступных из сети Интернет, не имеющих DNS-записей. Следовательно, самый простой и не затратный способ их эксплуатации злоумышленниками – IP-адреса. При этом, что большинство валидных WEB-ресурсов имеют собственные доменные имена. Задача этого алгоритма проверить наличие IP-адреса в основной части URL-адреса. Блок схема алгоритма представлена на рисунке 2.3.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Как и в разделе 2.2, функции алгоритма обрезают протокольную часть URL-адреса, часть, содержащую номер порта и полный путь до страницы для получения только основной части;

3. Производится проверка основной части, по регулярным выражениям соответствующим, маске IPv4 адресов, таких как, ‘xxx.xxx.xxx.xxx’, ‘xx.xx.xx.xx’, ‘x.x.x.x’;
4. Если основная часть не содержит IPv4 адрес, то переходим на п.5, в противном случае на п.6;
5. Вернуть 1;
6. Вернуть 0.

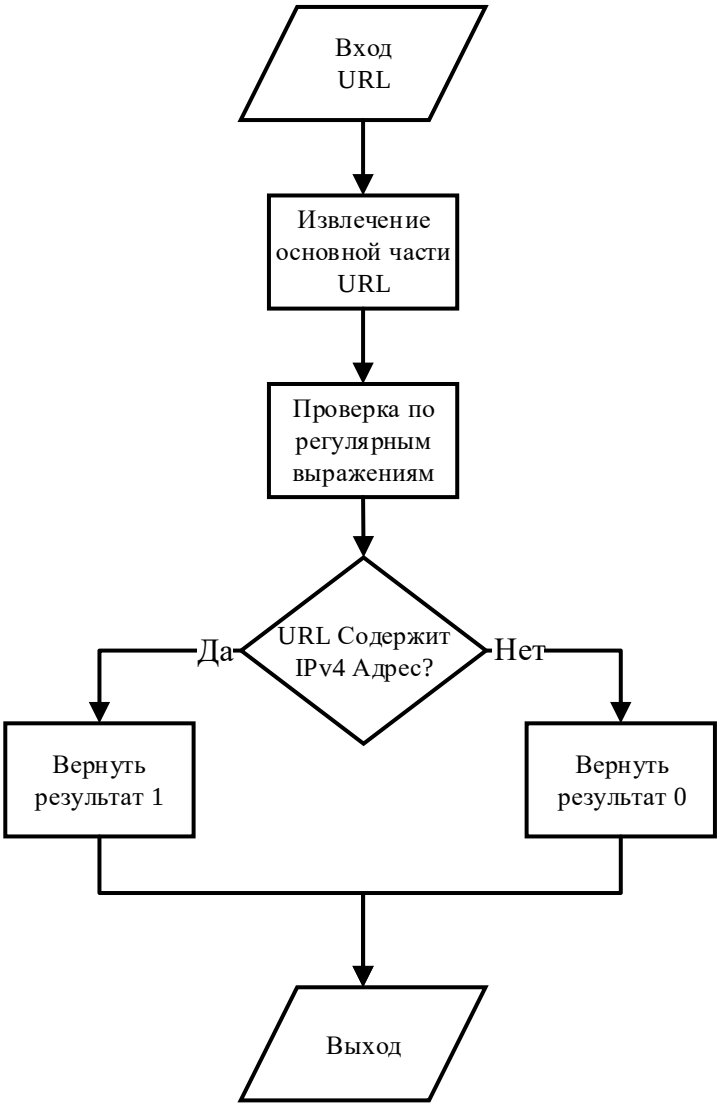


Рисунок 2.3 Алгоритм поиска IP-адреса в URL

В результате, если URL-адрес не содержит IP-адрес, алгоритм возвращает 0, что говорит о том, что исследуемый ресурс, с точки зрения

наличия IP-адреса в URL-адресе - легитимный, в противном случае фишинговый. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности и имитационные исследования алгоритма представлены в 4 главе.

2.3.2 Алгоритм поиска дублей доменов верхнего уровня в URL-адресе

Одним из возможных способов сокрытия фишингового URL-адреса, это применение URL-адресов, похожих на легитимные. Фишинговый URL-адрес может содержать часть легитимного адреса и дополнительные символы далее. В данном случае, пользователям АСУ промышленными объектами может сложно оценить валидность URL-адреса. При переходе по такому URL-адресу, пользователь может попасть на фишинговую форму авторизации и скомпрометировать себя. Для определения подобных фишинговых URL-адресов, разработан алгоритм, представленный на рисунке 2.4. Алгоритм проверяет структуру доменных имен в URL-адресе, по предварительно подготовленному, автообновляемому локальному списку, состоящему из доменов верхнего уровня.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. В полученном URL-адресе производится поиск доменов верхнего уровня по предварительно заданному списку посредством простого алгоритма поиска подстроки в строке;
3. Если домен верхнего уровня обнаружен в URL-адресе, проверяем наличие специальных символов (“:”, ”/”, “ “ и др.) в строке на позиции после домена верхнего уровня (с целью убедиться, что это

действительно домен верхнего уровня, а не набор идентичных символов);

4. Если доменов нет, перейти на шаг 7;
5. Если доменов больше 1, перейти на шаг 7, в противном случае переход на шаг 6;
6. Вернуть 0;
7. Вернуть 1.

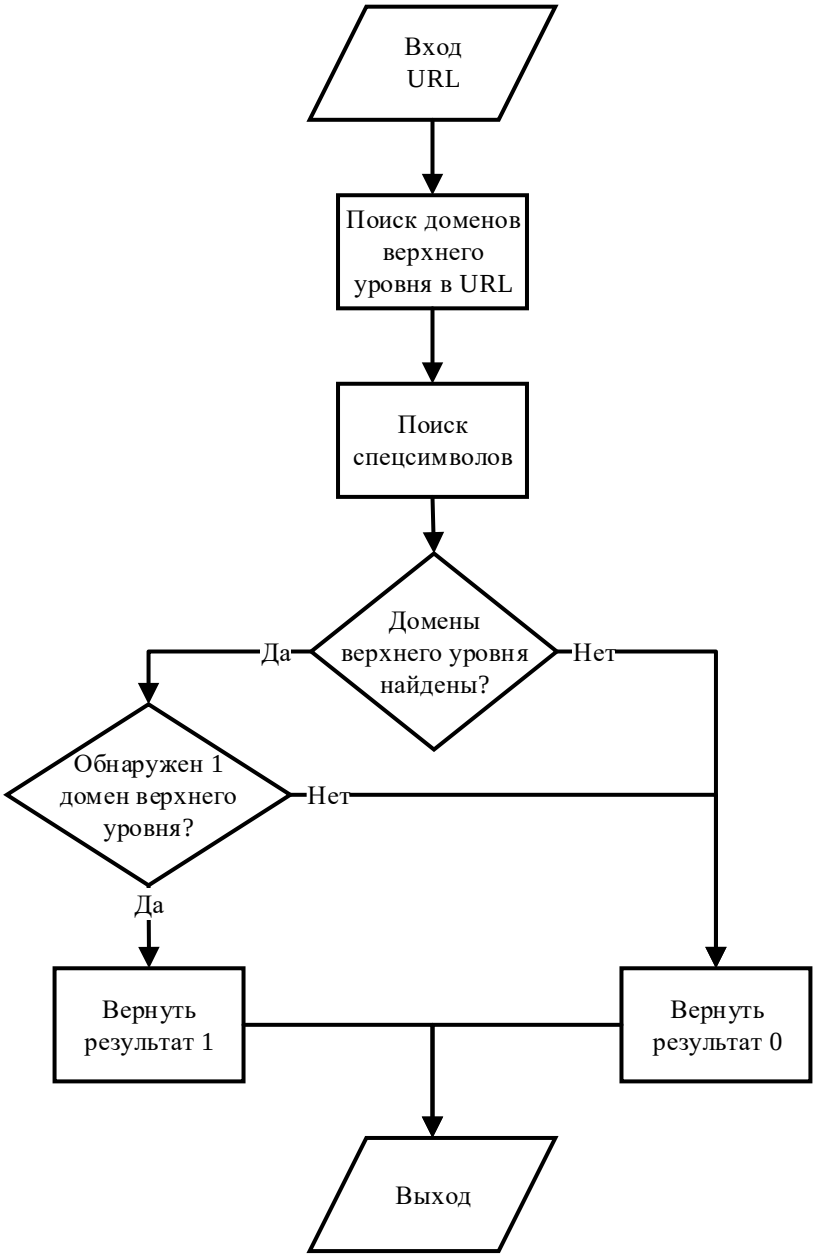


Рисунок 2.4 Алгоритм поиска повторений доменов верхнего уровня

В результате выполнения алгоритма, если URL-адрес не содержит доменов верхнего уровня или содержит более одного, алгоритм возвращает 0, что говорит о том, что исследуемый ресурс – подозрительный, в противном случае ресурс – легитимный, и возвращается 1.

Поскольку перечень доменов верхнего уровня не статичен, реализован механизм синхронизации перечня доменных имен верхнего уровня в локальном списке с официальной страницей Internet Assigned Numbers Authority (IANA) (http://data.iana.org/TLD/tlds-alpha-by-DOMain.txt), который принят, как эталонный. Раз в минуту проверяется наличие изменений в эталонном перечне, в случае изменений в нём, производится обновление перечня в локальном списке.

Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности и имитационные исследования алгоритма представлены в 4 главе.

2.3.3 Алгоритм определения нестандартного номера порта в URL-адресе

Наличие в URL-адресах нестандартных портов – один из индикаторов подозрительности URL-адресов. Например, в URL: http://yara.com:5800/login.html, для протокола http применен порт 5800, хотя, по умолчанию для http используются порты 80 или 8080. Следовательно, URL-адрес подозрительный. Аналогичная ситуация для https, валидным будет считаться исключительно порт 443. Алгоритм определения фишинговых ресурсов с нестандартным номером порта в URL-адресе представлен на рисунке 2.5.

Рисунок 2.5 Алгоритм определения нестандартного номера порта в пути URL

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Регулярными выражениями URL-адрес разбирается на протокол, доменное имя, порт;
3. Проверяется соответствие используемого протокола и порта для: http – 80,8080; для https – 443; для ftp – 21;
4. Если пары соответствуют либо присутствует только протокол, без номера порта, переход на шаг 8;

5. Сопоставляются исследуемые сегменты URL-адреса с теми, что ранее добавлены в список разрешений;
6. Если соответствие найдено, переход на шаг 8, в противном случае на шаг 7;
7. Вернуть 0;
8. Вернуть 1.

В результате выполнения алгоритма, если порт и протокол в URL-адресе соответствуют заданным условиям или домен и порт найдены в списке разрешений, то алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный, и возвращается 0. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности и имитационные исследования алгоритма представлены в 4 главе.

2.3.4 Алгоритм валидации доменных имён

Зачастую фишинговые ресурсы не могут существовать длительное время, так как они достаточно быстро попадают в «черные» списки. Затем, они отключаются злоумышленниками, поскольку смысл в их использовании пропадает. Поэтому, за время своей жизни, фишинговые ресурсы не успевают получать высокую оценку ранжирования Page Rank (PR) в поисковых системах. PR формируется алгоритмами ссылочного ранжирования. Алгоритмы получают на вход коллекцию WEB-страниц, на выходе отдают численное значение, отражающее «важность» WEB-страницы, в отношении к остальным страницам из выборки. Данный алгоритм, представленный на рисунке 2.6, разработан с целью получения PR URL-адресов и дальнейшего их анализа. Основным отличием алгоритма от созданных ранее, является

использование нескольких специальных сервисов (checkpagerank.net, prchecker.info, ahrefs.com, rankapi.net, dnschecker.org, gogolev.net) формирующих PR для URL-адреса, затем PR приводятся к одной шкале ранжирования и на основе полученных оценок формируется результат выполнения алгоритма.

Описание работы алгоритма:

1. На вход подаётся URL-адрес, флаг URL устанавливается в “true”;
2. В зависимости от механизмов приема запросов и особенностей, реализованных на сервисах формирования оценки PR, формируются RestAPI и Get запросы;
3. Сформированные запросы отправляются в системы для расчёта PR;
4. Получение PR, приведение их к общей оценочной шкале от 0 до 10(для шкалы 0–100, оценка умножается на 0.1);
5. Анализ PR, подсчёт средней оценки (все оценки складываются и делятся на количество их самих же), проверка на соответствие пороговому значению;
6. Если полученные PR - нулевые и исследовался URL-адрес(URL=true), а не доменное имя, то из URL-адреса извлекается доменное имя и подаётся на вход шага 2 доменное имя флаг URL устанавливается в false, в противном случае на шаг 6;
7. Если полученная средняя оценка выше порогового значения, на шаг 8, в противном случае на шаг 7;
8. Вернуть 0;
9. Вернуть 1.

В результате выполнения алгоритма, если средняя оценка выше порогового значения равного пяти, то алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0.

С целью увеличения точности данного алгоритма проверяется не только URL-адрес целиком, но и основной домен. Это связано с тем, что целевая

страница может быть новой, а сам домен может существовать более длительное время и иметь PR выше. Все оценки приведены к шкале от 0 до 10. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма и поиск оптимального порогового значения описаны в 4 главе.

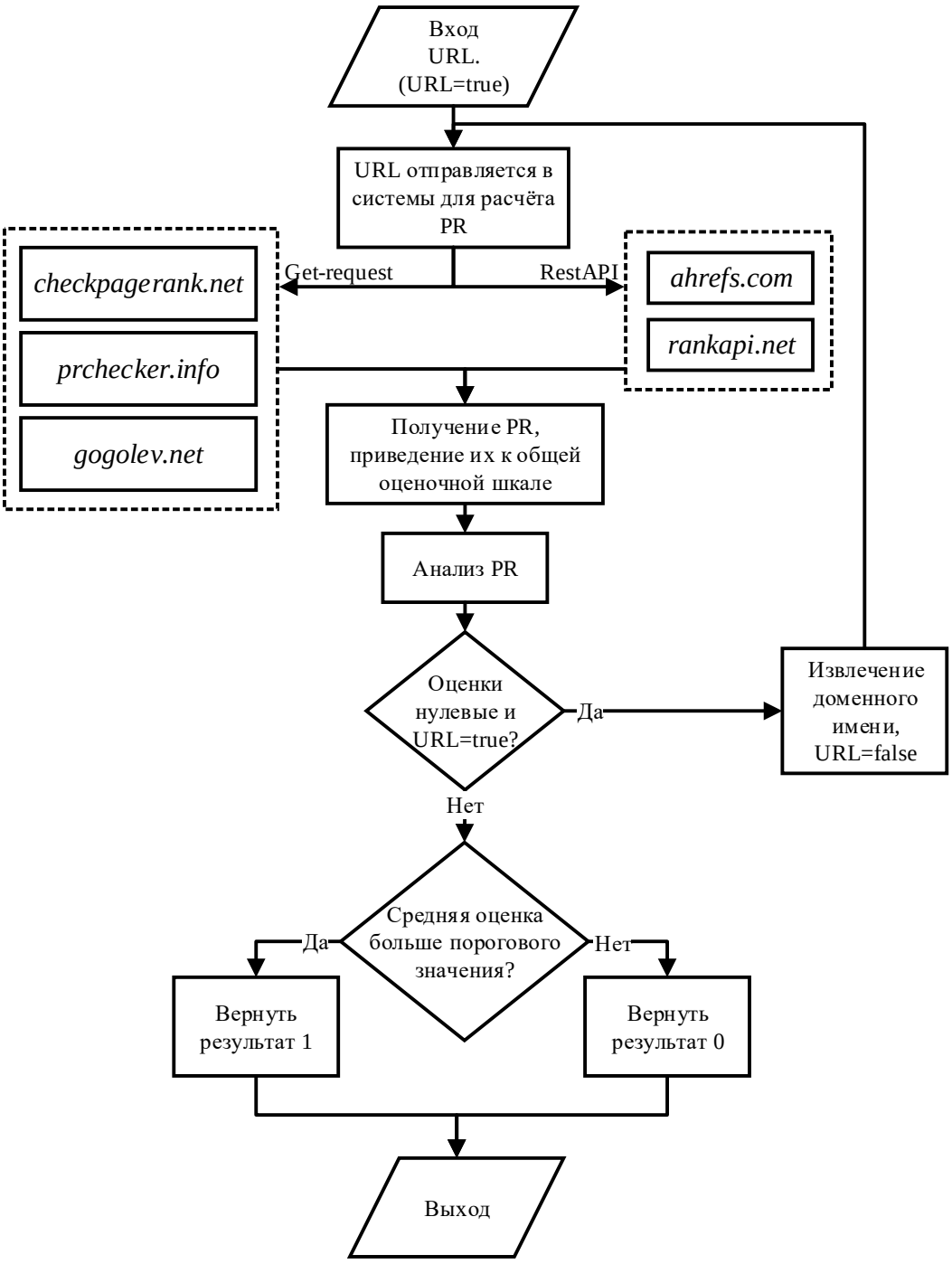


Рисунок 2.6 Алгоритм валидации доменных имён

2.3.5 Алгоритм определения возраста доменного имени

Злоумышленники часто создают новые доменные имена для реализации атак. Как правило, такие домены создаются и существуют короткие промежутки времени. Следовательно, возможно использовать значение срока жизни доменного имени, как индикатор потенциальной опасности URL. Для подсчета срока жизни доменного имени, требуется определить дату его создания. Дата создания домена, это дата его регистрации владельцем у регистратора. Что бы получить дату создания домена, можно использовать WHOIS протокол, который позволяет получить не только дату создания домена, но и информацию о владельце, администраторе, регистраторе, dns-серверах и др. Кратко, работа протокола представляется, как создание TCP соединения к WHOIS серверу, отправка к нему запроса, получение ответа, завершение сессии. Несмотря на простоту реализации, использование WHOIS протокола, создаёт ряд сложностей. Достаточно посмотреть в документ rfc3912, в котором задокументирована работа WHOIS протокола, где описано, что протокол не интернационализирован. Это означает, что WHOIS сервера за пределами РФ, могут в ответе отправлять символы в различной кодировке и различной, непрогнозируемой последовательности. Из этого следует, что первая проблема, которую нужно решить — это разный формат коммуникаций с WHOIS серверами, который требует соответствующей обработки ответа со стороны клиента. Вторая проблема, определение WHOIS сервера, который за короткое время сформирует ответ, на искомый запрос. Общий подход для решения — это проблемы, это запрос к whois.iana.org, но данный WHOIS сервер, в ряде случаев, содержит в ответе реферал на сервер, который содержит сведения о домене и иерархия рефералов, может быть больше трех. Следовательно, увеличивается время, на выполнение запроса, что плохо влияет на производительность системы в целом, т.к. алгоритм не может вернуть ответ, до тех пор, пока не получит запрашиваемые сведения.

В алгоритме, представленном на рисунке 2.7, использован WHOIS протокол для получения даты создания домена и добавлен механизм выбора WHOIS сервера.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Как и в разделе 2.2 функции алгоритма обрезают протокольную часть URL-адреса и часть, содержащую номер порта, полный путь до страницы, для получения только основной части;
3. Поиск домена в персональном списке WHOIS серверов;
4. Если домен найден в списке, переход на шаг 5, иначе на шаг 7;
5. Формируется WHOIS запрос к WHOIS серверу из персонального списка;
6. Если ответ от WHOIS сервера не получен, то переход на шаг 7, иначе на шаг 8;
7. Формируется WHOIS запрос и отправляется к WHOIS серверу whois.iana.org;
8. Обрабатывается ответ, из текущей даты вычитается дата создания домена;
9. Если полученное значение больше порогового, переход на шаг 7, в противном случае на шаг 6;
10. Вернуть 0;
11. Вернуть 1.

В результате выполнения алгоритма, если срок жизни домена выше порогового значения (14 дней), то алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0.

Для решения проблемы выбора WHOIS сервера, использован список содержащий перечень WHOIS серверов доменов, которые посещали пользователи. Данный список формируется раз в сутки. Каждому домену из персональных белых списков пользователей, формируется свой список WHOIS серверов путем WHOIS запросов к whois.iana.org.

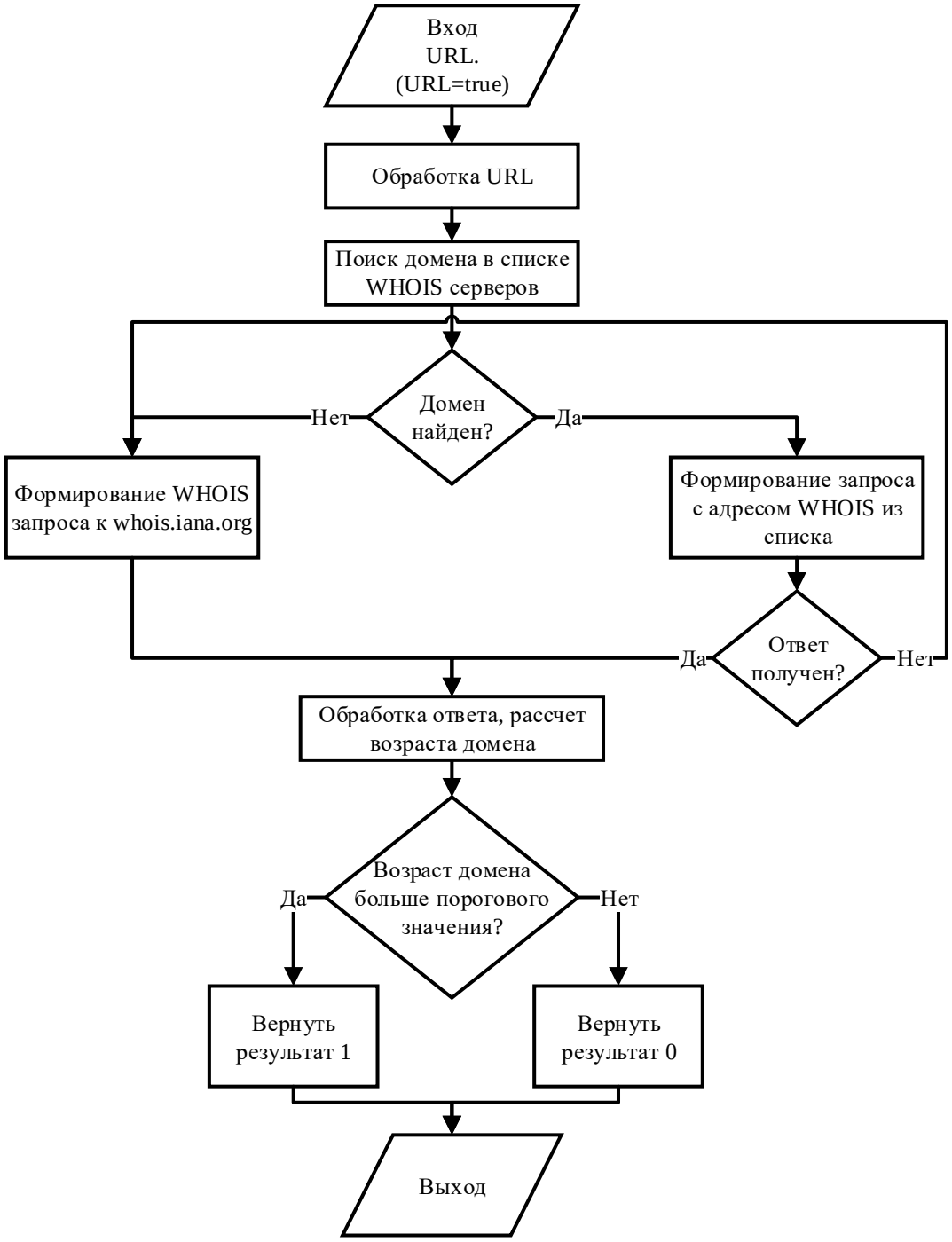


Рисунок 2.7 Алгоритм определения возраста доменного имени

С идейной точки зрения, это все тот же алгоритм, разработанный авторами в предыдущих исследованиях [97], но авторами не раскрыта реализация данного алгоритма в условиях использования WHOIS протокола. В частности, это формирование запросов к регистраторам различных зон; обработка валидных ответов; обработка ошибок в ответе и др. Практическая реализация алгоритма включена в общую архитектуру системы повышения

надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма и поиск оптимального порогового значения описаны в 4 главе.

2.3.6 Алгоритм определения возраста формы авторизации

В случаях, когда злоумышленник получил доступ к легитимному WEB-ресурсу, дальнейшим развитием атаки может быть размещение (одного или нескольких) фишингового URL-адреса, в области действий существующей формы авторизации, с целью сбора и использования учетных данных пользователей АСУ промышленными объектами в собственных целях. Следовательно, если хотя бы один из URL-адресов, из тэгов ввода и тэгов форм, имеет возраст меньше порогового, целевой URL-адрес может быть небезопасным. Исходя из этого, функционал алгоритма из раздела 2.3.5 расширен следующим образом: извлекаются URL-адреса из тэгов форм и тэгов ввода DOM исследуемой страницы. Полученные URL-адреса по одному отправляются в алгоритм из раздела 2.3.5. Затем, если возраст хотя бы одного URL-адреса меньше порогового значения, ресурс считается фишинговым, в противном случае, легитимным. Алгоритм представлен на рисунке 2.8.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Загрузка DOM страницы исследуемого URL-адреса;
3. Для каждого узла DOM исследуемого URL-адреса переходим на шаг 3.1, при завершении узлов на шаг 5;
    3.1Ищем тэги формы или ввода;
    3.2Если тэги не найдены, переход на следующий узел шаг 3, иначе на шаг 3.3;
    3.3Если тэги найдены, извлекаем URL-адрес;

3.4Отправляем извлечённый URL-адрес, в алгоритм из раздела 2.3.5;

3.5Получаем возраст извлеченного URL-адреса;

3.6Если возраст извлеченного URL-адреса больше порогового значения, то переход на следующий узел, шаг 3, иначе на шаг 4;

4. Вернуть 0;
5. Вернуть 1.

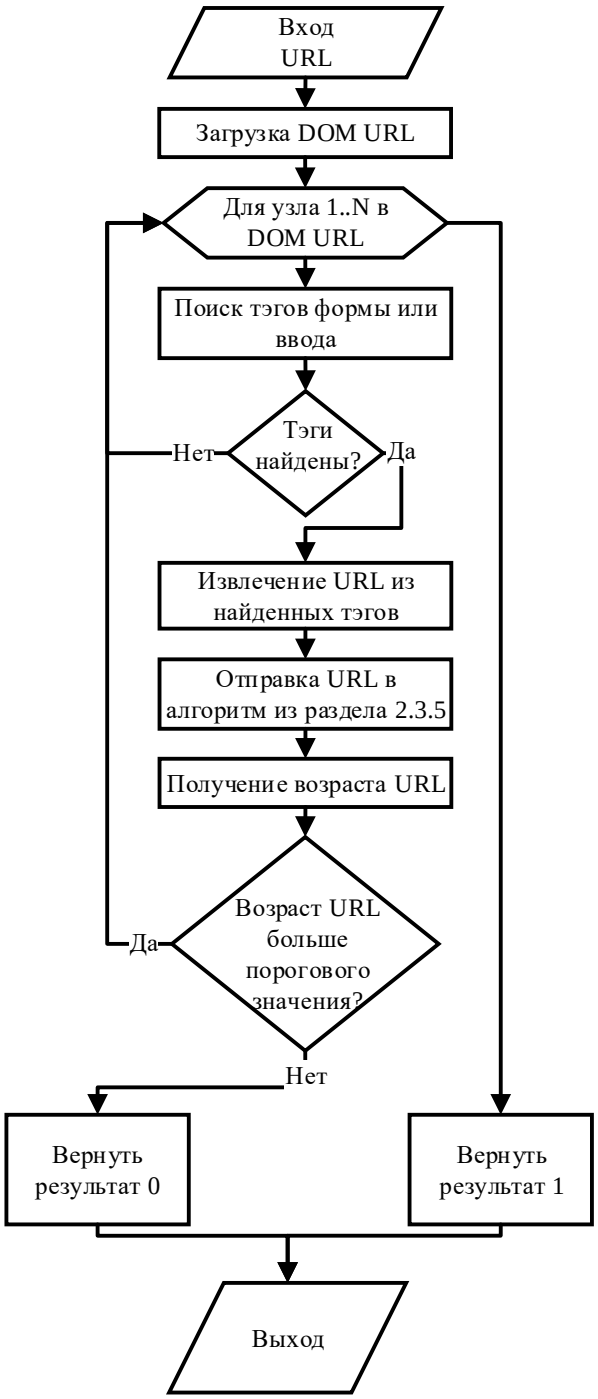


Рисунок 2.8 Алгоритм определения возраста формы авторизации

В результате выполнения представленного алгоритма, если алгоритм описанный в главе 2.3.5 возвращает 0, хотя бы для одного из URL-адреса (с целью обработки URL-адреса, которые содержат более 1 формы входа), то этот алгоритм возвращает 0, что говорит о том, что исследуемый URL-адрес – подозрительный, в противном случае URL-адрес – легитимный и возвращается 1. В данном случае, пороговое значение для возраста формы равно пороговому значению из алгоритма в главе 2.3.5. Это связано с тем, что исследуемые объекты - идентичны, а применение одного и того же значения уменьшает затраты на вычислительные мощности. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма описаны в 4 главе.

2.3.7 Алгоритм сопоставления контента страницы с доменным именем

Основная масса современных WEB-сайтов, размещенных в сети Интернет использует доменное имя, которое так или иначе связанное с контентом этого WEB-сайта. Как минимум с ним связаны основное меню, форма авторизации, форма поиска и т.д. Если исключить целевые атаки, направленные на определенную группу пользователей АСУ промышленными объектами, то злоумышленники не всегда реализуют фишинговые ресурсы с достаточной проработкой внутренних ссылок и внутреннего контента. Следовательно, эту связность внутренних ссылок и контента страницы с основным доменом, можно использовать как индикатор подозрительности URL. Например, если проанализировать DOM английской версии главной страницы WEB-сайта http://pstu.ru/, то большая часть гиперссылок ссылается на основной (на самого себя) домен, что подтверждает легитимность ресурса.

В свою очередь, в части анализа контента, DOM содержит в себе следующий набор ключевых слов: education; science; university; PSTU и др. В данном случае, набор ключевых слов, это слова, которые чаще всего встречаются в контенте исследуемого ресурса. Где одно из ключевых слов (PSTU), соответствует основной части доменного имени исследуемого ресурса, что подтверждает его легитимность. Основная идея алгоритма, представленного на рисунке 2.9 - провести проверку соответствия контента WEB-страницы (как текста, так и ссылок) к доменному имени.

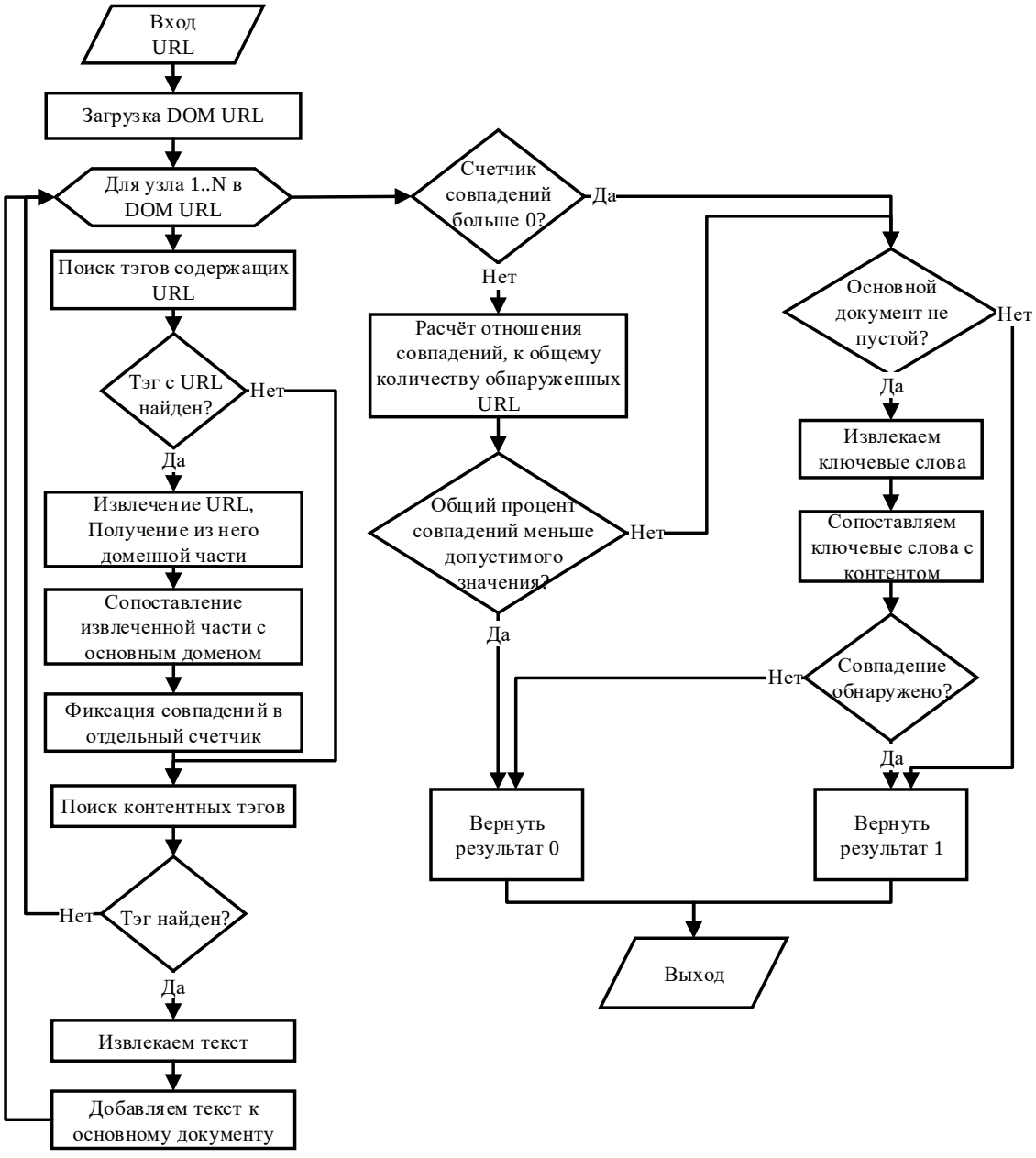


Рисунок 2.9 Алгоритм сопоставления контента страниц с доменным именем

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Загрузка DOM страницы по исследуемому URL-адресу;
3. Для каждого узла DOM исследуемого URL-адреса переходим на шаг 3.1, при завершении узлов на шаг 4;
   - 3.1 Поиск тэгов, содержащих ссылки;
   - 3.2 Если тэг обнаружен, на шаг 3.3, в противном случае на наш 3.6;
   - 3.3 Извлечение URL-адреса из тэга; как и в разделе 2.2 функции алгоритма обрезают извлеченный URL-адрес для получения только основной части;
   - 3.4 Сопоставление домена основного URL-адреса и извлеченного;
   - 3.5 Фиксация совпадений в счетчик;
   - 3.6 Поиск контентных тэгов;
   - 3.7 Если тэг обнаружен, то на шаг 3.8, иначе переход на следующий узел шаг 3;
   - 3.8 Извлекаем текст из тэга;
   - 3.9 Добавляем текст к основному документу, переход на следующий узел шаг 3;
4. Если счетчик совпадений больше нуля, то на шаг 7, иначе на шаг 5;
5. Расчет отношения совпадений доменов, к общему количеству обнаруженных URL-адресов;
6. Если общий процент совпадений меньше порогового значения 60%, то на шаг 7, в противном случае на шаг 7;
7. Если основной документ не пустой, то на шаг 8, иначе на шаг 12;
8. К основному документу сформированному из текста со всех найденных контентных тэгов применяется n-граммный метод (см. раздел 2.2), для извлечения ключевых слов, затем выбирается топ 10 релевантных ключевых слов (см. раздел 2.2);
9. Сопоставляем ключевые слова с контентом по регулярному выражению;

10. Если, обнаружено хотя бы одно совпадение ключевого слова с доменным именем, то на шаг 12, иначе на шаг 11;
11. Вернуть 0;
12. Вернуть 1.

В результате выполнения алгоритма, если контент URL-адреса содержит ссылки и текстовый контент, связанный с основным доменом, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0. Пороговым значением для п. 6 алгоритма определено 60%, значение получено при анализе эффективности и имитационных исследованиях алгоритма, представленных в 4 главе. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами, которая описана в 3 главе.

2.3.8 Алгоритм анализа истории DNS записей домена

Одним из возможных индикаторов подозрительности URL-адреса может быть частое изменение IP-адресов DNS-записи исследуемого домена. Т.е. злоумышленник может вносить изменения в записи на DNS-сервере, в случае обнаружения, что на подготовленный фишинговый ресурс нет трафика от внешних пользователей, либо вносить изменения для вновь созданного ресурса. Задача этого алгоритма обнаружить наличие частых изменений IP-адреса DNS-записи исследуемого домена, за последние N дней и посчитать K количество изменений IP-адреса, за последние N дней. В качестве источника исторических данных использованы DNS-коллекторы сети Интернет. Это сервисы, предоставляющие историческую информацию о смене IP-адресов доменов. Для получения этой информации формируется RestAPI запрос, содержащий имя исследуемой записи к коллектору suip.biz, затем, в ответ на

запрос направляется информация о датах изменения этой записи и IP-адресах, соответствующих каждому изменению. В данном случае обрабатываются «A» записи, поскольку другие ресурсные записи используются крайне редко, а значит не представляют интереса для анализа. Алгоритм представлен на рисунке 2.10.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Как и в разделе 2.2 функции алгоритма обрезают протокольную часть URL-адреса и часть, содержащую номер порта, полный путь до страницы, для получения только основной части;
3. Формируется RESTAPI запрос к коллектору для получения сведений об изменениях IP-адресов; его отправка и обработка ответа от коллектора;
4. Если IP-адреса менялись более чем 3 (K=3) раза за последние два (N=2), то на шаг 6, иначе на шаг 5;
5. Если IP-адреса менялись более чем 5 (K=5) раз за 1 (N=7) неделю, при условии несовпадения IP-адресов в 1 и 3 и 5 сменах, на шаг 6, иначе на шаг 7;
6. Вернуть 0;
7. Вернуть 1.

В результате выполнения алгоритма, если IP-адреса не менялись за последние два дня, либо смена была более чем три раза за одну неделю, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0. Значения переменных частоты и временных периодов смены IP-адресов для DNS-записей исследуемых доменных имён получены при анализе эффективности и имитационных исследованиях алгоритма, представленных в 4 главе. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами, которая описана в 3 главе.

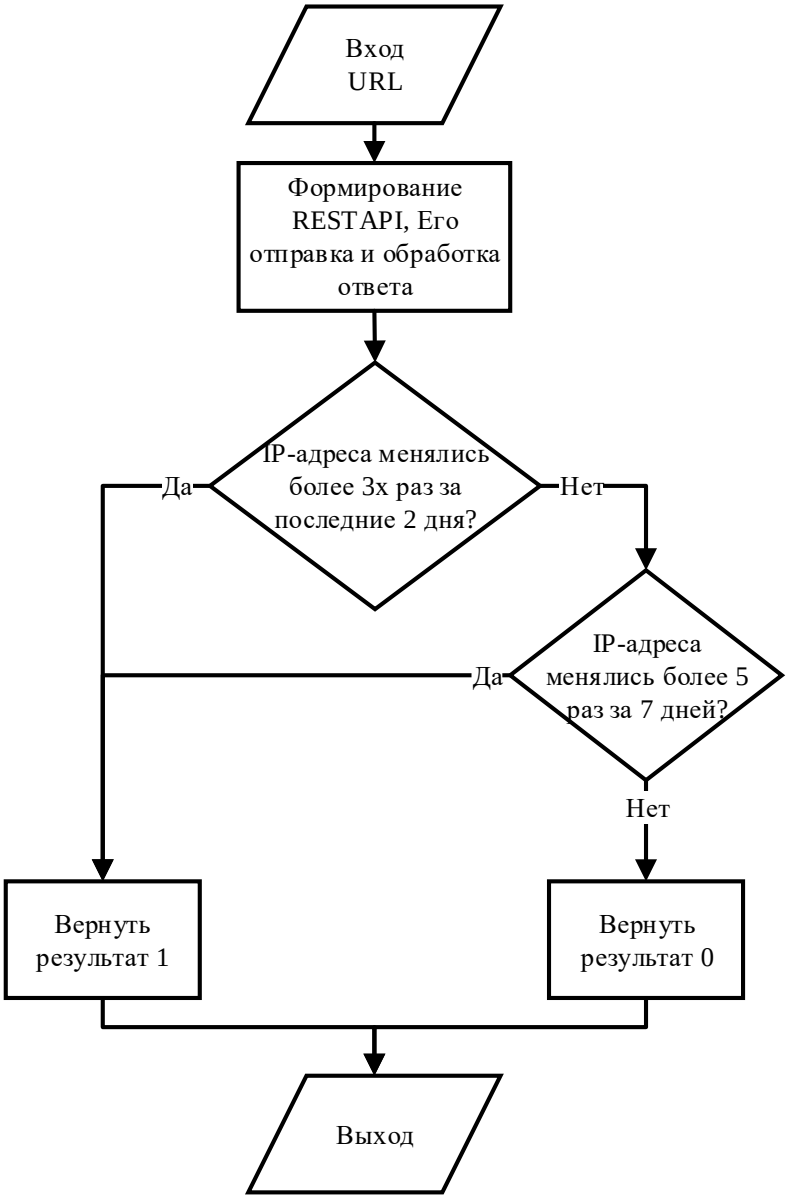


Рисунок 2.10 Алгоритм анализа истории DNS записей домена

2.3.9 Алгоритм сопоставления домена верхнего уровня с кодом страны его IP-адреса

Злоумышленники всё чаще предпринимают попытки создания фишинговых ресурсов, имитирующих национальные, государственные или иные ресурсы (например, ресурсы содержащие ПДН граждан) размещение невозможно за пределами этой страны, а значит противоречит

законодательству. При реализации таких фишинговых WEB-ресурсов, злоумышленники могут использовать домен верхнего уровня, принадлежащий одной национальной зоне, при этом сам фишинговый ресурс размещается в другой. Например, для национальных или государственных доменов “.ru” из национальной зоны РФ, фактическое размещение у хостинг-провайдера в Германии или Великобритании, считается подозрительным и может быть использовано, как индикатор опасности URL-адреса.

Алгоритм, представленный на рисунке 2.11 разработан с целью сверки страны национальных доменов и страны, где размещен исследуемый ресурс. Для получения информации о национальных доменах и соответствующих им странах, алгоритм выгружает список доменов из официального реестра IANA. В свою очередь, для получения страны для исследуемого URL-адреса, используется программа nslookup, для получения его IP-адреса и бесплатная база данных GeoLite2 при помощи RestAPI-запроса. Затем, результаты сравниваются и алгоритм возвращает результат.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Как и в разделе 2.2 функции алгоритма обрезают протокольную часть URL-адреса и часть, содержащую номер порта, полный путь до страницы, для получения верхнего уровня домена и основной части;
3. Для каждой записи в списке доменов и стран на шаг 3.1, при завершении записей в списке прекратить обработку, на шаг 11:
    3.1Сравнение домена верхнего уровня с текущей записью;
    3.2Если домен соответствует текущей записи, то передать значение страны на шаг, иначе переход к следующей записи, на шаг 3.
4. Резолвинг основной части URL-адреса в IP-адрес;
5. Формирование RestAPI запроса к сервису GeoLite2;
6. Обработка ответа от GeoLite2, получение страны в которой опубликован IP-адрес;

7. Сравнение страны домена верхнего уровня и страны, которая получена от GeoLite2;
8. Если страны идентичны, на шаг 10, иначе на шаг 9;
9. Вернуть 0;
10. Вернуть 1;
11. Выход.

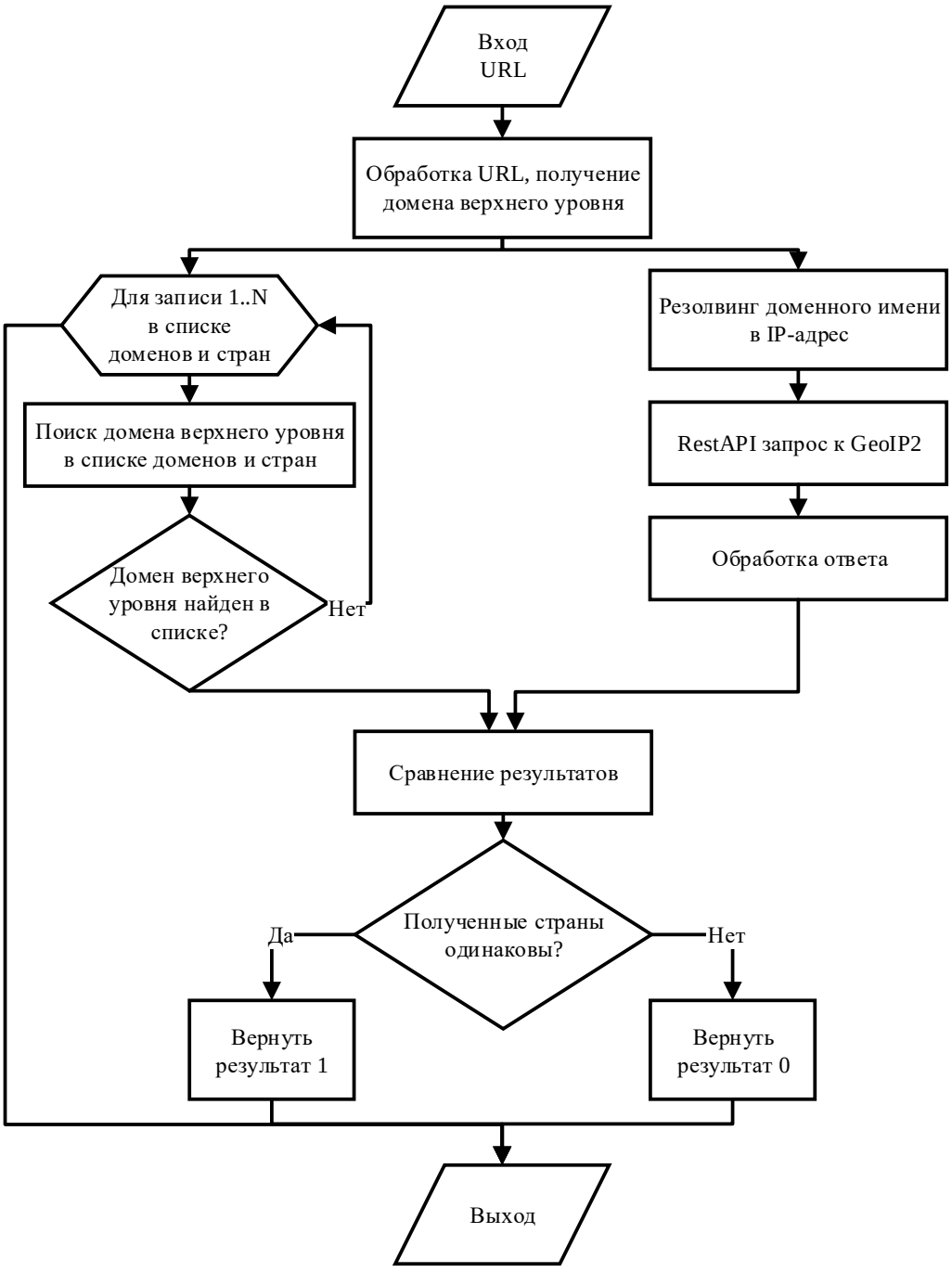


Рисунок 2.11 Блок схема алгоритма

Если страна домена совпадает со страной IP-адреса размещения, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0.

Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма представлены в 4 главе.

2.3.10 Алгоритм поиска ключевых слов в URL-адресе

Фишинговые WEB-ресурсы, часто, содержат в своем контенте фишинговые слова [98]. Как отражено в разделе 1.3, ряд исследователей сформировали статический список фишинговых ключевых слов, собранный из публичных источников, который использовали при поиске фишинговых ресурсов. Использование статического списка недостаточно эффективно, т.к. данный список актуален, только, на момент его создания для только той выборки, для которой он сделан. Следовательно, перечень ключевых слов должен содержать актуальные слова для фишинговых ресурсов, которые обнаружены и используются пользователями АСУ промышленными объектами. Алгоритм, представленный на рисунке 2.12 разработан с целью определения фишинговых ресурсов, по ключевым словам, в контенте исследуемой страницы, с механизмом обновления перечня ключевых слов. В данном случае флаг current установленный в true означает, что передаваемый URL-адреса исследуется с целью определения опасности в отношении конечного пользователя. При установке флага в false, алгоритму передаётся URL-адрес, который определён системой как фишинговый, затем анализируется контент страницы и формируется перечень ключевых слов, для добавления или изменения счетчика повторений в списке.

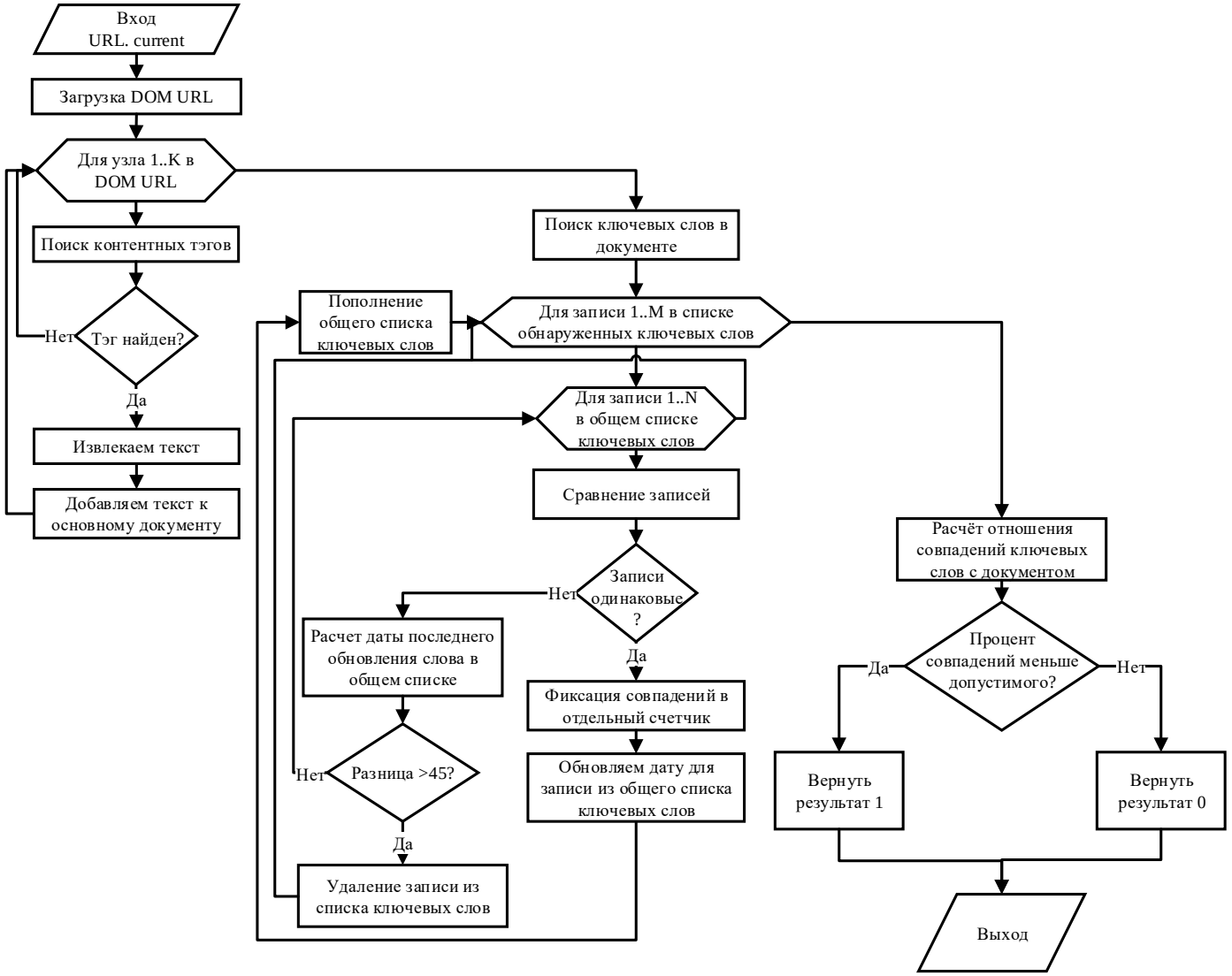


Рисунок 2.12 Алгоритм поиска ключевых слов на странице

Описание работы алгоритма:

1. На вход подаётся URL-адрес и флаг current;
2. Загрузка DOM страницы исследуемого URL-адреса;
3. Для каждого узла DOM исследуемого URL-адреса переходим на шаг 3.1, при завершении узлов на шаг 4;
   3.1 Поиск контентных тэгов;
   3.2 Если тэг обнаружен, то на шаг 3.3, иначе переход на следующий узел шаг 3;
   3.3 Извлекаем текст из тэга;
   3.4 Добавляем текст к основному документу, переход на следующий узел шаг 3;
4. К основному документу сформированному из текста со всех найденных контентных тэгов применяется n-граммный метод (см. раздел 2.2), для извлечения ключевых слов, затем выбирается топ 10 релевантных ключевых слов (см. раздел 2.2);
5. Для каждой записи из списка обнаруженных ключевых слов переходим на шаг 5.1, при завершении записей на шаг 6;
   5.1 Для каждой записи из основного списка ключевых слов переходим на шаг 5.1.1, при завершении записей на шаг 5;
      5.1.1. Поиск записи в основном документе;
      5.1.2. Если запись обнаружена в документе, на шаг 5.1.3, иначе на шаг 5.1.4;
      5.1.3. Инкрементируем счетчик совпадений и обновляем дату для записи из общего списка ключевых слов, переход на шаг 5.2;
      5.1.4. Расчет даты последнего обновления слова в общем списке, из сегодняшней даты вычитается дата текущей записи;
      5.1.5. Если разница больше порогового значения (45) дней, то на шаг 5.1.6, иначе переход на шаг, 5.1.7;
      5.1.6. Удаление записи из списка ключевых слов, переход на шаг, 5;

5.1.7. Переход на следующую запись из списка ключевых слов шаг 5.1;

5.2 Добавляем запись в общий список ключевых слов, на шаг 5;

6. Считаем процент отношения совпадений ключевых слов к документу;
7. Если процент совпадений меньше порогового, то на шаг 8, иначе на шаг 9;
8. Вернуть 0;
9. Вернуть 1.

В результате выполнения алгоритма, если в контенте исследуемого WEB-ресурса обнаружены фишинговые слова, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0.

Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма и результаты подбора порогового значения для отношения совпадений ключевых слов к документу, а также времени хранения записей ключевых фишинговых слов представлены в 4 главе.

2.3.11 Алгоритм валидации SSL/TLS сертификата

Для обеспечения защищенной передачи данных между клиентом и WEB-ресурсом, будь то фишинговый или легитимный, используются криптографические протоколы SSL и Transport Layer Security (TLS). Одним из важнейших элементов при использовании данных протоколов является использование цифровых сертификатов.

На сегодняшний день, возможны две конфигурации при их практической реализации. Первая - использование сертификата SSL/TLS, выданного

общедоверенным центром сертификации. В данном случае, ресурсу, который использует сертификат, будут доверять все браузеры и операционные системы, поскольку, общедоверенные центры сертификации добавлены в локальное хранилище операционной системы или браузера по умолчанию. Вторая - использование самоподписанного SSL/TLS сертификата. Здесь, доверия к ресурсу нет, если принудительно не разрешить переход по ссылке внутри браузера пользователя или без добавления публичного ключа сертификата или центра сертификации (выпустившего данный сертификат) в доверенные.

Как правило, сертификаты выпускаются центрами сертификации. Каждый сертификат содержит путь сертификации, который представляет из себя иерархическую структуру центров сертификации, которые принимали участие в выпуске сертификата и сам сертификат. Центры сертификации в пути сертификации, могут быть представлены как, корневые(root), если нет вышестоящих центров сертификации, либо промежуточным(intermediate) если они есть.

В свою очередь, протоколы шифрования широко применяются злоумышленниками, для шифрования данных при передаче между фишинговым ресурсом и клиентом. Несмотря на то, что современные браузеры имеют множество механизмов проверки валидности сертификатов, браузеры дают пользователю возможность принять возможные риски, что, потенциально, может привести к реализации фишинговой атаки.

С целью идентификации фишинговых ресурсов использующих невалидные SSL/TLS сертификаты и обнаружения атаки, до загрузки страницы браузером пользователя, разработан алгоритм, представленный на рисунке 2.13. Алгоритм предварительно проверяет протокол URL-адреса, затем если в URL-адресе найдены 443 порт или https протокол, тогда выполняется проверка сертификата данного URL-адреса.

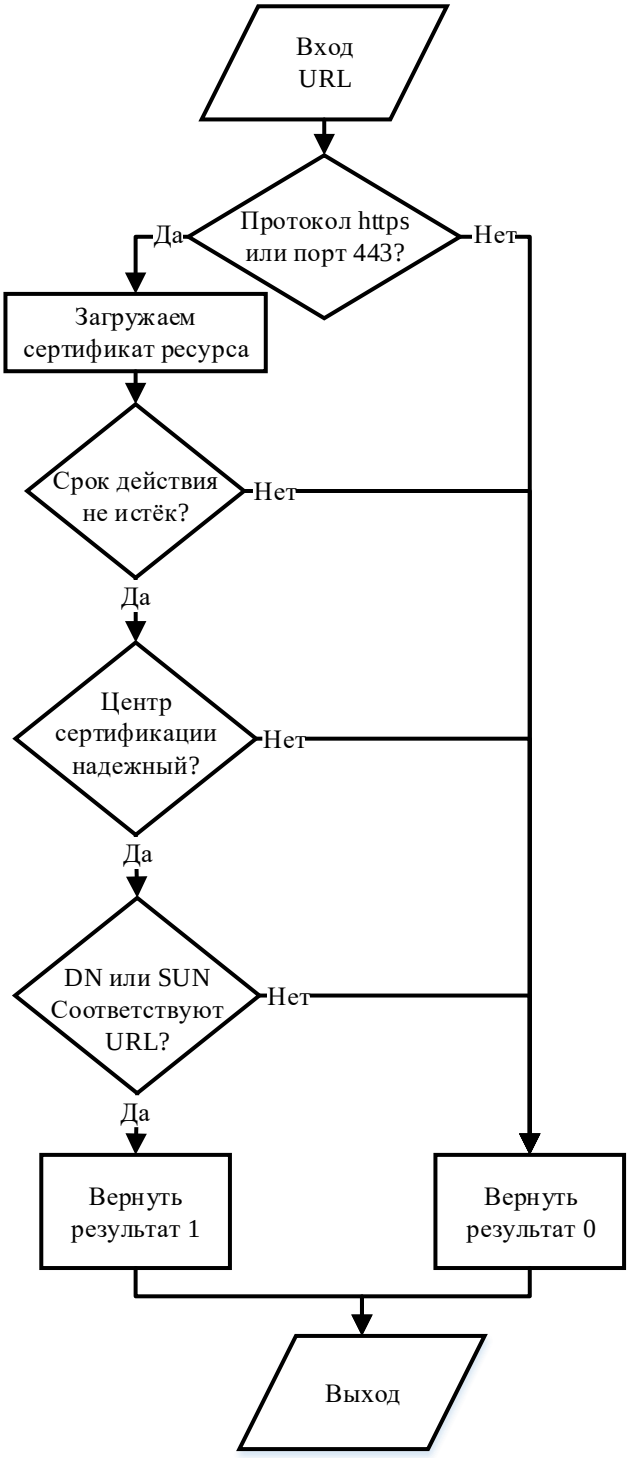


Рисунок 2.13 Алгоритм валидации SSL/TLS сертификата

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Как и в разделе 2.2 функции алгоритма обрезают протокольную часть URL-адреса и часть, содержащую номер порта, полный путь до страницы;

3. Проверяем протокольную часть страницы если обнаружен протокол https или порт 443, переход на шаг 4, в противном случае на шаг 7;
4. Загружаем сертификат ресурса;
5. Если сегодняшняя дата, в пределах срока действия сертификата, то на шаг 6, иначе на шаг 8;
6. Верифицируем сертификат командой при помощи команды openssl, если в сертификат выпущен не доверенным центром сертификации, на шаг 8, в противном случае на шаг 7;
7. Проверяем поля Distinguished Name (DN), Subject Alternative Name (SAN) и Subject сертификата, на наличие домена из исследуемого URL-адреса в этом поле. Если домен отсутствует, значит сертификат выпущен на другое/другие доменные имена, на шаг 8, иначе на шаг 9;
8. Вернуть 0;
9. Вернуть 1.

В результате выполнения алгоритма, если обнаружен протокол https или порт 443 и ресурс использует сертификат выпущенный доверенным корневым центром сертификации, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма представлены в 4 главе.

2.3.12 Алгоритм определения длины URL-адреса

Следующий индикатор опасности URL-адреса — это количество символов в URL-адресе. Это достаточно важная характеристика для детектирования фишинговых ресурсов. Злоумышленники, используют

длинные URL-адреса для того, чтобы визуально скрыть фишинговую часть адреса, в адресной строке. Соответственно, видимая часть адресной строки содержит легитимный URL-адрес, чтобы обмануть пользователя АСУ промышленными объектами, что он взаимодействует с легитимным ресурсом.

Алгоритм, представленный в разделе 1.3, использует статическое значение для допустимого количества символов, при превышении которых URL-адрес считается фишинговым. Данный алгоритм не учитывает изменение разрешения экрана ПК, при изменении которого, должно изменяться и допустимое количество символов. Т.е. существует прямая зависимость разрешения экрана и допустимого количества символов в адресной строке, которое может видеть пользователь. Поэтому, предварительно перед реализацией алгоритма, для каждого используемого разрешения экрана подсчитано количество символов, которые видны в адресной строке. Затем, составлен список «допустимых значений», состоящий из имён компьютеров, разрешения экрана, которое на них настроено, допустимого значения символов. Алгоритм, представленный на рисунке 2.14, подсчитывает количество символов в URL-адресе, в зависимости от имени ПК выбирает допустимое значения, в соответствии с предварительным перечнем, сравнивает количество символов в URL-адреса с допустимым значением.

Описание работы алгоритма:

1. На вход подаётся URL-адрес, hostname;
2. Для каждой записи в списке допустимых значений на шаг 2.1, при завершении записей в списке прекратить обработку, на шаг 8:
    - 2.1 Сравниваем hostname с текущей записью из списка допустимых значений;
    - 2.2 Если hostname найден, на шаг 2.3, иначе на шаг 2.4;
    - 2.3 Получение допустимого значения, прекратить обработку, на шаг 4;
    - 2.4 Переход к следующей записи из списка допустимых значений, на шаг 2;

3. Считаем количество символов в URL-адресе;
4. Сравниваем длину URL-адреса с допустимым значением;
5. Если длина URL-адреса, меньше допустимого значения, то на шаг 6, иначе на шаг 7;
6. Вернуть 0;
7. Вернуть 1;
8. Выход.

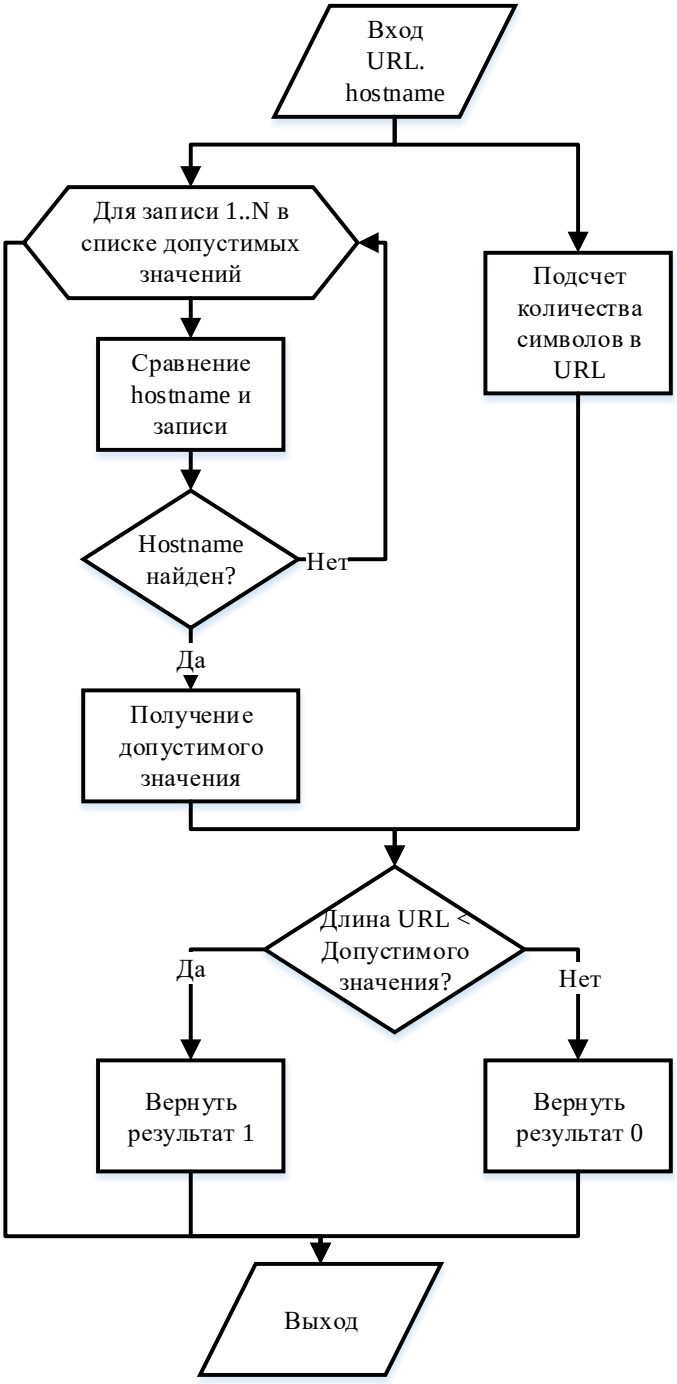


Рисунок 2.14 Алгоритм определения длины URL

В результате выполнения алгоритма, если длина URL-адреса меньше допустимого значения, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма представлены в 4 главе.

2.3.13 Алгоритм подсчёта точек в URL-адресе

Наличие специальных символов в URL-адресе, является следующим индикатором опасности URL. В данном случае речь идёт о точках в доменном имени URL. Опыт предыдущих исследователей, отмеченный в главе 1.3, показал, что количество точек в легитимных и фишинговых URL-адресов может быть различным.

Алгоритм, представленный на рисунке 2.15, подсчитывает количество точек в доменном имени, затем сравнивает полученное значение с допустимым значением.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Регулярными выражениями URL-адрес разбирается на сегменты, такие как протокол, доменное имя, порт, дополнительно извлекается домен верхнего уровня из доменного имени;
3. Для каждой записи в списке допустимых значений на шаг 3.1, при завершении записей в списке прекратить обработку, на шаг 5:
    3.1 Сравниваем домен верхнего уровня извлеченный из исследуемого URL с записью из списка исключений;
    3.2 Если домен совпадает, на шаг 3.3, иначе на шаг 3.4;

3.3 Допустимое значение извлекается из списка исключений, прекратить обработку на шаг 5;

3.4 Использовать допустимое значение «по умолчанию», равное пяти;

4. Алгоритмом поиска подстроки в строке ищем точки в доменном имени и считаем их количество;
5. Если количество точек в домене исследуемого URL-адреса меньше или равно пороговому значению, на шаг 7, иначе на шаг 6;
6. Вернуть 0;
7. Вернуть 1;
8. Выход.

При имитационных исследованиях и анализе эффективности алгоритма, описанных в 4 главе, была обнаружена зависимость между количеством точек в фишинговом домене и доменом верхнего уровня, но только при условии, что домен верхнего уровня национальный. В результате исследований, данный алгоритм был дополнен перечнем исключений, в котором описаны домены верхнего уровня и допустимое значение для них. Для доменов верхнего уровня вне этого списка, используется общее допустимое значение принятое пяти точкам, в соответствии с имитационным исследованием в 4 главе.

В результате выполнения алгоритма, если количество точек в доменном имени исследуемого ресурса меньше или равно допустимого значения, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе.

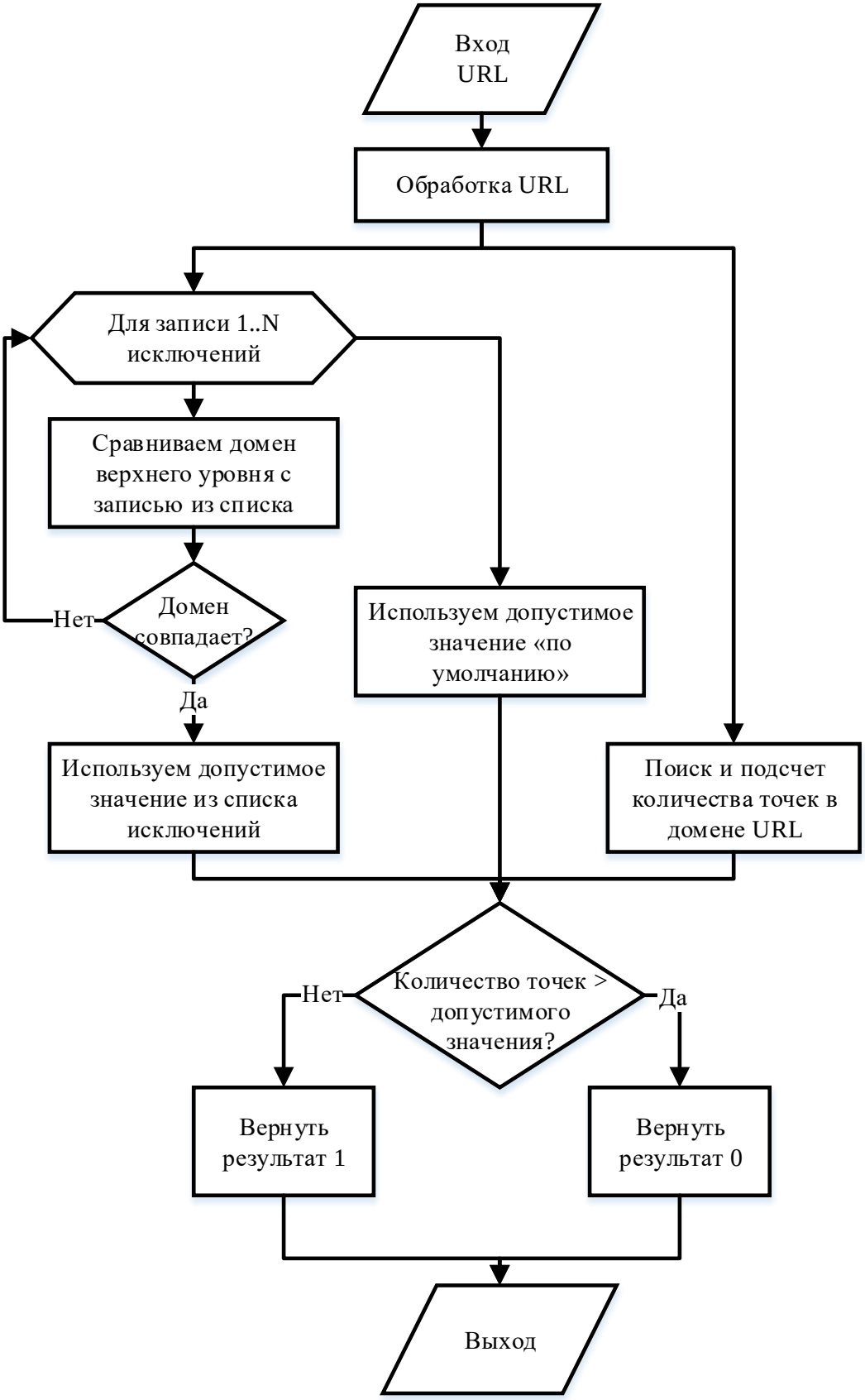


Рисунок 2.15 Алгоритм подсчета точек в URL

2.3.14 Алгоритм поиска специального символа «@» в URL-адресе

Одной из возможных реализаций фишингового URL-адреса, является наличие в нём символа «@». В строке вида https://root:password@domain.com: где, https – это протокол; «root» – логин, «password» – пароль, «domain.com» – доменное имя ресурса. На легитимных ресурсах, сегодня, данный синтаксис практически не встречается. Но злоумышленники, могут использовать этот механизм для создания URL-адреса, который будет перенаправлять пользователей на фишинговый ресурс. С целью поиска символа «@» в URL-адресе, реализован алгоритм, представленный на рисунке 2.16. В случае нахождения одного символа «@», алгоритм отбрасывает чувствительную информацию (логин и пароль) пользователя из URL-адреса, вместе с символом «@», чтобы вернуть URL-адрес без чувствительной информации. Например, для https://root:password@domain.com, результат извлечения будет таким: https://domain.com. В дальнейшем, в случае если URL-адрес признан легитимным, именно извлеченная часть будет разрешенным URL-адресом.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Поиск вхождений символа «@», в URL-адресе;
3. Если счетчик вхождений больше нуля, на шаг 4, иначе на шаг 7;
4. Если счетчик вхождений больше единицы, на шаг 8, иначе на шаг 5;
5. Отбрасываем левую часть URL-адреса, все что после протокола, до символа «@», включая сам символ;
6. Вернуть 0 и легитимный URL-адрес;
7. Вернуть 1;
8. Вернуть 0.

В результате выполнения алгоритма URL-адрес считается фишинговым: если в URL-адресе обнаружен символ «@» более 1 раза, алгоритм возвращает 0 или если в URL-адресе обнаружен символ «@» только 1 раз, алгоритм

возвращает 0, но при этом извлекает чувствительную информацию пользователя АСУ промышленными объектами и возвращает URL-адрес. В противном случае алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный. Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма представлены в 4 главе.

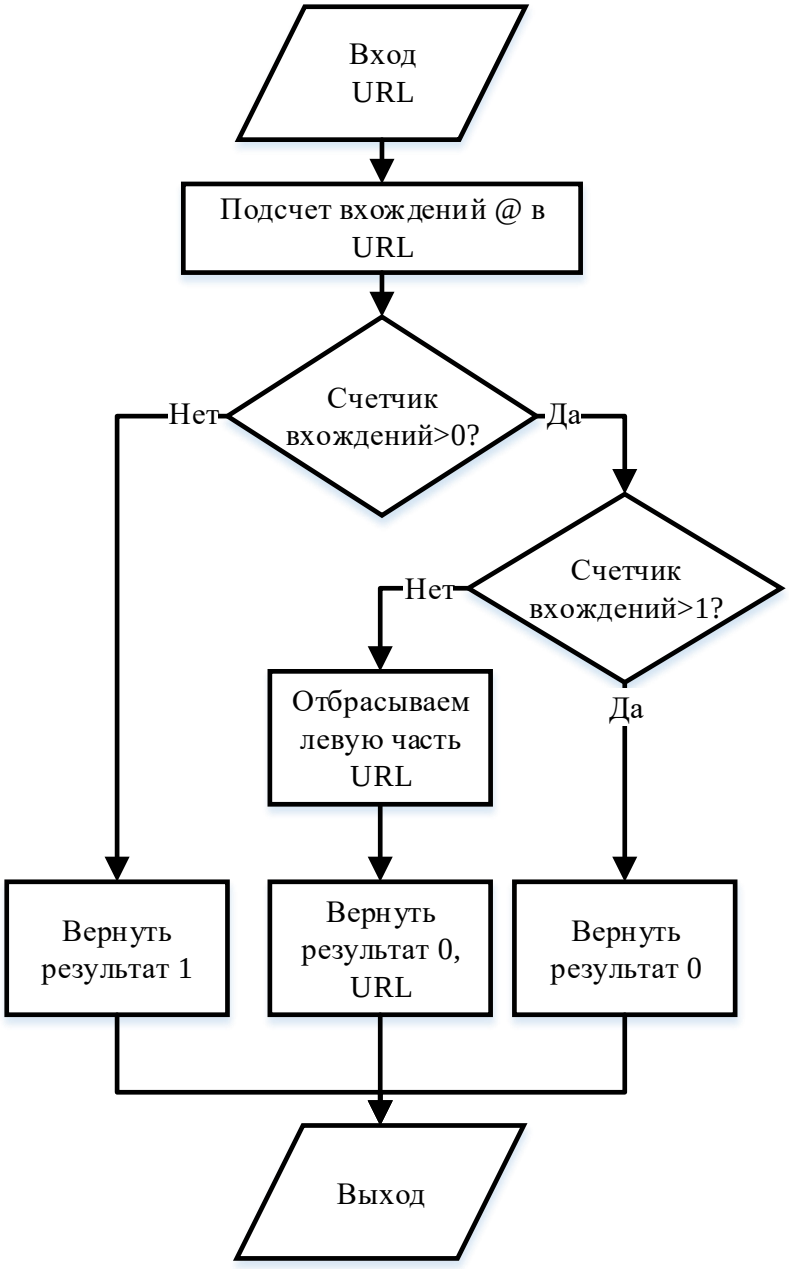


Рисунок 2.16 Алгоритм поиска спец. Символа «@» в URL.

2.3.15 Алгоритм поиска специальных символов “Слеши, протокол и порт” в URL-адресе

Как описано в разделе 2.3.13, наличие специальных символов может быть индикатором опасности URL. Например, для URL-адреса: https://click.sender.yandex.ru/l/7885/8297/2/L/RERVRklFREU4TnhjckdnWmNKaEExTTE0aEFrWTRLelFQSGtzYlNVWnpXMXg0UWdjSlJnPT06MzUxNjow/*https://phi.ru/app/?from=email_pushapp фишинговой частью является https://phi.ru/app/?from=email_pushapp, где есть и второй протокол, и двойные слеши. URL-адреса, в которых использованы более двух слешей после протокольной части, более одной протокольной части в URL-адресе и более одного порта, можно определить, как фишинговые. Алгоритм, представленный на рисунке 2.17, разработан с целью поиска вышеописанных сочетаний специальных символов. В алгоритме использован термин остальная часть, это часть URL-адреса, которая идёт после домена верхнего уровня. Например, для URL-адреса https://phi.ru/app/?from=email_pushapp, остальной частью будет «/app/?from=email_pushapp».

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Регулярными выражениями URL-адрес разбирается на сегменты, такие как протокол, доменное имя, порт и остальная часть;
3. Анализируется остальная часть на наличие сочетаний специальных символов;
4. Если в остальной части обнаружено более одного подряд слеша, переход на шаг 7, иначе на шаг 5;
5. Если в остальной части обнаружен протокол (http, https, ftp) переход на шаг 7, иначе на шаг 6;
6. Если в остальной части обнаружен порт вида “: port_number”, переход на шаг 7, иначе на шаг 8;

7. Вернуть 0;
8. Вернуть 1.

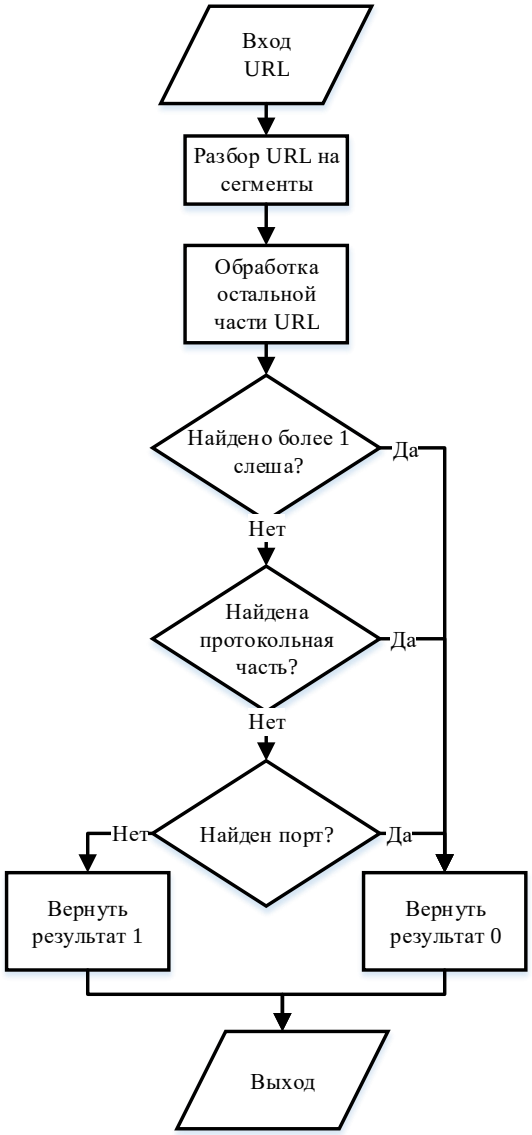


Рисунок 2.17 Алгоритм поиска спец. Символов в URL.

В результате выполнения алгоритма, если сочетание специальных символов не обнаружено, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае, если хотя бы одно из сочетаний обнаружено, ресурс – подозрительный, и возвращается 0.

Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами

и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма представлены в 4 главе.

2.3.16 Алгоритм оценки доступности URL-адреса

Подразделение Web Accessibility Initiative (WAI) группы World Wide Web Consortsium (W3C) уже с 1999 года, разрабатывает стандарты доступности WEB-контента, которые называются Web Content Accessibility Guidelines (WCAG). Основная цель стандартов - обеспечить доступность содержимого сети Интернет для всех категорий Интернет-пользователей. Выпущено 3 релиза данных стандартов: WCAG 1.0 была выпущена 05.05.1995; WCGA 2.0 была выпущена 11.12.2008; WCGA 2.1 был выпущен 05.06.2018. На основе WCAG стандартов разработано множество инструментов и в том числе, онлайн сервисов поиска ошибок на WEB-странице. Подобные инструменты, получая на вход URL-адрес и проверяют его по внутренним метрикам на наличие различных ошибок в содержимом кода WEB-страницы. В результате проверки формируется отчёт, который содержит в себе количество ошибок по типам: критичные, средней важности, низкой важности. Такие инструменты использованы в алгоритме, представленном на рисунке 2.18, с целью получения количества ошибок, найденных на ресурсе и дальнейшей их обработки. Алгоритм состоит из четырёх основных этапов.

Первый этап - проверка URL-адреса при помощи онлайн сервисов и получения количества ошибок, обнаруженных на них.

Второй этап – поиск ключевых слов на странице, отправка их в поисковую систему google, получение ответа от поисковой системы.

Третий этап обработка ответа от поисковой системы, поиск URL- адреса похожего на исследуемый и поиск ошибок на обнаруженном URL-адресе.

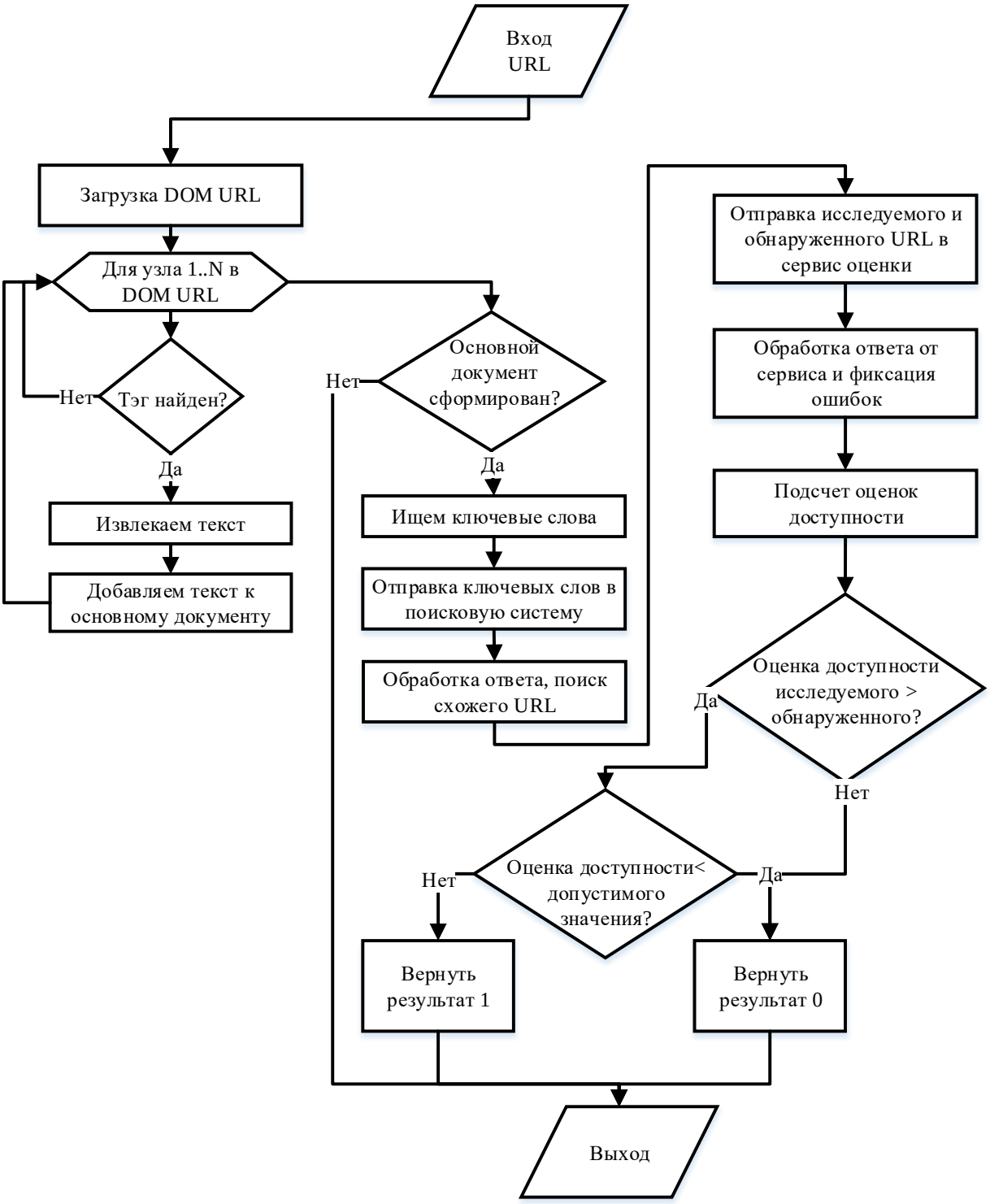


Рисунок 2.18 Алгоритм оценки доступности URL

Четвертый этап, подсчет оценки доступности исследуемого и обнаруженного URL-адресов и сравнение её с допустимым значением. Здесь, за оценку доступности (2.3) выбрано стандартное отклонение количества ошибок каждого типа и получение среднего показателя среди них.

$$AAC = \sqrt{\frac{\sum_{i=1}^{N}(X_{main,i}-X_{av})^2+(X_{find,i}-X_{av})^2}{N}}, \quad (2.3)$$

где, $X_{main,I}$ количество ошибок N-типа исследуемого ресурса, $X_{find,i}$ количество ошибок N-типа обнаруженного ресурса, X_{av} среднее значение всех ошибок, N-количество типов ошибок.

Описание работы алгоритма:

1. На вход подаётся URL-адрес;
2. Загрузка DOM страницы по исследуемому UR-адреса L;
3. Для каждого узла DOM исследуемого URL-адреса переходим на шаг 3.1, при завершении узлов на шаг 4;
   - 3.1 Поиск контентных тэгов;
   - 3.2 Если тэг обнаружен, то на шаг 3.3, иначе переход на следующий узел шаг 3;
   - 3.3 Извлекаем текст из тэга;
   - 3.4 Добавляем текст к основному документу, переход на следующий узел шаг 3;
4. Если основной документ сформирован, на шаг 5, иначе на шаг 18;
5. К основному документу сформированному из текста со всех найденных контентных тэгов применяется n-граммный метод (см. раздел 2.2), для извлечения ключевых слов, затем выбирается 4 наиболее релевантных ключевых слов (см. раздел 2.2);
6. Отправляем ключевые слова в поисковую систему google;
7. Обработка ответа, поиск обнаруженного URL-адреса, в ответе поисковой системы;
8. Отправка исследуемого и обнаруженного URL-адреса в сервис оценки посредством RestAPI;
9. Обработка ответов сервиса оценки;
10. Фиксация ошибок исследуемого URL-адреса;
11. Фиксация ошибок обнаруженного URL-адреса;

12. Подсчет оценки доступности;
13. Если оценка исследуемого больше или равна обнаруженному URL-адресу, то на шаг 14, иначе на шаг 16;
14. Если оценка доступности меньше допустимого значения, то на шаг 16, иначе на шаг 15;
15. Вернуть 0;
16. Вернуть 1;
17. Выход.

В результате выполнения алгоритма, если оценка доступности URL-адреса на шаге 3, меньше, чем допустимое значение равное восьми, алгоритм возвращает 1, что говорит о том, что исследуемый ресурс – легитимный, в противном случае ресурс – подозрительный и возвращается 0.

Практическая реализация алгоритма включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. Анализ эффективности, имитационные исследования алгоритма и анализ выбора допустимого значения представлены в 4 главе.

2.4 Модель оценки опасности ресурсов посещаемых пользователями автоматизированных систем управления промышленными объектами, основанная на методе опорных векторов

На следующем этапе требовалось решить задачу определения класса исследуемого URL-адреса, на основе данных, получаемых от алгоритмов из раздела 2.3. Исходя из имеющегося набора входных данных: два класса (фишинговый/легитимный); набор объектов и их атрибутов – это задача бинарной классификации. Поскольку в последние годы, для решения подобных задач успешно применятся метод опорных векторов, осуществляющий обучение с учителем [19] и позволяющий уменьшить

количество ошибок классификации – именно этот метод использован для решения данной задачи. Метод опорных векторов, строит бинарный классификатор при помощи функции ядра, переводя вектор атрибутов классифицируемого объекта в пространство высокой размерности и производит поиск в этом пространстве гиперплоскости с максимальным зазором, разделяющей объекты с разной классовой принадлежностью [13,14,115]. Две параллельные гиперплоскости, которые задают границы классов, строятся по сторонам разделяющей гиперплоскости на максимальном расстоянии друг от друга [19]. Чем больше расстояние между этими гиперплоскостями, тем качественнее производится классификация.

Итак, для применения метода опорных векторов первоначально определялась классифицирующая функция $F: Z \to Y$, которая устанавливает для объекта $z_i \in Z$ его класс принадлежности $y_i \in Y = \{-1; +1\}$. Для формирования классификатора использовался обучающий набор входных данных $U = \{< URL, z_1 y_1 >, \dots, < URL, z_n, y_n >\}$, где URL – адрес исследуемого ресурса, $<z_n, y_n>$ кортеж включающий в себя n-мерный вектор z_n результатов анализа алгоритмами из отрезка [0,1] и y_n метка класса, к которому принадлежит ресурс, которая может принимать значения +1 или -1, где +1 легитимный класс, -1 фишинговый. Этот набор U разбивался на обучающий и тестовый случайным образом множество раз. Проводились обучения и тестирования классификаторов, сформированных методом опорных векторов, с последующим определением лучшего классификатора, с максимальной точностью классификации. На такой каждой итерации выбирался тип функции ядра $k(z_i, z_\tau)$ - линейная, полиноминальная, радиальная базисная и сигмоидная, значения её параметров и значение признака легитимности. Здесь, параметр регуляции C (C>0) позволяет найти компромисс между уменьшением суммарной ошибки и увеличением ширины полосы. В качестве классифицирующей функции рассмотрены: линейная, полиноминальная, радиальная базисная и сигмоидная [13,14,115]. Линейная функция представлена в формуле (2.4):

$$k(z_i, z_\tau) = z_i \cdot z_\tau, \quad (2.4)$$

где, $z_i \cdot z_\tau$ – скалярное произведение между векторами признаков. Полиноминальная функция представлена в формуле (2.5):

$$k(z_i, z_\tau) = (z_i \cdot z_\tau + 1)^d, \quad (2.5)$$

где, $z_i - z_\tau$ – расстояние между двумя векторами признаков, d – степень ядра. Радиальная базисная функция представлена в формуле (2.6):

$$k(z_i, z_\tau) = \exp(-\frac{||z_i - z_\tau||^2}{2 \cdot \sigma^2}), \quad (2.6)$$

где, $||z_i - z_\tau||^2$ – квадрат расстояния между двумя векторами признаков, σ - регулируемый параметр, влияющий на производительность ядра. Сигмоидная функция представлена в формуле (17):

$$k(z_i, z_\tau) = th(k_1 + k_2 \cdot z_i \cdot z_\tau), \quad (2.7)$$

где, th – гиперболический тангенс, k_1, k_2 – параметры функции, k_1 = 1, k_2 = -1.

Далее, если классы линейно разделимы после обучения классификатора строится гиперплоскость, которая разделяет объекты z на два класса, в соответствии с формулой (2.8):

$$w \cdot z + b = 0, \quad (2.8)$$

где, b – смещение гиперплоскости относительно начала координат, w – вектор нормали к гиперплоскости, $w \cdot z$ – скалярное произведение вектора нормали и

характеристик объекта. Условие полосы разделения классов: $-1 < w \cdot z + b < 1$. Как отмечалось ранее, качество классификации напрямую зависит от ширины полосы, чем она шире, тем качественнее определяется класс объекта. Для максимизации ширины полосы, во избежание попадания в неё объектов из обучающего набора решается задача квадратичной оптимизации [13,14,115], представленная в формуле (2.9):

$$\begin{cases} w \cdot w \to \min\limits_{w,b}, \\ y_i \cdot (w \cdot z_i + b) \geq 1, i = \overline{1,n}. \end{cases} \quad (2.9)$$

Если классы линейно неразделимы, построение разделяющей гиперплоскости сводится к задаче квадратичного программирования, содержащей только двойственные переменные $\lambda_i (i = \overline{1,n})$, представленная в формуле (2.10):

$$\begin{cases} -L(\lambda) = \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{\tau=1}^{n} \lambda_i \cdot \lambda_\tau \cdot y_i \cdot y_\tau \cdot k(z_i, z_\tau) - \sum_{i=1}^{n} \lambda_i \to \min\limits_{\lambda}, \\ \sum_{i=1}^{n} \lambda_i \cdot y_i = 0, \\ 0 \leq \lambda_i \leq C, i = \overline{1,n}. \end{cases}, \quad (2.10)$$

После обучения классификатора формируются опорные векторы характеристик объектов z_i обучающего набора, соответствующие значения двойственных переменных λ_i которых отличаются от нуля. Опорные векторы располагаются близко к разделяющей гиперплоскости и несут информацию о разделении классов.

В результате обучения определялась классифицирующая функция F(z) представленная в формуле 2.11, устанавливающая класс принадлежности объекта z с соответствующей меткой:

$$F(z) = \text{sign}(\sum_{i=1}^{n} \lambda_i \cdot y_i \cdot k(z_i, z_\tau) + b), \quad (2.11)$$

где, $b = w \cdot z_i - y_i; w = \sum_{i=1}^{n} \lambda_i \cdot y_i \cdot z_i$.

В итоге, была сформирована модель для расчета класса принадлежности, которая устанавливает для объекта z его класс принадлежности и помечает его «-1» или «+1».

Практическая реализация модели включена в общую архитектуру системы повышения надежности через ИБ АСУ промышленными объектами и описана в 3 главе. В результате анализа эффективности модели и практических исследований с обучающим набором результаты которых представлены в 4 главе, была выбрана радиальная базисная функция в качестве классификатора ядра, представленная в формуле (2.6).

2.5 Выводы по главе 2

В разделах 2.1 и 2.2 представлены два алгоритма фильтрации фишинговых ресурсов в АСУ промышленными объектами, основанные на «белом листе» и «поиске форм авторизации». Алгоритмы снижают риски внесения изменений в персональные списки пользователей, актуализируют перечень записей в персональных списках пользователей, увеличивают точность определения фишинговых ресурсов за счет расширения параметров, которые их идентифицируют. Данные алгоритмы предлагается использовать в качестве первичных фильтров в методике повышения надежности через ИБ АСУ промышленными объектами.

В разделе 2.3 представлен метод алгоритмических проверок, состоящий из шестнадцати алгоритмов. Представлены потенциальные индикаторы фишинговых ресурсов характерные для АСУ промышленными объектами и соответствующие алгоритмы, позволяющие проверить их наличие отношении к исследуемому URL. Каждый из алгоритмов выполняется параллельно. Данные алгоритмы предлагается использовать в качестве вторичных фильтров

в методике повышения надежности через ИБ АСУ промышленными объектами.

В разделе 2.4 сформирована задача бинарной классификации, возникающая в результате работы метода алгоритмических проверок. Для решения задачи предложен метод опорных векторов, в качестве классификатора ядра которого выбрана радиальная базисная функция. Метод строит модель, которая позволяет классифицировать исследуемый ресурс и используется в качестве третьего компонента в методике повышения надежности через ИБ АСУ промышленными объектами.

Глава 3. Методика и программная реализация системы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

3.1 Методика повышения надежности через ИБ автоматизированных систем управления промышленными объектами

Методика повышения надежности через ИБ АСУ промышленными объектами представляет из себя – совокупность, предложенных ранее, модифицированных алгоритмов, метода и модели представленных во 2 главе и определяет основные этапы и шаги их использования, а так же содержит рекомендации по применению защитных механизмов при их реализации.

Методика повышения надежности через ИБ АСУ промышленными объектами основывается на процессе сбора URL-адресов посещаемых пользователями сетей АСУ промышленными объектами, с предварительным сопоставлением их по персональным белым спискам и поиском форм авторизации на соответствующих данным URL-адресам Интернет-страницах, с последующим применением метода алгоритмических проверок и анализа полученных результатов моделью оценки опасности ресурсов посещаемых пользователями АСУ промышленными объектами, основанной на методе опорных векторов, а так же окончательном принятии решения результатов проверки, в виде добавления легитимного URL-адреса в персональный белый список, соответствующего пользователя. Методика основана на следующих этапах:

Этап 1. Получение URL-адресов, которые пытаются посетить пользователи АСУ промышленными объектами.

Этап 2. Применение алгоритмов, метода и модели в отношении URL-адресов.

Этап 3. Анализ выходных данных, применение новой конфигурации и выполнение сопутствующих операций.

Опишем этапы методики повышения надежности через ИБ АСУ промышленными объектами более подробно.

Этап 1. Получение URL-адресов, которые пытаются посетить пользователи АСУ промышленными объектами

На первом этапе обеспечивается процесс сбора URL-адресов, которые посещают пользователи АСУ промышленными объектами с целью дальнейшей их обработки. Этап можно разделить на следующие подэтапы:

- сбор сведений о действующей инфраструктуре АСУ промышленными объектами, позволяет получить информацию о существующих механизмах обеспечения ИБ и схеме сетевых подключений конечных устройств;
- анализ действующего метода предоставления доступа в сеть Интернет, для пользователей АСУ промышленными объектами и её сервисов, с целью оценки возможности применения механизмов расшифровки трафика для получения URL-адресов;
- выбор метода сбора URL-адресов, осуществляется на основе результатов предыдущего этапа, поскольку, не во всех случаях возможно применение механизмов расшифровки трафика;
- подготовка инфраструктуры и настройка оборудования в соответствии с выбранным методом сбора, позволяет применить необходимые механизмы обеспечения ИБ, сбора URL-адресов и отправки их на следующий этап, в отношении конечных устройств, размещенных в сетях АСУ промышленными объектами.

Этап 2. Применение алгоритмов, метода и модели в отношении URL-адреса.

На данном этапе происходит анализ поступающих URL-адресов с помощью модифицированных алгоритмов, разработанного метода

алгоритмических проверок и сформированной модели оценки опасности ресурсов, посещаемых пользователями АСУ промышленными объектами, основанной на методе опорных векторов. Данный этап разделён на следующие подэтапы:

- обработка URL-адресов модифицированным алгоритмом фильтрации основанном на «белом списке», разрешение доступа, если URL-адрес обнаружен в списке;
- обработка URL-адресов модифицированным алгоритмом фильтрации основанном на поиске форм авторизации, разрешение доступа, если URL-адрес не содержит форм;
- параллельная обработка поступающих URL-адресов для анализа через механизм балансировки нагрузки и отправка URL-адресов в брокер сообщений, с целью улучшения производительности;
- параллельное чтение топиков брокера сообщений методом алгоритмических проверок, анализ полученных URL-адресов алгоритмами метода;
- отправка результатов в качестве входных данных для модели оценки опасности ресурсов, посещаемых пользователями АСУ промышленными объектами, основанной на методе опорных векторов;
- получение итоговой оценки прогнозирования.

  Этап 3. Анализ выходных данных, применение новой конфигурации и выполнение сопутствующих операций.

На данном этапе обеспечивается обработка результатов оценки URL-адреса и изменение персонального белого списка соответствующего пользователя, в случае легитимности исследованного URL-адреса. Дополнительно, на данном этапе выполняются сопутствующие операции, а именно механизм обновления белых списков и их шифрование на уровне файловой системы. Этап может быть разделен на следующие подэтапы:

- отправка результатов обработки;

- внесение изменений в конфигурацию оборудования, в части изменения белых списков;
- отдельная операция по обновлению белых списков.

Таким образом, в качестве выходных данных методики выступают URL-адреса посещаемых пользователями АСУ промышленными объектами Интернет-ресурсов, промежуточными данными – каждый конкретный URL-адрес с привязкой к соответствующему пользователю, а выходными данными результат обработки и изменение конфигурации оборудования.

К особенностям методики можно отнести существенную её зависимость от метода предоставления доступа к сети Интернет в сетях АСУ промышленными объектами и метода сбора URL-адресов на первом этапе. Поскольку, чем более распространенный метод доступа и более ёмкий метод сбора использованы, тем большее количество URL-адресов он может получить из трафика, тем самым большее количество URL-адресов анализируется. Что в свою очередь положительно влияет на работу механизмов обеспечения надежности через ИБ АСУ промышленными объектами в целом.

В данной методике допускается применение следующих методов машинного обучения: Ближайших соседей; Случайного леса; Опорных векторов (ядро - радиальная базисная функция); Опорных векторов (ядро - линейная функция); Опорных векторов (ядро - полиноминальная функция); Опорных векторов (ядро - сигмоидная функция). Выбор метода и соответствующей функции ядра может существенно влиять на качество определения фишинговых URL-адресов. Для оценки каждого метода, после проведения тестирования на одинаковых наборах входных данных, были использованы метрики оценки, представленные в разделе 1.4, а также выполнена оценка ошибок обучающего и тестовых наборов данных. В результате, в соответствии с разделом 4.3, наиболее эффективным методом машинного обучения для оценки URL-адресов, посещаемых пользователями

АСУ промышленными объектами выбран метод опорных векторов с радиальной функцией в качестве ядра.

Отдельным, но не менее важным элементом методики являются рекомендации, описанные в Приложении А. Применение которых позволяет в значительной степени снизить риски ИБ связанные либо с работой сервисов, которые встречаются в сетях АСУ промышленными объектами, либо с обслуживанием самих АСУ промышленными объектами.

3.2 Архитектура системы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

Для раскрытия предложенной методики предлагается следующая реализация системы повышения надежности через ИБ АСУ промышленными объектами, архитектура которой представлена на рисунке 3.1.

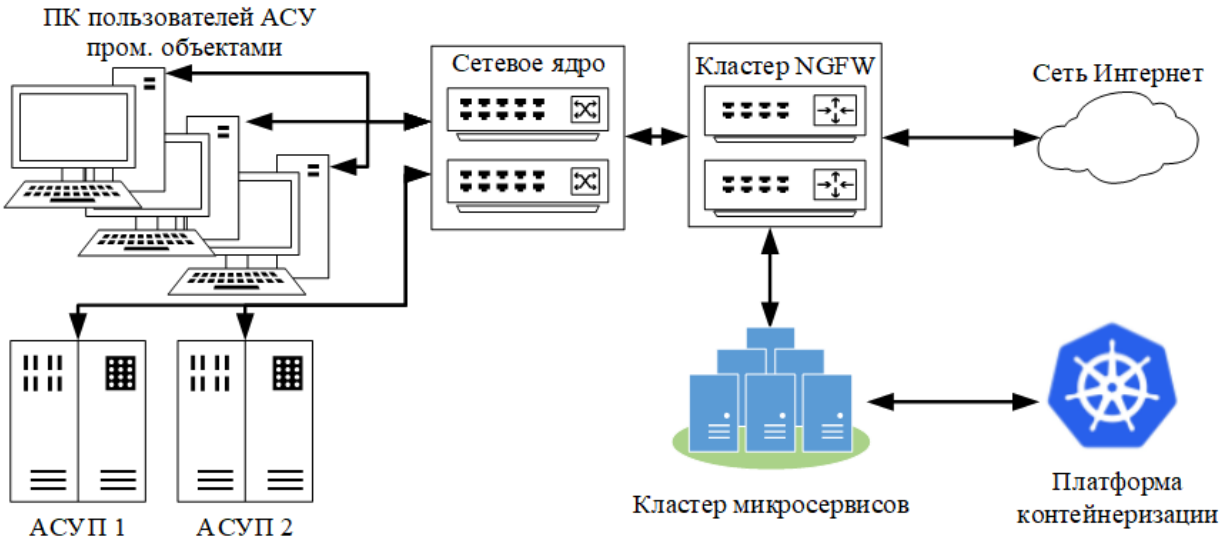


Рисунок 3.1 Общая архитектура системы

Цель системы – обеспечение безопасности пользователей АСУ промышленными объектами в части проверки посещаемых URL-адресов на наличие фишинговых признаков. В соответствии с рисунком 3.1, система состоит из трёх основных компонентов:

1. Кластер NGFW, представляет из себя платформу сетевой безопасности, состоящую из не менее двух идентичных межсетевых экранов, которые помимо традиционного функционала межсетевых экранов, содержат функционал глубокого анализа трафика Deep packet inspection (DPI), функционал контентной фильтрации и механизм SSL/TLS инспекции, которые необходимы для реализации системы.
2. Кластер микросервисов, представляет из себя среду контейнеризации, где каждый из алгоритмов, представленных во 2 главе, реализован в виде отдельного контейнера (микросервиса).
3. Платформа контейнеризации, представляет из себя платформу управления контейнерами, контейнерными нагрузками и сервисами кластера микросервисов.

Схема взаимодействия компонентов системы представлена на рисунке 3.2. Ниже приведено описание взаимодействия компонентов.

При попытке пользователем АСУ промышленными объектами посетить WEB-ресурс, весь трафик маршрутизируется в кластер NGFW. Кластер NGFW используя механизм DPI и SSL-инспекции расшифровывает трафик и получает URL-адрес, по которому пользователь АСУ промышленными объектами пытается перейти. Затем, производится попытка поиска данного URL-адреса в персональном белом списке соответствующего пользователя (в общем виде это перечень URL-адресов или имен доменов, так же перечень может включать номера портов), если URL-адрес найден, доступ разрешается, иначе запрашиваемый URL-адрес записывается в лог ноды кластера. Логи запрашиваемых ресурсов перенаправляются при помощи syslog в сторону syslog-агентов.

Каждое отправленное syslog сообщение от кластера NGFW попадает в балансировщик нагрузки, который равномерно распределяет сообщения между агентами. Агенты обрабатывают полученное сообщение, извлекают URL-адрес, который отправляется для анализа далее.

Затем, сообщения от агентов попадают в брокер сообщений. Который складывает все входящие сообщения в одну очередь. При чтении сообщений из очереди, они автоматически удаляются. Каждый такой топик брокера сообщений используется в том числе для взаимодействия между контейнерами.

Контейнер поиска форм авторизации, прочитав сообщение в топике брокера сообщений, извлекает URL-адрес из сообщения и обрабатывает его. Первоначально производится поиск данного URL-адреса в историческом списке, если проверка по данному URL-адресу проводилась ранее, все дальнейшие действия прекращаются. Если URL-адрес не проверялся ранее, производится поиск форм авторизации. Если URL-адрес не содержит форм авторизации, то URL-адрес легитимный, формируется RestAPI запрос, который добавляет данный URL-адрес в персональный белый список на кластере NGFW. В противном случае URL-адрес отправляется для параллельного анализа в отдельные топики брокера сообщений.

Каждый из шестнадцати контейнеров читает свой «входной» топик. Затем обрабатывает URL-адрес и в результате анализа возвращает значение фишинговый или легитимный. Результат отправляется в «выходной» топик, в соответствии с контейнером, который возвращает результат. Контейнер поиска формы авторизации, ждёт сообщения в выходных топиках. Как только все сообщений получены, контейнер поиска формирует N-мерный вектор и отправляет его в контейнер модели оценки опасности, основанной на методе опорных векторов. Контейнер возвращает результат классификации контейнеру поиска формы авторизации.

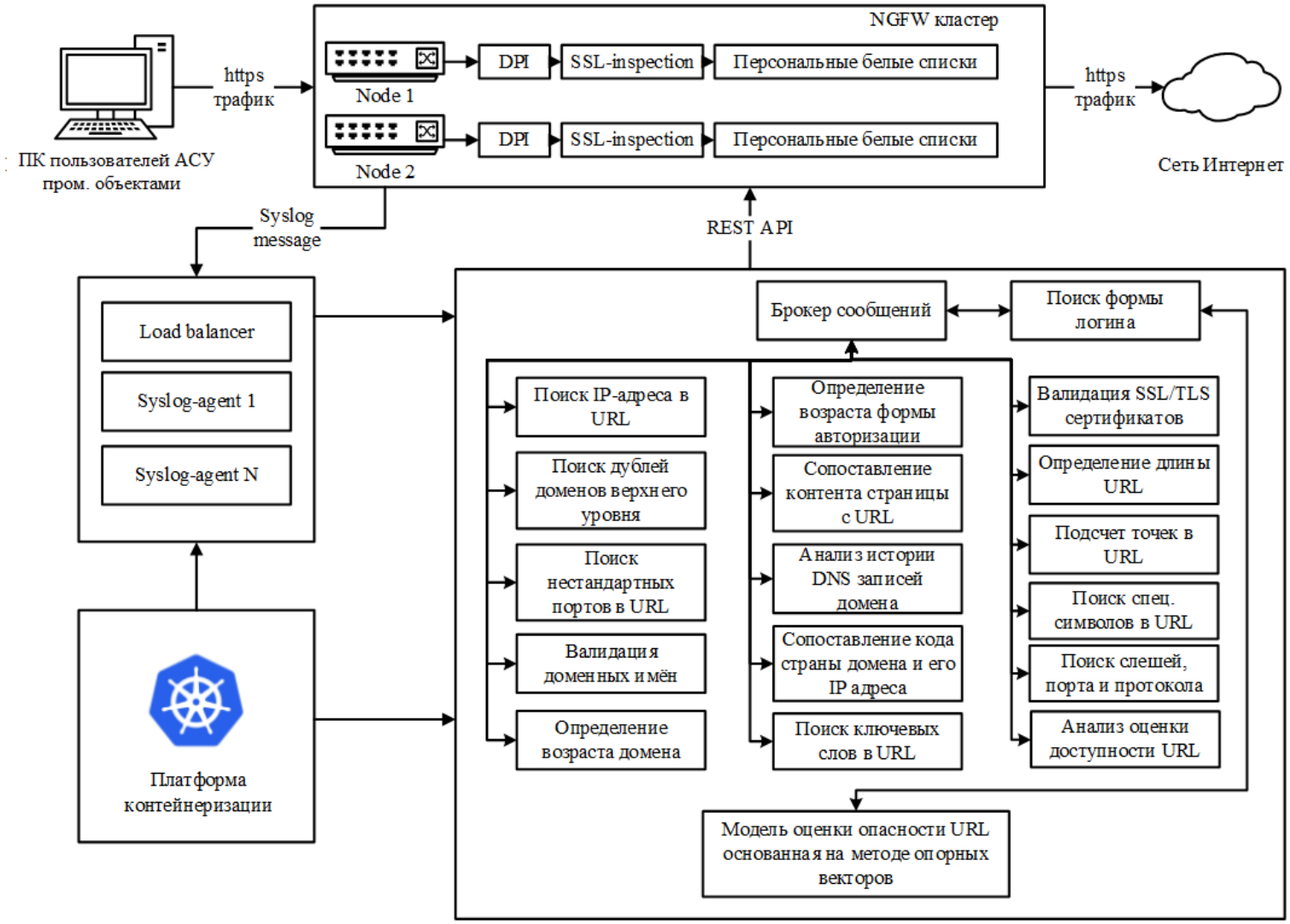


Рисунок 3.2 Схема взаимодействия компонентов системы

Если в результате анализа URL-адрес легитимный, формируется RestAPI запрос, который добавляет данный URL-адрес в персональный белый список на кластере NGFW, иначе URL-адрес записывается в исторический список вместе с датой и результатом анализа. Исторический список регулярно удаляет URL-адреса, дата которых более заданного параметра N дней от текущей даты.

3.3 Программная реализация системы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

Для запуска прототипа программной реализации системы использован специальный стенд, аппаратная архитектура стенда представлена на рисунке 3.3. При сетевом взаимодействии компьютеров пользователей АСУ промышленными объектами с сетью Интернет, создаваемый при этой активности трафик попадает в Network switch, который маршрутизирует трафик в кластер NGFW. Затем трафик обрабатывается в соответствии с схемой взаимодействия компонентов системы, представленной на рисунке 3.2.

В качестве NGFW решения использованы два Fortigate 3100D, от компании Fortinet. К причинам выбора оборудования можно отнести: низкую рыночную стоимость, в сравнении с аналогичными решениями; попадание компании Fortinet в верхний квадрат Gartner по NGFW решениям; само оборудование относится к high level классу NGFW решений и обладает требуемым функционалом, с возможностью использования API-интерфейса, для решения задач, которые должна решать система.

В основе кластера микросервисов использованы три физических сервера со следующими характеристиками: CPU Intel® Xeon GOLD 6148 2,1 GHz, RAM 64 GB, 2 TB HDD. Каждый такой сервер принято считать нодой кластера.

В качестве платформенной операционной системы использована Centos 7. Поскольку Centos обладает длительным жизненным циклом поддержки – 10 лет, что положительно сказывается на выпуск и исправление критичных уязвимостей этой ОС.

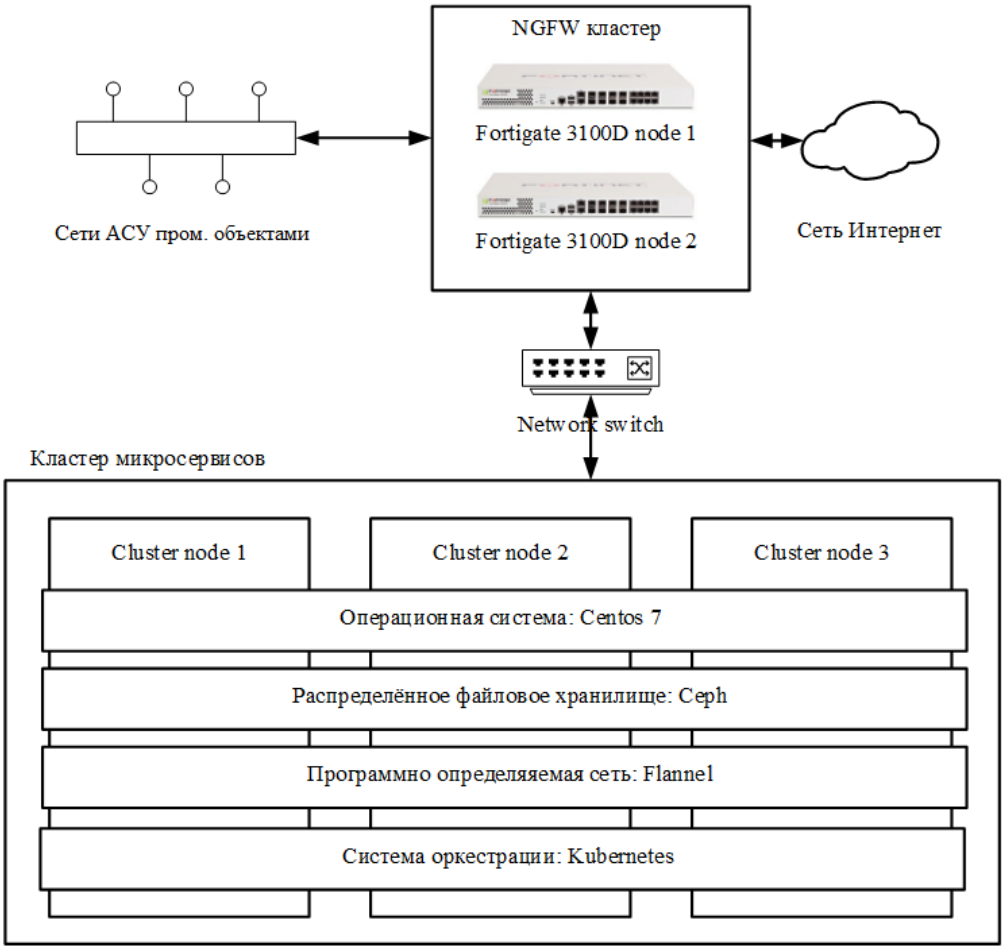


Рисунок 3.3 Аппаратная архитектура тестового стенда

В качестве платформы контейнеризации использован kubernetes. Выбор платформы обусловлен её возможностью гибко управлять docker-контейнерами и их ресурсами. В случае увеличения количества поступающих событий с URL-адресами и увеличения времени их обработки, автоматически добавляются контейнеры с syslog-агентами. В случае нехватки аппаратных ресурсов платформа контейнеризации уведомляет ответственное лицо о необходимости добавления аппаратных ресурсов. Аналогичным образом

произведена настройка управления ресурсами для контейнеров, анализирующих URL-адреса. Использование самих docker-контейнеров заключается в выделении изолированной области в рамках одной ОС и запуске в данной области сервиса. Сама изоляция контейнеров реализуется за счет механизмов Linux-систем, таких как namespace и control groups. Здесь, namespace обеспечивают изоляцию в рамках одной ОС, а control groups позволяют настроить ограничения для контейнера по использованию ресурсов хоста. Выбор docker-контейнеров изоляция сервисов друг от друга, возможность балансировки ресурсов между сервисами, гибкая горизонтальная масштабируемость сервисов и множество готовых образов, которые можно использовать для подготовки контейнеров.

Нужно отметить, что обычные жесткие диски не поддерживаются в kubernetes, поскольку взаимодействие с партициями происходит посредствам распределённых файловых хранилищ Distributed File System (DFS). Это требовалось для возможности размещения любого контейнера на любой ноде кластера, без потери данных. В качестве DFS использован Ceph.

Сетевое взаимодействие между контейнерами и локальной сетью аналогичным образом не работает без отдельной программно-определяемой сети Software Definition Network (SDN). В качестве SDN использован flannel.

Как было отмечено ранее, каждый компонент кластера микросервисов системы повышения надежности через ИБ АСУ промышленными объектами представляется в виде отдельного docker-контейнера. Каждый такой контейнер представляет из себя совокупность компонентов необходимых для работы микросервиса, таких как: среда выполнения, код реализации микросервиса и библиотеки, используемые для взаимодействия микросервисов через брокера сообщений. В качестве брокера сообщений использован RabbitMQ. Это программный брокер сообщений на основе стандарта AMQP. Брокер - является достаточно легковесным, не требователен к ресурсам и для работы с данным брокером имеется множество свободно распространяемых библиотек.

Итак, для использования контейнеров, предварительно их требовалось собрать. Сборка docker-контейнера осуществляется при помощи набора инструкций, которые записаны в Dockerfile, следуя которым docker собирает контейнер. Первый слой инструкций содержит в себе инструкции, описывающие установку базового образа. Базовый образ, в данном случае это основа, за счет которой реализуется запуск самого приложения. В качестве базового образа каждого контейнера системы был использован образ python. Следующий слой содержит инструкции, описывающие установку дополнительных модулей и библиотек. Для каждого из алгоритмов описанных в разделах 2.2, 2.3, 2.4 добавлялась библиотека pika – с целью реализации обмена сообщениями через брокер и библиотека re – с целью работы с регулярными выражениями. Дополнительно, для алгоритма из раздела 2.4, добавлена библиотека scikit-learn. Следующий слой считается «тонким», он содержит, как правило, данные, которые поддаются изменению, в данном случае это описание инструкций о запускаемом скрипте в контейнере. Ниже представлена типовая конфигурация Dockerfile, которая использовалась при создании образа.

```
#Слой описывающий инструкции базового образа
FROM python:2

#Слой описывающий инструкции по установке и обновлению дополнительных модулей
WORKDIR /opt/alg_1
RUN pip install --no-cache-dir –r pika
RUN pip install --no-cache-dir –r scikit-learn
RUN pip update

COPY ./alg_1.py /home/alg1/alg_1.py
#Слой описывающий инструкции по запуску скрипта в контейнере
USER root
RUN chown alg1:alg1 /home/alg1/alg_1.py
CMD [ "python", "./alg_1.py" ]
```

Для программной реализации представленных ранее алгоритмов, метода и модели из 2 главы, использованы методы объектно-ориентированного программирования языка Python. В качестве среды выполнения выбрана реализация Python версии 2.7.5. Выбор обусловлен наличием существующей библиотеки для анализа данных, реализующих машинного обучения scikit-learn и наличием интеграции среды выполнения Python в docker-контейнер. Поскольку Python – это интерпретируемый язык, который не компилируется, реализация кода представляется в виде отдельного файла с расширением «.py».

В результате проведения анализа URL-адреса алгоритмами из разделов 2.1-2.3 и оценки опасности ресурса по модели из раздела 2.4, тестовый стенд добавляет легитимные URL-адреса в белые списки, соответствующих пользователей. Дополнительно, фиксирует все полученные результаты анализа, будь то фишинговые или легитимные в отдельный файл вида: URL – vector ($z_1 \dots z_n$) – phishing/legitimate. Эти данные использованы в 4 главе в качестве исходных данных для оценки качества работы представленной системы повышения надежности через ИБ АСУ промышленными объектами.

3.4 Выводы по главе 3

В разделе 3.1 представлена методика повышения надежности через ИБ АСУ промышленными объектами. Методика включает в себя три основных этапа, следуя которым, возможно её применение.

В разделе 3.2 представлена архитектура системы повышения надежности через ИБ АСУ промышленными объектами. Эта архитектура состоит из трех основных компонентов: кластер NGFW, кластер микросервисов, платформа контейнеризации.

В разделе 3.2 представлена схема взаимодействия компонентов системы повышения надежности через ИБ АСУ промышленными объектами. Каждый компонент системы представлен отдельным микросервисом.

В разделе 3.3 представлена программная реализация алгоритмов и методов повышения надежности через ИБ АСУ промышленными объектами.

В приложении А приведены предложения по совершенствованию методов повышения надежности сервисов АСУ промышленными объектами при программной реализации систем повышения надежности через ИБ. Такие как:

- Совершенствование методов повышения надежности DNS сервисов АСУ промышленными объектами;
- Совершенствование методов повышения надежности (Система управления базами данных) СУБД АСУ промышленными объектами;
- Совершенствование методов повышения надежности инструментов удалённого администрирования сервисов обеспечения работоспособности АСУ промышленными объектами;
- Совершенствование методов повышения надежности при публикации сервисов АСУ промышленными объектами в сеть Интернет;
- Совершенствование методов повышения надежности сервисов АСУ промышленными объектами с использованием инструментов регулярного аудита.

Глава 4. Имитационные исследования компонентов системы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

4.1 Имитационные исследования алгоритмов фильтрации

С целью проведения имитационных исследований методики повышения надежности через ИБ АСУ промышленными объектами и анализа эффективности, как самой методики, так и отдельных алгоритмов, метода и модели представленных в разделе 2 и реализованных в разделе 3, было проведено имитационное моделирование пользовательской активности при помощи отправки сообщений содержащих URL-адреса, которые пытаются посетить пользователи, сразу в балансировщик нагрузки, минуя NGFW кластер, в соответствии со схемой представленной на рисунке 4.1.

В качестве набора данных были собраны 3257 фишинговых URL-адресов и 570 легитимных URL-адресов из сети Интернет за период с сентября по ноябрь 2018 года. Для сбора легитимных URL-адресов в качестве основных источников были выбраны следующие рейтинговые ресурсы: ahrefs.com, moz.com, alexa.com и др. При сборе легитимной подборки фокус был направлен на наиболее посещаемые ресурсы пользователями АСУ промышленными объектами. В свою очередь, для сбора фишинговых URL-адресов использованы открытые базы данных, такие как: phishtank.org и www.azsecure-data.org. Каждая запись в этом наборе уникальна и исходя из того, что фишинговые ресурсы имеют небольшое «время жизни», с целью их дальнейшего исследования, каждый ресурс из фишинговой подборки был загружен, на момент его активности. При проведении моделирования проверялась предварительно активность ресурса, в случае если ресурс был неактивен, он проверялся в автономном режиме.

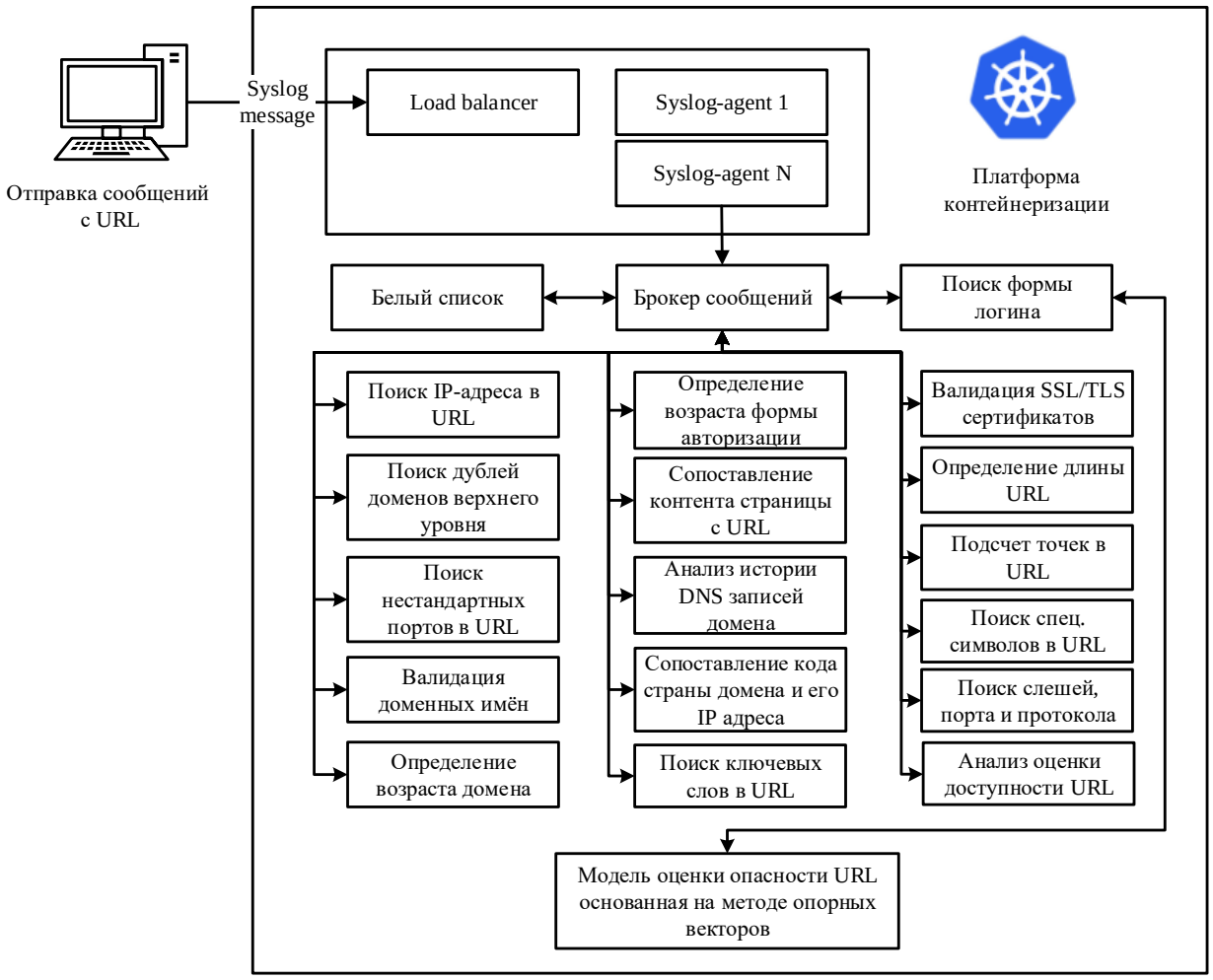


Рисунок 4.1 Схема моделирования пользовательской активности

В первую очередь были проведены исследования алгоритмов описанных в разделах 2.1 и 2.2, поскольку именно эти алгоритмы анализируют URL-адреса - первыми. Нужно отметить, что алгоритм из раздела 2.1, является мерой, позволяющей снизить нагрузку на систему в целом, при условии правильной классификации URL-адресов другими алгоритмами, ранее. Сам же, алгоритм оценивался с точки зрения улучшения производительности, за счёт механизма очистки белых списков. Как описано в разделе 2.1, параметр N (количество дней хранения) выступает в качестве регулирующего параметра, позволяющего снизить общую нагрузку на систему.

В таблице 4.1 представлены результаты исследований, где в качестве набора данных использованы URL-адреса, собранные ранее. При условии, определённого количества записей в белом списке и заданного значения N, за

которое проходит ротация записей белых списков. Дополнительно, при проведении исследований были учтены URL-адреса, в отношении которых проводились проверки, после ротации записей в белом списке. В качестве среднего значения времени обработки любого нового URL-адреса всей системой было взято значение 0.5 секунды.

Таблица 4.1 Результаты исследований
алгоритма основанного на «белом листе»

№	Записей во входном наборе данных	Записей в белом списке	N	Время обработки без повторов легитимных URL(s)	Среднее время обработки (s)	URL которые используют реже N (дни)	Время обработки URL всей системой(s)
1	500	100	10	0.45	0.0009	30	20.15
2	500	200	20	0.92	0.0018	20	11
3	500	300	30	1.44	0.0029	3	3
4	500	400	40	1.98	0.004	2	3.5
5	500	500	50	2.49	0.005	1	4
6	500	600	60	3.01	0.006	0	4.5

Таблица 4.1, показывает, что время обработки входящих URL-адресов увеличивается, при увеличении времени хранения записей в белом списке, при условии отсутствия URL-адресов, которые используются реже N дней, а значит обрабатываются повторно. В случаях же, когда такие URL-адреса присутствуют, общее время обработки уменьшается, при увеличении времени хранения. В соответствии с рисунком 4.2, в третьем исследовании значения времени минимизированы, а значит оптимальным значением N определено 30.

Исходя из полученных результатов исследований, можно предположить, что производительность метода улучшена, не менее чем в три раза, в соответствии с условиями в которых проводились исследования.

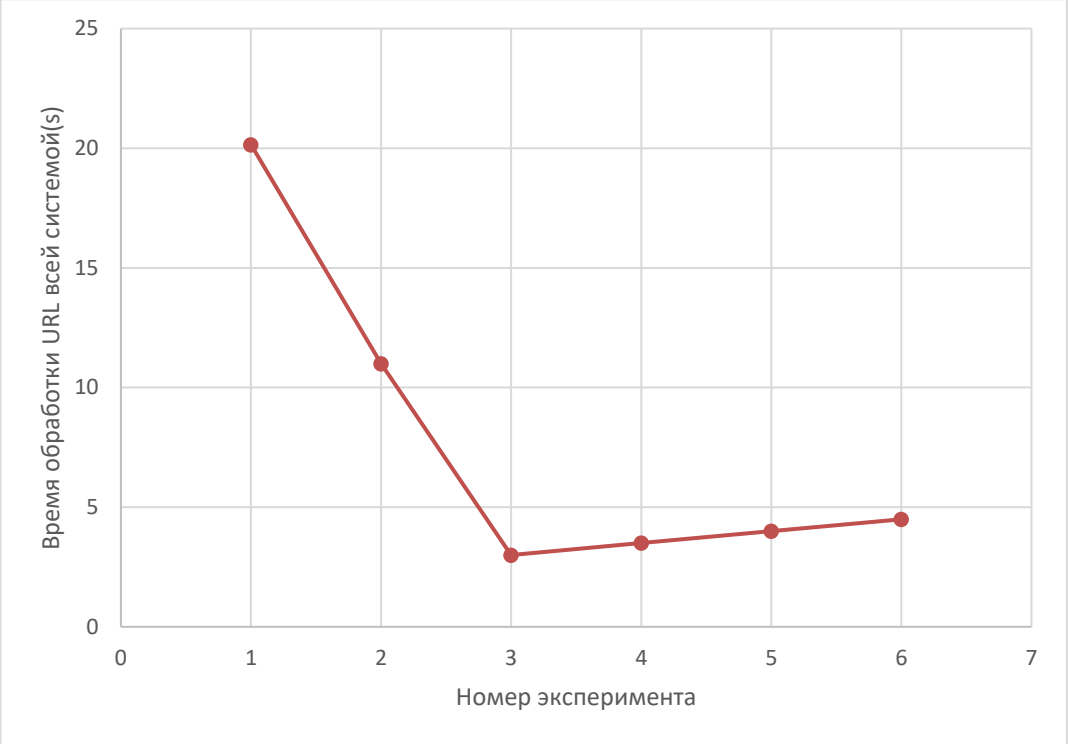


Рисунок 4.2 График времени обработки URL

В случае с алгоритмом из раздела 2.2, проводилась оценка качества нахождения форм авторизации. Для данного алгоритма было проведено два исследования. Первый, без использования n-граммного метода для подбора ключевых слов, а второй с его использованием. Результаты исследований представлены в таблице 4.2.

Таблица 4.2 Результаты исследований алгоритма, основанного на поиске формы авторизации

№	Общее количество фишинговых страниц	Общее количество легитимных станиц	Обнаружено форм среди фишинговых	Обнаружено форм среди легитимных
1	3257	560	3151	543
2	3257	560	3223	560

В соответствии с таблицей 4.2 результаты исследований показали, что использование статического словаря ключевых слов - позволяет достигнуть точности в 96.7 %, что меньше, чем в аналогичном исследовании [100], где

была достигнута точность в 98.05 %. В случае использования n-граммного метода, удалось достигнуть точности 98.95 % обнаружения форм входа.

В результате проведённых исследований, при использовании алгоритма из раздела 2.1 удалось улучшить производительность примерно в три раза, а для алгоритма из раздела 2.2, увеличить точность обнаружения форм входа на 0.9% и тем самым достичь точности 98.95%.

4.2 Имитационные исследования алгоритмов метода алгоритмических проверок

На этом этапе проводилось исследование алгоритмов метода алгоритмических проверок, описанного в разделе 2.3. Тестирование всех алгоритмов метода проводилось на том же наборе данных, что был собран в разделе 4.1. Ниже отдельно представлены результаты исследований алгоритмов из разделов 2.3.4, 2.3.5, 2.3.8, 2.3.12, 2.3.13, 2.3.15. Поскольку в данных алгоритмах использованы количественные параметры, применение которых требовало дополнительного исследования.

Для алгоритма валидации доменных имён, представленного в разделе 2.3.4, дополнительно, проводились исследования с целью поиска оптимального порогового значения оценки ресурса. При проведении исследований пороговое значение уменьшалось с шагом 0.5 и оценивалась динамика изменения метрик, описанных в разделе 1.4. Результаты исследований представлены в таблице 4.3.

В результате проведения исследований, в качестве порогового значения выбрано 5, т.к. F1-мера была максимальной для этого значения, как представлено на рисунке 4.3. При проведении исследований были обнаружены фишинговые ресурсы, размещённые на скомпрометированных легитимных ресурсах, что в свою очередь увеличивало FN.

Таблица 4.3 Результаты исследований алгоритма валидации доменных имён

№	Порог. знач.	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	10	1918	1339	85	485	0.588885	0.149123	0.850877	0.411115	0.729278
2	9.5	1960	1297	73	497	0.601781	0.12807	0.87193	0.398219	0.741021
3	9	1982	1275	60	510	0.608535	0.105263	0.894737	0.391465	0.748066
4	8.5	2027	1230	50	520	0.622352	0.087719	0.912281	0.377648	0.76003
5	8	2050	1207	48	522	0.629414	0.084211	0.915789	0.370586	0.76564
6	7.5	2099	1158	42	528	0.644458	0.073684	0.926316	0.355542	0.777695
7	7	2162	1095	42	528	0.663801	0.073684	0.926316	0.336199	0.791796
8	6.5	2457	800	39	531	0.754375	0.068421	0.931579	0.245625	0.854163
9	6	2515	742	38	532	0.772183	0.066667	0.933333	0.227817	0.865749
10	5.5	2585	672	35	535	0.793675	0.061404	0.938596	0.206325	0.879701
11	5	2638	619	35	535	0.809948	0.061404	0.938596	0.190052	0.889713
12	4.5	2587	670	33	537	0.794289	0.057895	0.942105	0.205711	0.880381

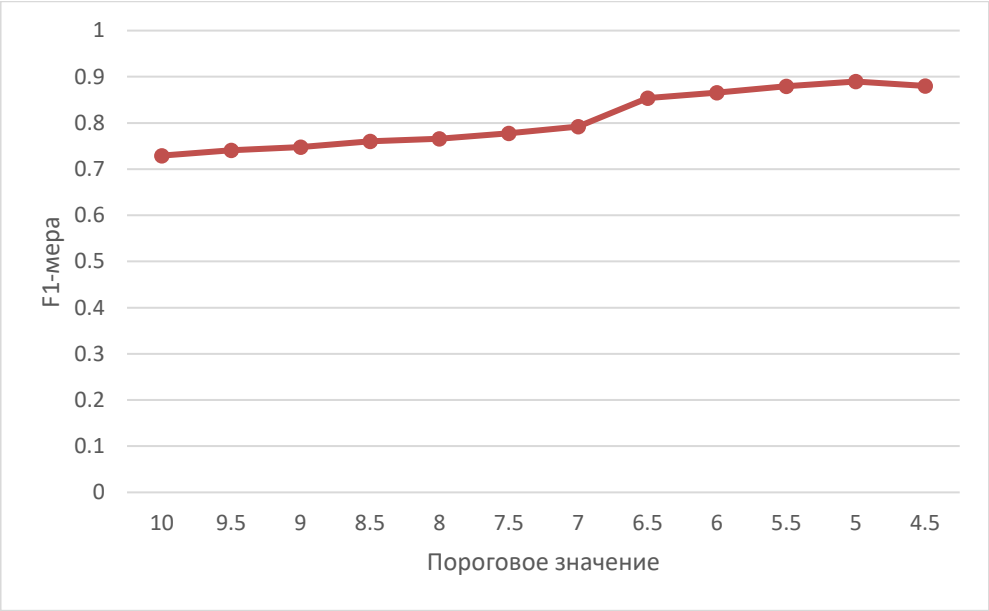


Рисунок 4.3 Динамика изменения F1-меры алгоритма валидации доменных имён

При этом, так же были обнаружены легитимные домены, имеющие низкую оценку среди всех сервисов, что увеличивало FP.

В случае с алгоритмом определения возраста доменного имени, представленного в разделе 2.3.5, дополнительно, были проведены исследования с целью определения оптимального порогового значения периода существования исследуемого ресурса. При проведении исследований, пороговое значение менялось с шагом в 1 день и оценивалась динамика изменения метрик, описанных в разделе 1.4. Результаты исследований представлены в таблице 1.1.

В результате проведения исследований, в качестве порогового значения выбрано пять дней, т.к. F1-мера была максимальной для этого порогового значения, как представлено на рисунке 4.3. При проведении исследований, были обнаружены фишинговые ресурсы, размещённые на легитимных ресурсах, которые существовали длительное время, что в свою очередь увеличивало FN. Стоит отметить, что, выбрав пять дней в качестве порогового значения, был учтён положительный опыт авторов предыдущих исследований

[97]. А также, это же пороговое значение использовано при обработке запросов от алгоритма, представленного в разделе 2.3.6.

Следующие исследования связаны с алгоритмом анализа истории DNS записей домена, представленного в разделе 2.3.8. Было выявлено, что при фишинговой атаке IP-адреса фишинговых ресурсов могут часто изменяться, либо оставаться без изменения. В случае с исследуемым набором данных, IP-адреса меняются, как только злоумышленник видит прекращение активности со стороны пользователей в сторону подготовленного ресурса в единицу времени. Отсутствие трафика приходящего к фишинговому ресурсу служит индикатором смены IP-адреса для злоумышленника. При проведении исследований были использованы, только, активные фишинговые ресурсы. Для каждого такого ресурса фиксировался период ротации IP-адресов и частота их смены. Результаты исследований представлены в таблице 4.5.

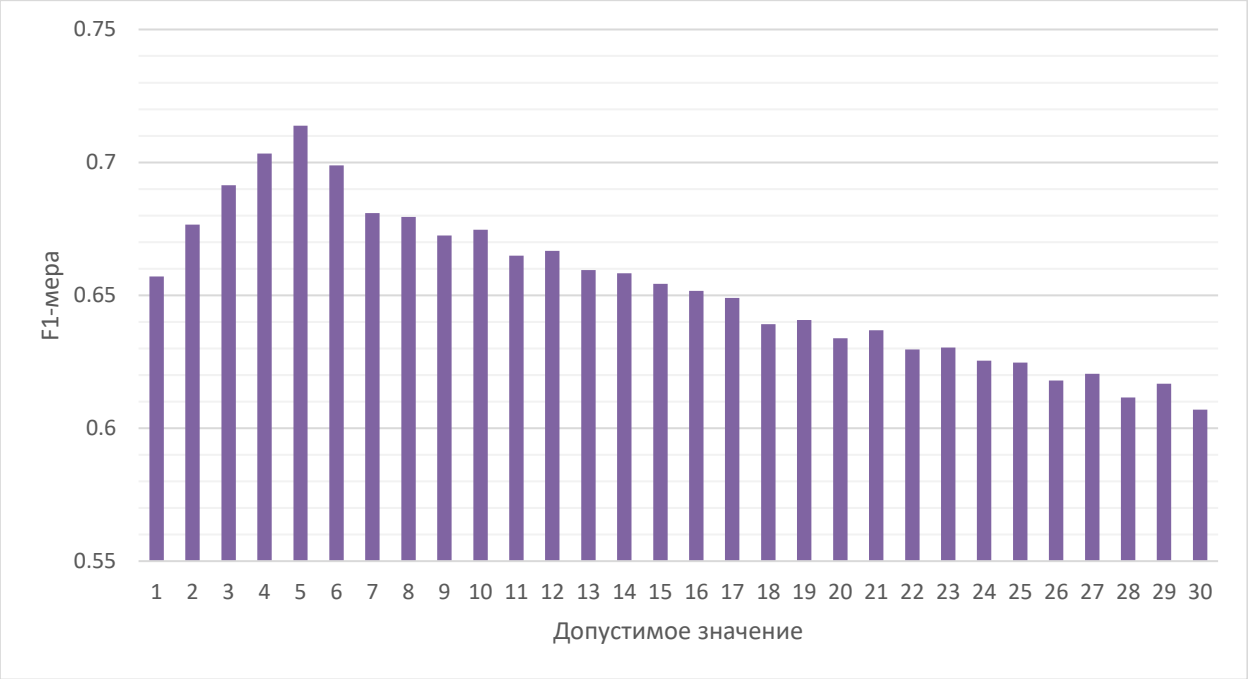


Рисунок 4.4 Динамика изменения F1-меры алгоритма определения возраста доменного имени

Таблица 4.4 Результаты исследований алгоритма определения возраста доменного имени

№	П. знач.	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	1	1687	1570	191	379	0.517961	0.33509	0.66491	0.48204	0.65706
2	2	1720	1537	107	463	0.528093	0.18772	0.81228	0.47191	0.67663
3	3	1763	1494	80	490	0.541296	0.14035	0.85965	0.4587	0.69137
4	4	1805	1452	71	499	0.554191	0.12456	0.87544	0.44581	0.70329
5	5	1827	1430	35	535	0.560946	0.0614	0.9386	0.43905	0.71381
6	6	1767	1490	33	537	0.542524	0.05789	0.94211	0.45748	0.69883
7	7	1699	1558	34	536	0.521646	0.05965	0.94035	0.47835	0.68096
8	8	1694	1563	35	535	0.520111	0.0614	0.9386	0.47989	0.6795
9	9	1668	1589	36	534	0.512128	0.06316	0.93684	0.48787	0.67245
10	10	1677	1580	37	533	0.514891	0.06491	0.93509	0.48511	0.67471
11	11	1641	1616	38	532	0.503838	0.06667	0.93333	0.49616	0.66491
12	12	1648	1609	39	531	0.505987	0.06842	0.93158	0.49401	0.66667
13	13	1622	1635	40	530	0.498004	0.07018	0.92982	0.502	0.65948
14	14	1618	1639	41	529	0.496776	0.07193	0.92807	0.50322	0.65826
15	15	1604	1653	42	528	0.492478	0.07368	0.92632	0.50752	0.65429
16	16	1595	1662	43	527	0.489714	0.07544	0.92456	0.51029	0.65169
17	17	1586	1671	44	526	0.486951	0.07719	0.92281	0.51305	0.64907
18	18	1551	1706	45	525	0.476205	0.07895	0.92105	0.52379	0.63919
19	19	1557	1700	46	524	0.478047	0.0807	0.9193	0.52195	0.64074
20	20	1533	1724	47	523	0.470679	0.08246	0.91754	0.52932	0.63386

№	П. знач.	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
21	21	1544	1713	48	522	0.474056	0.08421	0.91579	0.52594	0.63683
22	22	1519	1738	49	521	0.46638	0.08596	0.91404	0.53362	0.62964
23	23	1522	1735	50	520	0.467301	0.08772	0.91228	0.5327	0.63036
24	24	1505	1752	51	519	0.462082	0.08947	0.91053	0.53792	0.62539
25	25	1503	1754	52	518	0.461468	0.09123	0.90877	0.53853	0.62469
26	26	1480	1777	53	517	0.454406	0.09298	0.90702	0.54559	0.61795
27	27	1489	1768	54	516	0.457169	0.09474	0.90526	0.54283	0.62042
28	28	1459	1798	55	515	0.447958	0.09649	0.90351	0.55204	0.61161
29	29	1477	1780	56	514	0.453485	0.09825	0.90175	0.54652	0.6167
30	30	1444	1813	57	513	0.443353	0.1	0.9	0.55665	0.60698

Таблица 4.5 Результаты исследований

алгоритма анализа истории DNS записей

№	N	K	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	1	1	1919	1338	270	300	0.589193	0.473684	0.526316	0.410807	0.704737
2		2	1967	1290	253	317	0.60393	0.443368	0.556632	0.39607	0.718313
3		3	2018	1239	241	329	0.619589	0.422087	0.577913	0.380411	0.731744
4		4	2047	1210	229	341	0.628492	0.400982	0.599018	0.371508	0.739983
5	2	1	2104	1153	222	348	0.645993	0.389755	0.610245	0.354007	0.753695
6		2	2185	1072	210	360	0.670863	0.368708	0.631292	0.329137	0.773155
7		3	2442	815	195	565	0.74977	0.008772	0.991228	0.25023	0.856241

№	N	K	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
8		4	2420	837	154	416	0.743015	0.269725	0.730275	0.256985	0.830083
9	3	1	2296	961	159	411	0.704943	0.278895	0.721105	0.295057	0.803926
10		2	2222	1035	175	395	0.682223	0.307343	0.692657	0.317777	0.785966
11		3	2173	1084	186	384	0.667178	0.327013	0.672987	0.332822	0.773806
12		4	2092	1165	195	375	0.642309	0.341401	0.658599	0.357691	0.754744
13	4	1	2058	1199	209	361	0.63187	0.366665	0.633335	0.36813	0.745112
14		2	1988	1269	216	354	0.610378	0.378398	0.621602	0.389622	0.728114
15		3	1918	1339	230	340	0.588885	0.404129	0.595871	0.411115	0.709667
16		4	2013	1244	246	324	0.618053	0.432418	0.567582	0.381947	0.729813
17	5	1	1944	1313	271	299	0.596868	0.47566	0.52434	0.403132	0.71051
18		2	1868	1389	290	280	0.573534	0.508005	0.491995	0.426466	0.689991
19		3	1843	1414	312	258	0.565858	0.547629	0.452371	0.434142	0.68106
20		4	1872	1385	320	250	0.574762	0.561868	0.438132	0.425238	0.687065
21	6	1	1795	1462	310	260	0.551121	0.543888	0.456112	0.448879	0.669524
22		2	1836	1421	285	285	0.563709	0.499289	0.500711	0.436291	0.682833
23		3	1874	1383	272	298	0.575376	0.476322	0.523678	0.424624	0.693752
24		4	1896	1361	260	310	0.582131	0.456316	0.543684	0.417869	0.700523
25	7	1	1990	1267	254	316	0.610992	0.445365	0.554635	0.389008	0.723524
26		2	2089	1168	228	342	0.641388	0.400828	0.599172	0.358612	0.749488
27		3	2128	1129	206	364	0.653362	0.360745	0.639255	0.346638	0.761274
28		4	2242	1015	198	372	0.688364	0.347037	0.652963	0.311636	0.787107

В результате проведения исследований, в качестве периода ротации выбрано два дня, а в качестве частоты смены выбрано значение более трёх, в соответствии с графиком представленном на рисунке 4.5. Дополнительно, при проведении иследований были обнаружены ресурсы, частота смены адресации которых превышала выбранное значение как периода ротации, так и самой частоты. Поэтому, с целью увеличение точности алгоритма, было принято решение реализовать дополнительную проверку для таких ресурсов. В качестве дополнительных параметров определено значение периода ротации равное семи и частоты ротации более пяти. Тем самым, удалось увеличить TPR алгоритма на 1.7% и уменьшить FPR на 0.8%.

Рисунок 4.5 Динамика изменения F1-меры алгоритма анализа истории DNS записей домена

Для алгоритма определения длины URL-адреса, представленного в разделе 2.3.12, предварительно была составлена таблица разрешений экранов и допустимого значения количества символов в адресной строке браузера. Таблица разрешений сформирована во время исследований с каждым

отдельным разрешением экрана, результат исследований представлен в таблице 4.6. Основная отличительная особенность данного алгоритма от других — это персональное допустимое значение для каждого ПК определенного пользователя. Применение такого механизма допустимо, т.к. пользователи АСУ промышленными объектами не являются администраторами ПК, за которыми работают и, как следствие, не имеют возможности, менять конфигурацию ПК в соответствии с архитектурой АСУ промышленными объектами из главы 1.3.

Таблица 4.6 Таблица разрешений алгоритма определения длины URL

№	Разрешение	Среднее количество символов
1	800x600	65
2	1024x768	84
3	1150x824	96
4	1280x720	115
5	1280x786	115
6	1280x800	115
7	1280x960	115
8	1280x1024	115
9	1360x768	125
10	1366x768	125
11	1600x900	135
12	1600x1024	135
13	1680x1050	143
14	1920x1080	169

После того, как была сформирована таблица разрешений, представленная в таблице 4.6, были проведены два исследования: первое - со статическим значением количества символов, равным пятидесяти трём, взятым в одном из предыдущих исследований [101] и второе - с сформированной таблицей разрешений. Результаты исследований представлены в таблице 4.7

Таблица 4.7 Результаты исследований
алгоритма определения длины URL

№	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	830	2427	210	3047	0.2548	0.0645	0.9355	0.7452	0.3863
2	1139	2118	60	510	0.3497	0.1053	0.8947	0.6503	0.5112

В результате проведения исследований, при условии наличия клиентов с наиболее популярными разрешениями из таблицы 4.6, удалось достигнуть улучшения F1-меры на 0.1249, а также в целом увеличить TP и уменьшить FP. Это позволило, более точно, определять фишинговые ресурсы и адаптировать механизмы алгоритма к особенностям АСУ промышленными объектами.

Для алгоритма подсчёта точек в URL-адресе представленного в разделе 2.3.13, дополнительно проведено исследование в части наличия количества точек в фишинговых URL-адресах. При проведении исследований оценивалась работа алгоритма целиком, количество точек увеличивалось с шагом 1 начиная с 3-х. Результаты исследований представлены в таблице 1.1.

В результате проведения исследований, было выявлено, что наличие пяти и более точек в URL-адресе являются наиболее точным индикаторов фишинговых ресурсов, т.к. F1-мера была максимальной для этого значения точек, как представлено на рисунке 4.6. При проведении исследований было обнаружено, что легитимные ресурсы, так же могут использовать множество точек, следовательно это увеличивало FP и по этой же причине, F1- мера ниже 0.5.

В случае с алгоритмом оценки доступности URL-адреса, представленным в разделе 2.3.15, дополнительно, были проведены исследования, с целью определения оптимального порогового значения оценки доступности URL-адреса. При проведении исследований, пороговое значение менялось от 1 до 14, с шагом 1 и оценивалась динамика изменения метрик, описанных в разделе 1.4. Результаты исследований представлены в таблице .

Таблица 4.8 Результаты исследований алгоритма подсчёта точек в URL

№	Количество точек	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	3	963	2294	380	190	0.295671	0.666667	0.333333	0.704329	0.418696
2	4	1014	2243	350	220	0.311329	0.614035	0.385965	0.688671	0.438866
3	5	1058	2199	290	280	0.324839	0.508772	0.491228	0.675161	0.459501
4	6	1010	2247	266	304	0.310101	0.467053	0.532947	0.689899	0.445599
5	7	989	2268	242	328	0.303654	0.425018	0.574982	0.696346	0.440705
6	8	1001	2256	233	337	0.307338	0.408017	0.591983	0.692662	0.445823
7	9	840	2417	220	350	0.257906	0.385984	0.614016	0.742094	0.389158
8	10	819	2438	150	420	0.251458	0.262469	0.737531	0.748542	0.387637
9	11	856	2401	142	428	0.262819	0.249871	0.750129	0.737181	0.40231
10	12	804	2453	129	441	0.246853	0.226883	0.773117	0.753147	0.383741
11	13	781	2476	114	456	0.239791	0.199657	0.800343	0.760209	0.376222

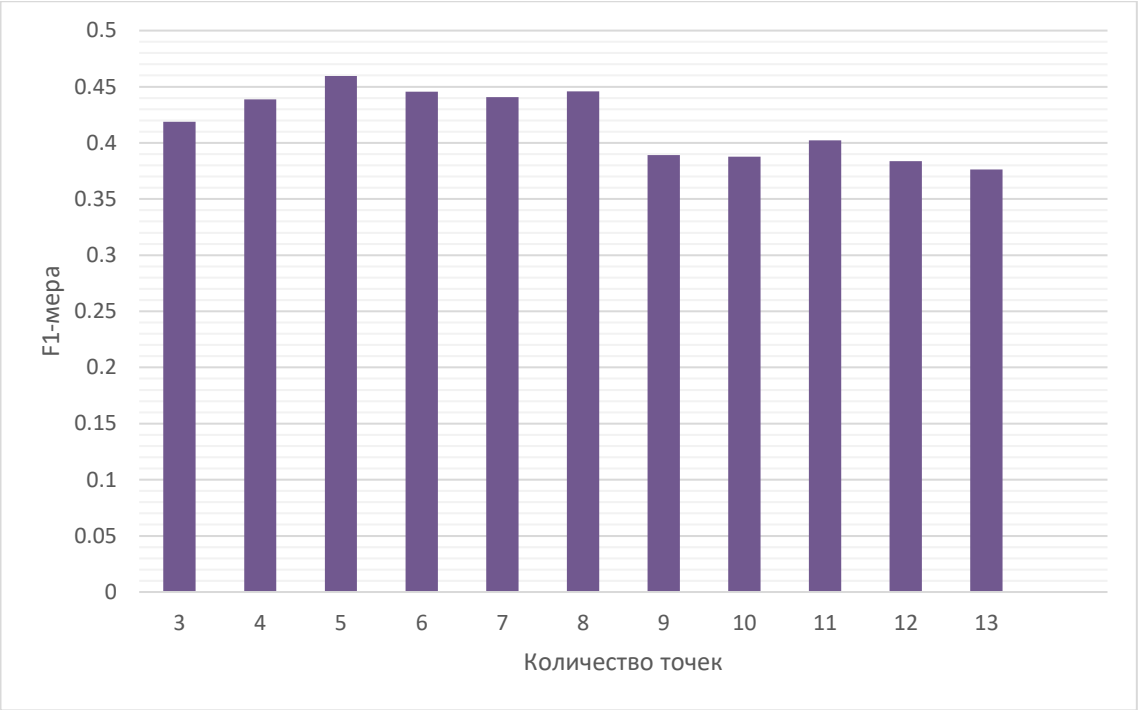


Рисунок 4.6 Динамика изменения F1-меры
алгоритма подсчёта точек в URL

В результате проведения исследований, было выявлено, что допустимое значение равное 8, с наиболее точной вероятностью позволяет идентифицировать фишинговые ресурсы, т.к. F1-мера максимальна для этого значения, как представлено на рисунке 4.7. Так же, при проведении исследования было отмечено, что система ранжирования возвращала, URL-адрес на главную страницу ресурса, а текущий исследуемый URL-адрес указывал на страницу, которая никак не связана с главной (например, на форум). Это связано с тем, что не все страницы ресурсов индексируются. С целью исключения таких ситуаций, было принято решение проверять, только, главную страницу ресурса. Это увеличило F1-меру алгоритма и уменьшило ложные срабатывания. С целью оценки работы каждого алгоритма из раздела 2.3, так же были проведены исследования с алгоритмами из остальных разделов 2.3.1, 2.3.2, 2.3.3, 2.3.6, 2.3.7, 2.3.9, 2.3.10, 2.3.11, 2.3.14, 2.3.15. В результате исследований была сформирована общая таблица метрик, представленная в таблице 4.10, номер строки соответствует порядковому

номеру каждого алгоритма из раздела 2.3. Следующим этапом сравнивалась F1-мера, MCC и BER каждого из алгоритмов, результат сравнения представлен на рисунке 4.8 и рисунке 4.9 соответственно.

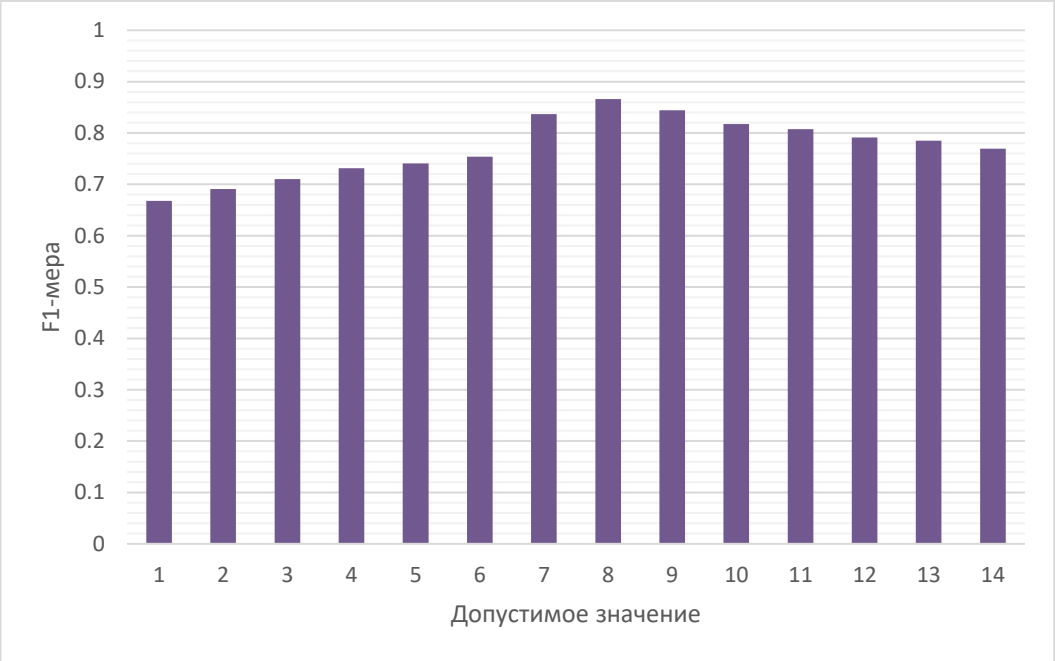


Рисунок 4.7 Динамика изменения F1-меры алгоритма оценки доступности URL

Исходя из этого стоит отметить, что алгоритм определения возраста формы авторизации, алгоритм поиска ключевых слов, алгоритм сопоставления домена верхнего уровня с кодом страны его IP-адреса и алгоритм оценки доступности URL-адреса имеют F1-меру более 0.9; MCC приближающийся к 0.7; BER менее 0.1. Что говорит о том, что данные алгоритмы имеют высокую точность, при этом количество ошибок для них минимизировано, в сравнении с остальными. Алгоритмы: поиска IP-адреса в URL-адресе, определения нестандартного номера порта в пути URL-адреса, определения длины URL-адреса, подсчёта точек в URL-адресе не обладают высокими показателями F1-меры, MCC и BER. Несмотря на это, метрика TNR для них достаточно высока, а значит, что данные алгоритмы позволяют увеличить шанс детектирования легитимных ресурсов для системы в целом.

Таблица 4.9 Результаты исследований алгоритма оценки доступности URL

№	Доп. знач.	TP	FN	FP	TN	TPR	FPR	TNR	FNR	F1
1	1	1824	1433	380	190	0.56002	0.66667	0.33333	0.43998	0.66801
2	2	1921	1336	381	189	0.58981	0.66842	0.33158	0.41019	0.691131
3	3	2004	1253	382	188	0.61529	0.67018	0.32983	0.38471	0.71026
4	4	2099	1158	383	187	0.64446	0.67193	0.32807	0.35554	0.731486
5	5	2142	1115	384	186	0.65766	0.67368	0.32632	0.34234	0.740792
6	6	2202	1055	385	185	0.67608	0.67544	0.32456	0.32392	0.753593
7	7	2622	635	386	184	0.80504	0.67719	0.32281	0.19497	0.837031
8	8	2687	570	262	308	0.82499	0.46049	0.53951	0.17501	0.865869
9	9	2563	694	250	320	0.78692	0.43839	0.56161	0.21308	0.844498
10	10	2409	848	227	343	0.73964	0.39806	0.60194	0.26036	0.817595
11	11	2341	916	200	370	0.71876	0.35029	0.64971	0.28124	0.807567
12	12	2254	1003	188	382	0.69205	0.33067	0.66933	0.30796	0.790949
13	13	2217	1040	175	395	0.68069	0.30620	0.6938	0.31931	0.784982
14	14	2141	1116	169	401	0.65735	0.29641	0.7036	0.34265	0.769182

Таблица 4.10 Общая таблица оценок

№	TP	FN	FP	TN	TPR	FPR	TNR	FNR	Precision	Recall	F1	MCC	BER
1	1074	2183	0	570	0.3	0	1	0.67025	1	0.32975	0.49596	0.26129	0.33512
2	2540	717	0	570	0.8	0	1	0.22014	1	0.77986	0.87632	0.5877	0.11007
3	846	2411	12	558	0.3	0	0.97895	0.74025	0.986014	0.25975	0.41118	0.20377	0.38065
4	2638	619	35	535	0.8	0.1	0.9386	0.19005	0.9869061	0.80995	0.88971	0.58071	0.12573
5	1777	1480	5	565	0.5	0	0.99123	0.45441	0.9971942	0.54559	0.7053	0.38316	0.23159
6	2833	424	5	565	0.9	0	0.99123	0.13018	0.9982382	0.86982	0.92961	0.70028	0.06948
7	2540	717	30	540	0.8	0.1	0.94737	0.22014	0.9883268	0.77986	0.8718	0.55129	0.13639
8	2442	815	5	565	0.7	0	0.99123	0.25023	0.9979567	0.74977	0.85624	0.54942	0.1295
9	2670	587	40	530	0.8	0.1	0.92982	0.18023	0.9852399	0.81977	0.89492	0.58703	0.1252
10	2801	456	8	562	0.9	0	0.98596	0.14001	0.997152	0.85999	0.92351	0.68163	0.07702
11	1270	1987	21	549	0.4	0	0.96316	0.61007	0.9837335	0.38993	0.55849	0.26588	0.32346
12	1139	2118	60	510	0.3	0.1	0.89474	0.65029	0.9499583	0.34971	0.51122	0.18763	0.37778
13	1058	2199	2	568	0.3	0	0.99649	0.67516	0.9981132	0.32484	0.49016	0.25565	0.33933
14	1205	2052	0	570	0.4	0	1	0.63003	1	0.36997	0.54012	0.2836	0.31501
15	2475	782	14	556	0.8	0	0.97544	0.2401	0.9943753	0.7599	0.86147	0.54903	0.13233
16	2687	570	9	561	0.8	0	0.98421	0.17501	0.9966617	0.82499	0.90274	0.63141	0.0954

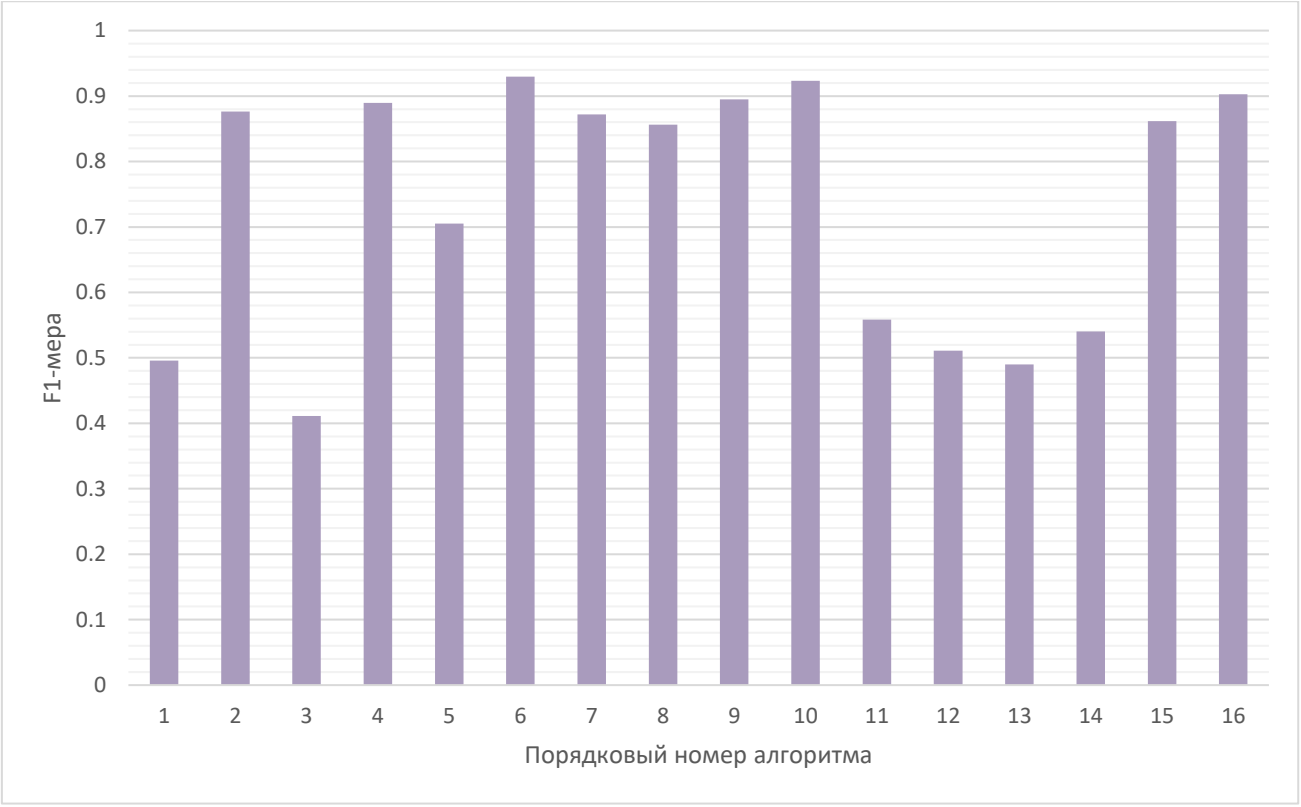


Рисунок 4.8 Сравнение метрики F1-мера каждого отдельного алгоритма

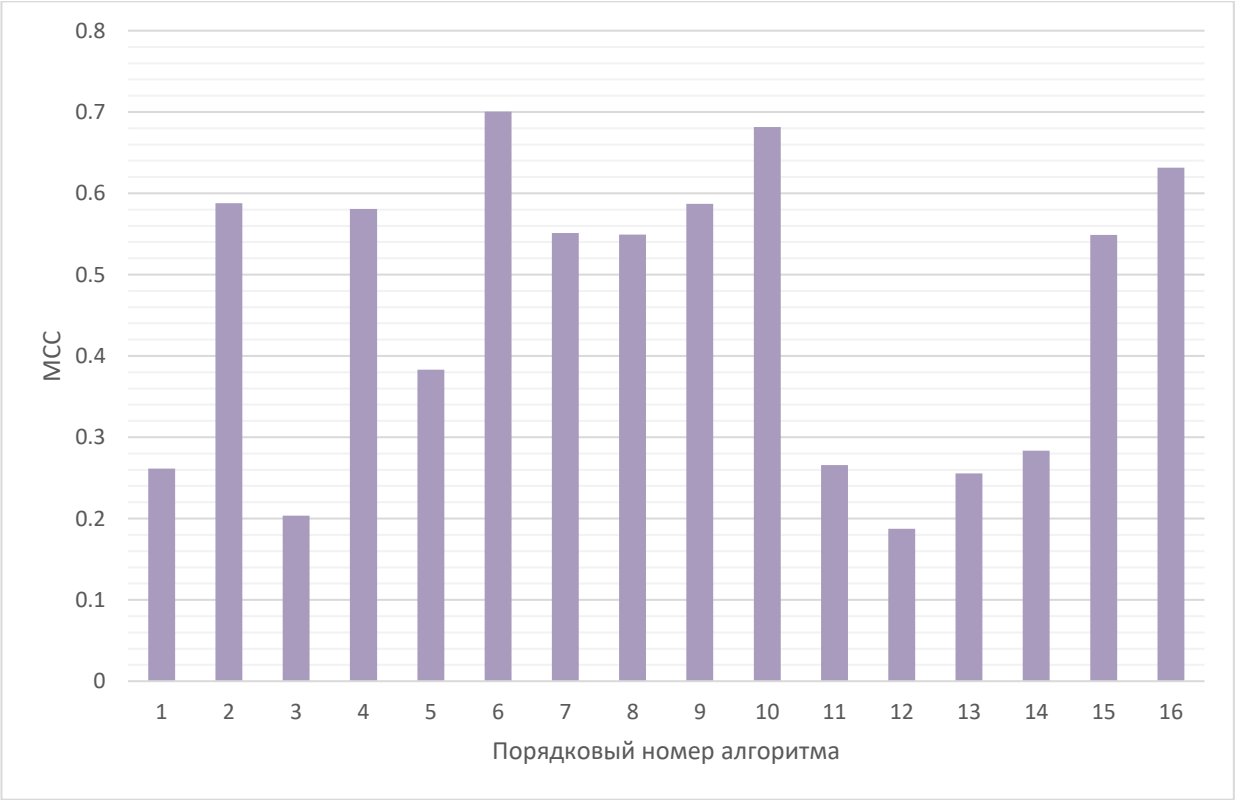


Рисунок 4.9 Сравнение метрики MCC между всеми алгоритмами

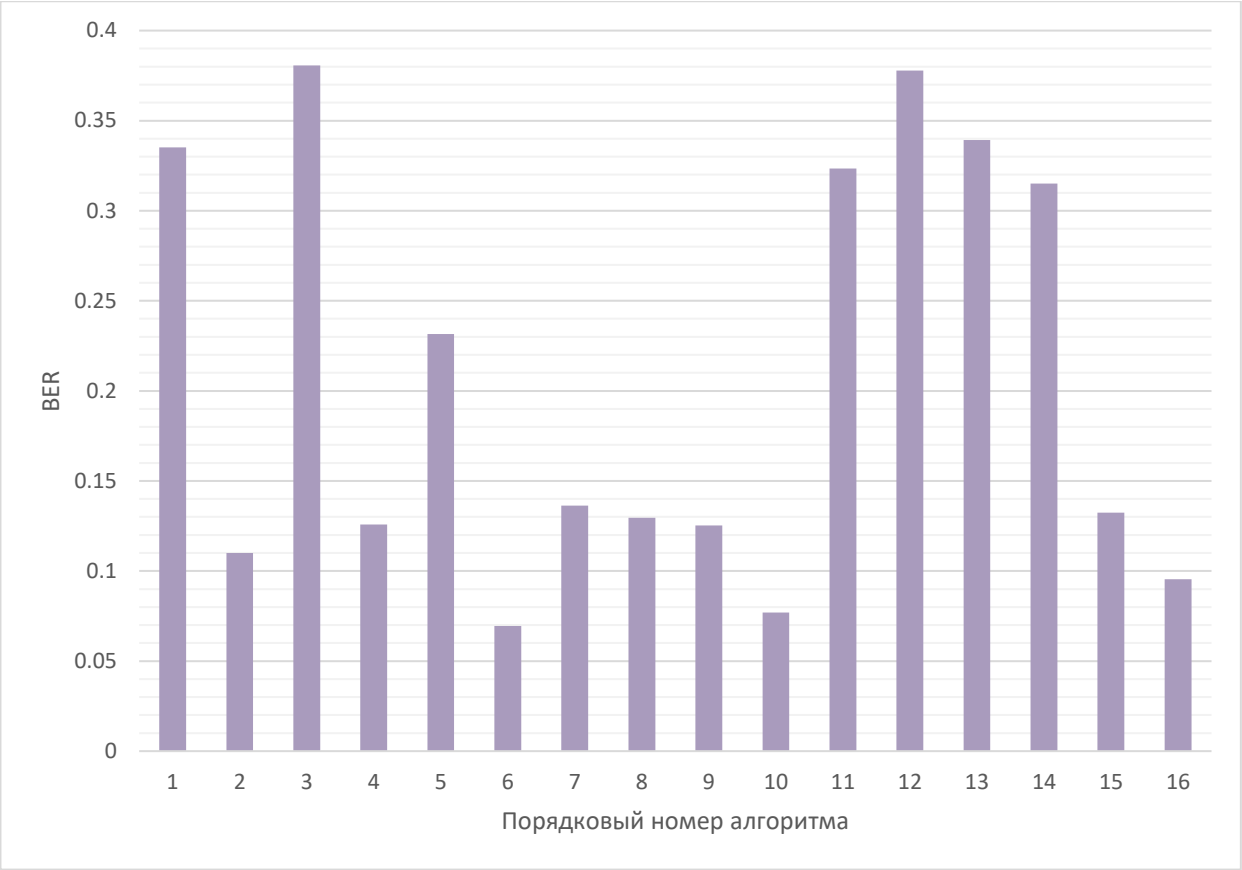


Рисунок 4.10 Сравнение метрики BER между всеми алгоритмами

4.3 Имитационные исследования модели оценки опасности ресурсов и методики повышения надежности через ИБ автоматизированных систем управления промышленными объектами

На этом этапе проводились имитационные исследования модели оценки опасности ресурсов, основанной на методе опорных векторов описанной в разделе 2.4. Затем, проводилась оценка качества работы каждого из алгоритмов, описанных в разделе 2.3.

Для исследования модели оценки опасности ресурсов, основанной на методе опорных векторов, использовались данные, собранные в разделе 4.2. Первоначально, данные были обработаны и проанализированы на предмет отсутствующих значений и необходимости нормализации значений. Для

отсутствующих значений был проведен повторный анализ, в случае отсутствия успеха, данные значения были исключены из набора с целью уменьшения ошибок классификации. Нормализация значений признаков, в данном наборе, не требовалась, т.к. они бинарные.

Как уже было отмечено ранее, модель строилась на обучающем наборе, а её качество проверялось на тестовом наборе. В данном случае, был использован метод train_test_split библиотеки sklearn.model_selection, для разделения существующего набора на обучающий и тестовый. Разделение было выполнено в 3-х вариациях 30/70 в пользу обучающего набора, 50/50 и 70/30 в пользу тестового набора.

Затем, при помощи полученных данных, были реализованы обучены и исследованы модели по следующим методам:

- метод ближайших соседей (kNN);
- метод опорных векторов с различными функциями ядра (линейная, полиноминальная, радиальная базисная и сигмоидная);
- метод случайного леса (random forest).

Результаты анализа качества моделей на каждую вариацию обучающего набора представлены в таблице 4.11.

Таблица 4.11 Результаты анализа качества моделей

Метод	**Набор тестовый/ обучающий**	**Ошибки обучающего набора, %**	**Ошибки тестового набора, %**
Ближайших соседей	30/70	0.14264	0.13489
	50/50	0.14688	0.13479
	70/30	0.14895	0.13811
Опорных векторов (радиальная базисная функция)	30/70	0.14227	0.13315
	50/50	0.14375	0.13480
	70/30	0.14459	0.13923
Опорных векторов (линейная функция)	30/70	0.14600	0.13490
	50/50	0.14950	0.13584
	70/30	0.14982	0.1396
	30/70	0.14600	0.13490

Метод	Набор тестовый/ обучающий	Ошибки обучающего набора, %	Ошибки тестового набора, %
Опорных векторов (полиноминальная функция)	50/50	0.14950	0.13584
	70/30	0.14982	0.13960
Опорных векторов (сигмоидная функция)	30/70	0.14600	0.13490
	50/50	0.14950	0.13584
	70/30	0.14982	0.13960
Случайного леса	30/70	0.00672	0.14186
	50/50	0.00366	0.13949
	70/30	0.00174	0.14371

В результате анализа моделей, наибольшее внимание было уделено ошибкам тестового набора, с целью увеличения точности определения класса для новых объектов, которые будет исследовать модель в будущем. Наименьшее количество ошибок, в части тестового набора, выявлено у метода опорных векторов с радиальной функцией. Несмотря на то, что метод случайного леса показал минимальное количество ошибок на обучающем наборе, ошибки на тестовом наборе достаточно велики. Так же, стоит отметить, что был получен одинаковый процент ошибок для методов: опорных векторов с линейным, полиноминальным и сигмоидным ядром. Исходя из вышеизложенного, метод опорных векторов с радиальной функцией качестве ядра, предлагается как основной метод для построения модели. На следующем этапе проводилась оценка качества работы каждого из алгоритмов из раздела 2.3. В качестве основной метрики оценки использована площадь под кривой ошибок (AUC) описанная в разделе 1.3. Как было отмечено ранее, чем ближе эта метрика к единице, тем лучше алгоритм выполняет свою задачу – определяет фишинговые URL. После анализа URL-адреса, алгоритмами формируются результаты анализа, и они сравниваются с меткой класса (для обучающего набора) или с результатом прогнозирования классификатора (для тестового данных). Далее результаты помечаются знаками “+” и “-“, для положительных и отрицательных результатов

соответственно, и группируются. В результате получается последовательность отсортированных в порядке убывания результатов.

Результаты оценки качества работы каждого из алгоритмов представлены на рисунке 4.11. Нужно отметить, что алгоритмы определения возраста формы авторизации, поиска ключевых слов, оценки доступности URL-адреса, сопоставления домена верхнего уровня с кодом страны и валидации доменных имён обладают высоким AUC более 0.8. Это позволило увеличить точность работы системы в целом. Нужно отметить, что AUC для алгоритмов поиска IP-адреса в URL-адресе, подсчета точек в URL-адресе и определения нестандартного номера порта в пути URL-адреса менее 0.66, что само по себе указывает на уменьшение вероятности качественного прогнозирования с использованием данных алгоритмов. Несмотря на данный факт, применение этих алгоритмов снижает риски, возникающие при использовании злоумышленниками большого количества точек и использование нестандартного номера порта в пути URL-адреса.

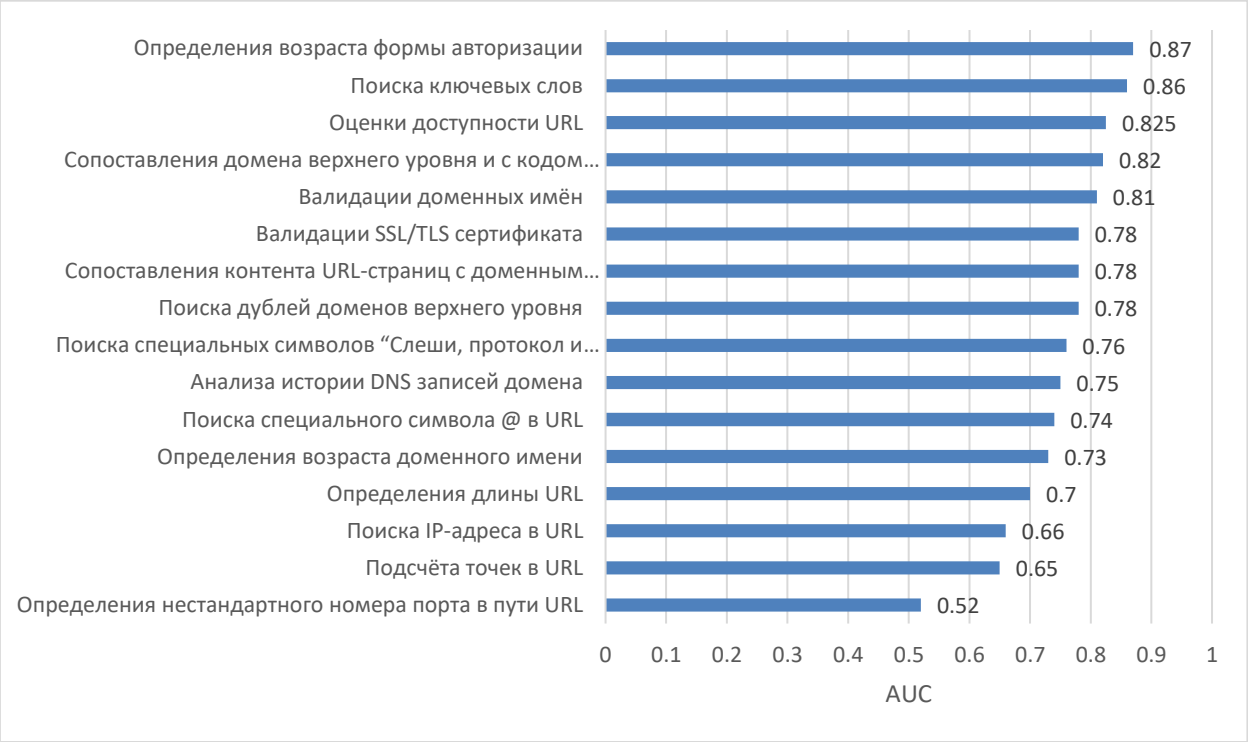


Рисунок 4.11 Результаты оценки качества работы алгоритмов

Затем, на основе вышеописанных данных была выполнена оценка качества работы методики повышения надежности через ИБ АСУ промышленными объектами. В качестве основной метрики сравнения была выбрана точность. Результат оценки качества работы методики представлен в таблице, сравнение результатов аналогичных работ представлены в таблице 4.12.

Таблица 4.12 Сравнительная таблица

№	Подход/Метод/Модель	Точность, %
1	Метод, основанный на свойствах URL-адресов [47]	87.18
2	Подход, основанный на лексических свойствах и регрессии [78]	91.5
3	Модель PhiDMA [47]	92.72
4	Подход, основанный на анализе ассоциативными правилами [79]	93
5	Cantina [37]	~95
6	Система, основанная на методе опорных векторов и индексы сходства [77]	95.8
7	Подход, основанный на свойствах URL-адресов с использованием SVM [80]	96.35
8	Методика повышения надежности через ИБ АСУ промышленными объектами	96.68

Таблица 4.13 Метрики оценки модели

Метрика	Значение
TP	3190
FN	67

Метрика	Значение
FP	60
TN	510
TPR	0.979429
FPR	0.105263
TNR	0.894737
FNR	0.020571
Precision	0.981538
Recall	0.979429
F1	0.980483
MCC	0.869782
BER	0.062917
Accuracy	0.966815

По результатам сравнения точности методики повышения надежности через ИБ АСУ промышленными объектами, с другими решениями, нужно отметить, что точность данной методики выше на 0.33 чем у ближайшего аналогичного решения. Сравнение остальных метрик методики с аналогичными подходами проведено в соответствии с данными, представленными в разделе 1.3. А именно: TPR выше на 0.003; FPR ниже на 0.035; F1-мера выше на 0.0044, Precision выше на 0.0055.

4.4 Опытная эксплуатация системы повышения надежности через ИБ автоматизированных систем управления промышленными объектами

На данном этапе проводилась опытная эксплуатация системы повышения надежности через ИБ АСУ промышленными объектами на АСУ промышленными объектами ПАО «Уралкалий» на протяжении 3-х месяцев 2019 года. Архитектура системы полностью соответствует архитектуре, представленной в разделе 3.2.

Статистические данные, полученные в процессе опытной эксплуатации, являются конфиденциальными, а значит не могут быть опубликованы в публичных источниках, в том числе в данной работе. В дальнейших расчетах использованы немногие опубликованные значения и экспертные оценки [115] (Табл. 4.14).

Таблица 4.14 Статистические данные

Объем производства за 2019 год, т.	11100000
Стоимость продукции за тонну, $.	40.8
Средний курс доллара за 2019 год, руб.	67.33
Потери продукции из-за отказов, т.	111000
Суммарные потери из-за отказов за 2019 год, руб.	304924104
Потери за 1 час отказа, руб.	519532.6518
Потери продукции за 1 час отказа, т.	189.1228788
Среднее время 1 отказа, час	4
Средние потери при 1 отказе, руб.	2078130.607
Средние потери продукции из-за 1 отказа, т.	756.491515

В процессе опытной эксплуатации были успешно обнаружены и предотвращены 6 таргетированных атак. Исходя из значительных рисков отказа оборудования при успешной реализации атак, каждая такая атака приравнивалась к отказу. На основе вышеперечисленных данных и данных из публичных источников, определены следующие статистические данные (Табл. 4.15).

Таблица 4.15 показатели за период опытной эксплуатации

Показатель\Период	**1 Месяц (без применения системы)**	**1 Месяц (во время опытной эксплуатации системы)**
Количество часов наработки АСУ промышленными объектами, час	720	720

Показатель\Период	1 Месяц (без применения системы)	1 Месяц (во время опытной эксплуатации системы)
Общее количество часов отказов АСУ промышленными объектами, час	48.24	40.32
Среднее время одного отказа АСУ промышленными объектами, час	4	4
Всего отказов АСУ промышленными объектами	12.2275	10.22
Средняя наработка на отказ АСУ промышленными объектами, час	671.76	679.68
Среднее время восстановления АСУ промышленными объектами, час	48.24	40.32
Коэффициент готовности АСУ промышленными объектами	0.933	0.944

В качестве ключевой качественной характеристики надежности через ИБ выбран коэффициент готовности К АСУ промышленными объектами (4.3), поскольку, данный показатель является комплексным и применимым к сложным системам:

$$T_{пасу} = \frac{\sum_1^n t}{n} \quad (4.1)$$

$$T_{в.пасу} = \frac{\sum_1^m t_в}{m} \quad (4.2)$$

$$K = \frac{T_{пасу}}{T_{пасу} + T_{в.пасу}} \quad (4.3)$$

где, n – число отказов; m–число восстановлений;t–наработка до наступления отказа; $t_в$ – время восстановления;$T_{пасу}$ – средняя наработка на отказ АСУ промышленными объектами (4.1);$T_{в.пасу}$ – среднее время восстановления АСУ промышленными объектами (4.2).

Применение системы улучшило ключевой показатель на 1.17% и уменьшило общее количество часов отказов на 16.42%. В случае применения системы на более длительный период, например на протяжении всего 2019

года. Оценочный экономический эффект представляется в снижении потерь продукции на 18223.88 тонн или 50 062 166.33 руб.

4.5 Выводы по главе 4

В разделе 4.1 представлен анализ имитационных исследований алгоритмов фильтрации из разделов 2.1, 2.2. Полученные результаты показывают увеличение производительности для алгоритма из раздела 2.1 приблизительно в 3 раза, за счет использования локальных мощностей и увеличение точности для алгоритма из раздела 2.2 на 0.9%.

В разделе 4.2, представлен анализ имитационных исследований алгоритмов метода алгоритмических проверок. В результате проведенного анализа, определены алгоритмы, имеющие наиболее высокую точность с минимизированным количеством ошибок. Такие как алгоритм определения возраста формы авторизации, алгоритм поиска ключевых слов, алгоритм сопоставления домена верхнего уровня с кодом страны его IP-адреса и алгоритм оценки доступности URL. Но несмотря на то, что ряд алгоритмов не обладают высокими показателями метрик из раздела 1.4, включение их результатов в обработку моделью позволяют увеличить шанс детектирования легитимных URL.

В разделе 4.3 представлен анализ имитационных исследований модели оценки опасности ресурсов и методики повышения надежности через ИБ АСУ промышленными объектами. Сформирован входной набор данных и поделён на обучающий и тестовый наборы с целью проведения исследований в разных условиях. Определена радиальная базисная функция, как функция ядра метода опорных векторов, поскольку данная функция имеет наименьший процент ошибок на тестовом наборе. Затем проведена оценка качества работы алгоритмов из раздела 2.3. В результате анализа результатов, определены

алгоритмы результат работы которых в наибольшей степени влияет на результат качества работы классификатора, ими стали: алгоритмы определения возраста формы авторизации, поиска ключевых слов, оценки доступности URL-адреса, сопоставления домена верхнего уровня с кодом страны и валидации доменных имён. Несмотря на наличие алгоритмов с низким влиянием, их наличие позволит снизить риски, возникающие при использовании злоумышленниками большого количества точек и использование нестандартного номера порта в пути URL. Так же, представлена оценка качества работы методики повышения надежности через ИБ АСУ промышленными объектами, а также сравнение полученных результатов с аналогичными работами. В результате оценки качества, разработанная методика обладает следующими метриками, которые выше, чем у аналогичного решения Accuracy, TPR, F1-мера. Так же, методика обладает низким FPR.

В разделе 4.4, представлены результаты опытной эксплуатации системы повышения надежности через ИБ АСУ промышленными объектами ПАО «Уралкалий» на протяжении 3-х месяцев 2019 года. По результатам статистических данных проведена оценка работы системы и экономического эффекта. Применение системы улучшило коэффициент готовности на 1.17% и снизило количество часов отказов на 16.42%. Оценочный экономический эффект представляется в снижении потерь продукции на 18223.88 тонн или 50 062 166.33 руб., при условии применения системы на протяжении 2019 года.

ЗАКЛЮЧЕНИЕ

Основным результатом диссертационного исследования является решение научной задачи, состоящей в обосновании пути повышения надежности АСУ промышленными объектами крупного химико-технологического предприятия за счет совершенствования системы информационной безопасности.

В ходе исследования получены результаты:

1. Модифицированы и предложены к использованию в качестве первичных фильтров алгоритмы фильтрации фишинговых ресурсов в АСУ промышленными объектами, позволяющие снизить риски внесения изменений в информационную базу, сопоставлять посещаемый ресурс пользователем с его персональным белым списком и искать формы авторизации.
2. Разработан метод алгоритмических проверок фишинговых ресурсов в АСУ промышленными объектами, позволяющий проводить анализ ресурсов на предмет наличия различных фишинговых индикаторов.
3. Разработана модель оценки опасности внешних ресурсов, основанная на методе опорных векторов с радиальной базисной функцией в качестве функции ядра, позволяющая рассчитать принадлежность ресурса к одному из классов.
4. Разработана и реализована методика повышения надежности через ИБ АСУ промышленными объектами за счет обеспечения информационной безопасности с использованием платформы сетевой безопасности, кластера микросервисов и платформы контейнеризации, позволяющая объединить ранее представленные научные продукты и выстроить механизмы взаимодействия между ними.

5. По результатам промышленной эксплуатации и имитационных исследований удалось достигнуть увеличения производительности алгоритмов фильтрации до 3-х раз, увеличения точности определения форм авторизации на 0.9% и точности распознавания фишинговых ресурсов, как минимум, на 0.33%. А также, улучшения коэффициента готовности АСУ промышленными объектами на 1.17% и уменьшения общего количества часов отказов на 16.42%.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

Аббревиатуры

АРМ – автоматизированное рабочее место

АСУ – автоматизированная система управления

АСУТП – автоматизированная система управления технологическими процессами

АСУП – автоматизированная система управления производствами

АСТПП – автоматизированная система управления технической подготовкой производства

ДИБ – доктрина информационной безопасности

ИБ – информационная безопасность

КИИ – критическая информационная инфраструктура

КСПД – корпоративная сеть передачи данных

НСД – несанкционированный доступ

ОС – операционная система

ПО – программное обеспечение

СКУД – Система контроля и управления доступом

СУБД – Система управления базами данных

ФСТЭК России – федеральная служба по техническому и экспортному контролю

DFS – Distributed File System (распределённая файловая система)

DN – Distinguished Name (уникальное имя)

DNS – Domain Name System (система доменных имен)

DOM – Document Object Model (объектная модель документа)

DoS – Denial of Service (отказ в обслуживании)

DPI – Deep packet inspection (глубокий анализ пакетов)

HAL – Hardware Abstraction Layer (слой аппаратных абстракций)

IANA – Internet Assigned Numbers Authority (администрация адресного пространства Интернет)

IE – Information Extraction (извлечение информации)

IP – Internet Protocol (Интернет протокол)

IR – Information Retrieval (информационный поиск)

MES – Manufacturing Execution System (система управления производственными процессами)

MS – Microsoft (Майкрософт)

OPC – Open Platform Communications (открытые платформенные коммуникации)

PLC – Programmable Logic Controller (программируемый логический контроллер

PR – Page Rank (ранг страницы)

SaaS – Software as a Service (ПО как услуга)

SAN – Subject Alternative Name (альтернативное имя субъекта)

SCADA – Supervisory Control And Data Acquisition (диспетчерское управление и сбор данных)

SDN – Software Definition Network (программно-определяемая сеть)

SQL – Structured Query Language (язык структурированных запросов)

SSL – Secure Sockets Layer (слой защищенных сокетов)

TF-IDF – Term frequency and inverse document frequency (частота термина-обратная частота документа)

TLS – Transport Layer Security (протокол защиты транспортного уровня)

W3C – World Wide Web Consortium (Консорциум Всемирной паутины)

WAI – Web Accessibility Initiative (инициатива веб-доступности)

WCAG – Web Content Accessibility Guidelines (рекомендации по обеспечению доступности веб-контента)

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Баранкова, И.И. Принципы построения модели надежности системы защиты информации АСУ ТП доменной печи / И.И. Баранкова, У.В. Михайлова, М.В. Афанасьева, М.Ю Афанасьев // Актуальные проблемы современной науки, техники и образования. Тезисы докладов 77-й международной научно-технической конференции. – 2019. – С. 424.
2. Баранкова, И.И. Проблемы обеспечения информационной безопасности асу тп на нижнем уровне / И.И. Баранкова, У.В. Михайлова, Г.И. Лукьянов // Актуальные проблемы современной науки, техники и образования. Тезисы докладов 77-й международной научно-технической конференции. – 2019. – С. 401-402.
3. Баранкова, И.И. Сложности, возникающие при проведении аудита информационной безопасности на предприятии / И.И. Баранкова, У.В. Михайлова, Т.В. Быкова // Вестник УрФО. Безопасность в информационной сфере. – 2019. – № 1 (31). – С. 53-56.
4. Бильфельд Н.В., Затонский А.В. Применение самоорганизующихся систем при управлении сложными процессами // Проблемы теории практики и управления. – 2007. – №12. – С.70-74.
5. Болодурина, И.П. Моделирование обеспечения надежности функционирования объектов сетевой инфраструктуры кибер-физической системы / И.П. Болодурина, Д.И. Парфёнов // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2018. – Т. 18. – № 4. – С. 41-51.
6. Болодурина, И.П. Подходы к идентификации сетевых потоков и организации маршрутов трафика в виртуальном центре обработки данных на базе нейронной сети / И.П. Болодурина, Д.И. Парфёнов //

Программные продукты и системы. – 2018. – № 3. – С. 507-513.

7. Васильев, В.И. Комплексная оценка выполнения требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами / В.И. Васильев, А.М. Вульфин, М.Б. Гузаиров, А.Д. Кириллова // Инфокоммуникационные технологии. – 2017. – Т. 15. – № 4. – С. 319-325.
8. Васильев, В.И. Методика определения актуальных угроз кибербезопасности АСУ ТП на основе стандарта ГОСТ Р 62443 / В.И. Васильев, Н.В. Кучкарова, К.И. Муслимова // Сборник избранных статей по материалам научных конференций ГНИИ "Нацразвитие". – 2018. – С. 122-126.
9. Васильев, В.И. Методика оценки рисков кибербезопасности АСУ ТП промышленного объекта / В.И. Васильев, А.М. Вульфин, К.И. Муслимова // Информационные технологии интеллектуальной поддержки принятия решений. Труды VII Всероссийской научной конференции. – 2019. – Т.1. – С. 197-201.
10. Васильев, В.И. Система поддержки принятия решений по обеспечению информационной безопасности автоматизированной системы управления технологическими процессами / В.И. Васильев, В.Е. Гвоздев, М.Б. Гузаиров, А.Д. Кириллова // Информация и безопасность. – 2017. – Т. 20. – № 4. – С. 618-623.
11. Веревкин, А.П. Диагностика, верификация и достоверизация данных для автоматизированных систем управления / А.П Веревкин // Нефтегазовое дело. – 2020. – № 3. – С. 239.
12. Веревкин, А.П. Модернизация систем управления и обеспечения безопасности как инструмент повышения эффективности процессов переработки нефти и газа / А.П. Веревкин, Т.М. Муртазин, Ф.Г. Насибуллин // Территория Нефтегаз. – 2019. – № 10. – С. 12-17.
13. Воронцов К.В. Математические методы обучения по прецедентам

(теория обучения машин). [Электронный ресурс]. – Режим доступа: www.MachineLearning.ru.

14. Вьюгин, В.В. Элементы математической теории машинного обучения: [Текст]: Учебное пособие. / В.В Вьюгин. М.: МФТИ, 2010. – 252 с.
15. Галимов, Р.Р. Анализ защищенности средств удаленного доступа корпоративной сети / Р.Р. Галимов, А.В. Корнейченко, Е.А. Мазалов // Университетский комплекс как региональный центр образования, науки и культуры. Материалы Всероссийской научно-методической конференции. – 2020. – С. 1441-1447.
16. Галимов, Р.Р. Оценка уровня защищённости информационных ресурсов на основе тестов на проникновение / Р.Р. Галимов, В.П. Членов // Информация и безопасность. – 2017. – Т. 20. – № 4. – С. 535-538.
17. Груздева, Л.М. Повышение производительности корпоративной сети АСУ в условиях воздействия угроз информационной безопасности / Л.М. Груздева, М.Ю. Монахов // Известия высших учебных заведений. Приборостроение. – 2012. – №8. – С. 53-56.
18. Дворецкий, Д.С. Новые подходы к интегрированному синтезу гибких автоматизированных химико-технологических систем / Д.С. Дворецкий, С.И. Дворецкий, С.В. Мищенко, Г.М. Островский // Теоретические основы химической технологии. – 2010. – Т.44. – № 1. – С. 69-77.
19. Демидова, Л. А. Классификация данных на основе SVM-алгоритма и алгоритма k-ближайших соседей / Л. А. Демидова, Ю. С. Соколова // Вестник РГРТУ. – 2017. – № 62. – С. 119-120.
20. Дик, Д.И. Методы обнаружения аномалий в системах обнаружения вторжений для веб-приложений / Д.И Дик // Вестник УрФО. Безопасность в информационной сфере. – 2017. – № 2 (23). – С. 10-13.
21. Дик, Д.И. Методы обнаружения вторжений для web-приложений / Д.И. Дик // НАУКА XXI ВЕКА: ТЕХНОЛОГИИ, УПРАВЛЕНИЕ,

БЕЗОПАСНОСТЬ. Сборник материалов I международной научно-практической конференции. – 2017. – С. 299-304.

22. Дмитриевский, Б.С. Автоматизированная информационная система для управления бизнес-процессами наукоемкого химического предприятия / Б.С. Дмитриевский // Системы управления и информационные технологии. – 2007. – № 2-1 (28). – С. 135-138.
23. Дмитриевский, Б.С. Автоматизированное управление производственной системой: построение модели и перевод в инновационное состояние / Б.С. Дмитриевский, О.В. Дмитриева // Вестник Тамбовского государственного технического университета. – 2014. – Т. 20. – № 2. – С. 284-291.
24. Ермилов, Е.В. Риск-анализ распределенных систем на основе параметров рисков их компонентов / Е.В. Ермилов, Е.А. Попов, М.М. Жуков, О.Н. Чопоров // Информация и безопасность. – 2013. – Т. 16. – № 1. – С. 123-126.
25. Затонский, А.В. Информационные технологии: Разработка информационных моделей и систем [Текст]: Учебное пособие / Затонский А.В. – Пермь: Издательство ГОУ ВПО Перм. гос. техн. ун-т, 2011. – С. 254-270.
26. Затонский, А.В. Разработка объектных средств имитационного и многоагентного моделирования производственных процессов / А.В. Затонский, В.Н. Уфимцева // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. Астрахань. – 2018. – № 4. – С. 56-62.
27. Затонский, А.В. Теоретический подход к управлению социально-техническими системами / А.В. Затонский // Программные продукты и системы. – 2008. – № 1. – С. 29-32.
28. Зинкевич, А.В. Аудит информационной безопасности / А.В. Зинкевич, М.С. Михайлов // Ученые заметки ТОГУ. – 2018. – Т. 9. – № 1. – С. 307-312.

29. Иевлев, О.П. Моделирование угроз информационной безопасности с использованием банка данных ФСТЭК / О.П. Иевлев, О.И. Шелухин, А.С., Большаков, Д.И. Раковский // Сборник докладов XXIII пленума ФУМО ВО ИБ и Всероссийской научной конференции "Фундаментальные проблемы информационной безопасности в условиях цифровой трансформации" (ИНФОБЕЗОПАСНОСТЬ -2019), доклады XXIII пленума ФУМО ВО ИБ и Всероссийской научной конференции. – 2019. – С. 41-47.
30. Комарова, Ю.А. Nmap как инструмент для изучения хоста/сети / Ю.А. Комарова, В.С. Вечер, М.С. Рублев // Сборник научных трудов 6-й Международной молодежной научно-практической конференции КАЧЕСТВО ПРОДУКЦИИ: КОНТРОЛЬ, УПРАВЛЕНИЕ, ПОВЫШЕНИЕ, ПЛАНИРОВАНИЕ. – Курск, 2019. – С. 161-163.
31. Кэмпбелл, Я. Информационная безопасность в контексте прав человека, новых технологий и свободы человека / Я. Кэмпбелл // Идеи и новации. – 2018. – Т.6. – №4. – С.15.
32. Лаптева, У.В. Использование методов оценки уровня защиты данных / У.В. Лаптева, О.Н. Кузяков // Проблемы формирования единого пространства экономического и социального развития стран СНГ (СНГ-2018). Материалы ежегодной Международной научно-практической конференции. – 2018. – С. 2.
33. Логиновский, О.В. Построение современных корпоративных систем / О.В. Логиновский, А.Л. Шестаков, А.А. Шинкарев // Управление большими системами: сборник трудов. – 2019. – № 81. – С. 113-146.
34. Логиновский, О.В. Применение автоматизированных информационных систем в корпоративном управлении / О.В. Логиновский, А.А. Максимов, А.С. Козлов, А.С. Зинкевич // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2007. – № 23 (95). – С. 26-34.

35. Лукьянов, Г.И. Кибербезопасность АСУ ТП предприятий индустрии 4.0 / Г.И. Лукьянов, И.И. Баранкова, Г.П. Корнилов // Энергетические и электротехнические системы. Международный сборник научных трудов. – Магнитогорск, 2019. – С. 136-142.
36. Матвейкин, В.Г. Проектирование системы управления инновационно-производственной системой / В.Г. Матвейкин, Б.С. Дмитриевский, И.С. Панченко // Вестник Тамбовского государственного технического университета. – 2011. – Т. 17. – № 2. – С. 289-296.
37. Матвейкин, В.Г. Роль науки и образования в решении проблем экосферной безопасности / В.Г. Матвейкин, Б.В. Путин, В.Д. Самарин // Вопросы современной науки и практики. Университет им. В.И. Вернадского. – 2012. – № S2 (39). – С. 42-54.
38. Машкина, И.В. Разработка EPC-моделей угроз нарушения информационной безопасности автоматизированной системы управления технологическими процессами / И.В. Машкина, И.Р. Гарипов // Безопасность информационных технологий. – 2019. – Т. 26. – № 4. – С. 6-20.
39. Митюков, Е.А. Аспекты информационной безопасности АСУ ТП / Е.А. Митюков // Первый шаг в науку. – 2016. – № 11 (23). – С. 37-39.
40. Митюков, Е.А. Атаки на DNS-сервисы АСУП, или Использование Lame delegation в своих целях / Е.А. Митюков, А.В. Затонский // Защита информации. Инсайд. – 2019. – № 4 (88). – С. 63-67.
41. Митюков, Е.А. Аудит безопасности *SCADA*-систем / Е.А. Митюков, А.В. Затонский, П.В. Плехов// Защита информации. Инсайд. – 2016. – №4. – С.72-77.
42. Митюков, Е.А. Жизненный цикл фишинговых атак и техники их реализации / Е.А. Митюков // Решение. – 2019. – Т. 1. – С. 140-142.
43. Митюков, Е.А. Меры элементарной информационной защиты для Simatic / Е.А. Митюков // Решение. – 2015. – С. 87-89.
44. Митюков, Е.А. Метрики оценки антифишинговых методов и моделей /

Е.А. Митюков // Молодежная наука в развитии регионов. – 2020. – Т. 1. – С. 65-67.

45. Митюков, Е.А. Модель обнаружения фишинговых атак на основе гибридного подхода для защиты автоматизированных систем управления производством / Е.А. Митюков, А.В. Затонский // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2020. – Т. 20. – № 2. – С. 56-66.
46. Митюков, Е.А. Модель определения фишинга на основе гибридного подхода для защиты АСУП / Е.А. Митюков, А.В. Затонский // Защита информации. Инсайд. – 2020. – № 3 (93). – С. 42-47.
47. Митюков, Е.А. Обратный SSH-туннель / Е.А. Митюков // Молодежная наука в развитии регионов. – 2018. – Т. 1. – С. 11-12.
48. Митюков, Е.А. Поиск специфических устройств, подключенных к сети Интернет / Е.А. Митюков // Молодежная наука в развитии регионов. – 2017. – Т. 1. – С. 14-16.
49. Митюков, Е.А. Практика применения reverse-proxy для защиты корпоративных приложений в Интернете / Е.А. Митюков // Автоматизированные системы управления и информационные технологии. Материалы всероссийской научно-технической конференции. – 2018. – Т.1. – С. 288-292.
50. Митюков, Е.А. Регрессионно-дифферециальное моделирование отрасли связи Российской Федерации / Е.А. Митюков // Первый шаг в науку. – 2015. – № 5-6 (5-6). – С. 3-10.
51. Митюков, Е.А. Типовая архитектура распределённой АСУ ТП / Е.А. Митюков // Молодежная наука в развитии регионов. – 2019. – Т. 1. – С. 9-10.
52. Митюков, Е.А. Уязвимости MS SQL server, или использование хранимых процедур в своих целях / Е.А. Митюков // Защита информации. Инсайд. – 2017. – № 6 (78). – С. 44-47.

53. Митюков, Е.А. Уязвимости промышленных Wi-Fi точек доступа / Е.А. Митюков // Решение. – 2016. – Т. 1. – С. 157-159.
54. Митюков, Е.А. Фишинг в автоматизированных системах управления производством / Е.А. Митюков // Решение. – 2018. – Т. 1. – С. 171-174.
55. Михайлова, У.В. Аудит информационной безопасности на предприятии / У.В. Михайлова, Т.В. Быкова // Сборник избранных статей по материалам научных конференций ГНИИ "Нацразвитие". Материалы конференций ГНИИ «НАЦРАЗВИТИЕ». – 2019. – С. 341-345.
56. Мищенко, С.В. Методика разработки системы менеджмента качества предприятия / С.В. Мищенко, С.В. Пономарев, В.А. Самородов, А.В. Трофимов // Вестник Тамбовского государственного технического университета. – 2014. – Т.10. – № 1. – С. 8.
57. Мищенко, С.В. Реализация концепции ноосферного мышления, науки и образования в решении глобальных проблем техногенной безопасности в ТГТУ / С.В. Мищенко // Вопросы современной науки и практики. Университет им. В.И. Вернадского. – 2012. – № S2(39). – С. 20-33.
58. Монахов, М.Ю. Интернет-источники для аналитика информационной безопасности / М.Ю. Монахов, А.П. Кузнецова, Е.И. Яковлева // ПЕРСПЕКТИВНЫЕ ТЕХНОЛОГИИ В СРЕДСТВАХ ПЕРЕДАЧИ ИНФОРМАЦИИ - ПТСПИ-2017. Материалы 12-ой международной научно - технической конференции. – 2017. – Т.1. – С. 213-215.
59. Монахов, М.Ю. Модели обеспечения достоверности и доступности информации в информационно-телекоммуникационных системах: монография / М.Ю. Монахов, Ю.М. Монахов, Д.А. Полянский, И.И. Семенова; Владимир; Издательство Владимирский государственный университет, 2015. – 208 с.
60. Монахов, М.Ю. Особенности среды обеспечения достоверности информации в информационно-телекоммуникационных системах /

М.Ю. Монахов, И.И. Семенова, Д.А. Полянский, Ю.М. Монахов // Фундаментальные исследования. – 2015. – № 9-11. – С. 2403-2407.

61. Монахов, Ю.М. Модели угрозы распространения запрещенной информации в информационно-телекоммуникационных сетях / Ю.М. Монахов, М.Ю. Монахов // Динамика сложных систем. – 2015. – №2. – С. 65-69.
62. Муромцев, Д.Ю. Усовершенствование подсистемы обеспечения работоспособности средств защиты информации в системе мониторинга инцидентов информационной безопасности банка / Д.Ю. Муромцев, С.В. Попов, В.Н. Шамкин // Вестник Тамбовского государственного технического университета. – 2020. – Т.26. – №2. – С. 176-187.
63. Нигерийский фишинг: атаки на промышленные компании. [Электронный ресурс]. – Режим доступа: https://ics-cert.kaspersky.ru/reports/2017/06/15/nigerian-phishing-industrial-companies-under-attack/
64. Носаль, И.А. Потенциал нападения и типовая модель нарушителя / Носаль И.А. // Информационная безопасность и защита персональных данных: Проблемы и пути их решения: материалы VI Межрегиональной научно-практической конф. – Брянск: Издательство БГТУ, 2014. – С.96-101.
65. Остапенко, А.Г. Инновационные тренды развития и информационные риски развития IT-сферы в контексте обеспечения критически важных объектов / А.Г. Остапенко, Е.В. Ермилов, А.О. Калашников // Информация и безопасность. – 2013. – Т. 16. – № 3. – С. 323-334.
66. Остапенко, А.Г. Формализация процесса управления рисками в информационно-технологической инфраструктуре критически важного объекта / А.Г. Остапенко, А.О. Калашников, Е.В. Ермилов, Н.Н. Корнеева // Информация и безопасность. – 2014. – Т. 17. – № 2. – С. 164-179.

67. Плешко, Д.Ю. Уязвимости низкоуровневых протоколов как инструмент для атаки на АСУ ТП / Д.Ю. Плешко // Актуальные проблемы энергетики: материалы 74-й научно-технической конференции студентов и аспирантов. – 2018. – С. 730-733.
68. Пономарев, С.В. Практические подходы к оценке рисков в смк / С.В. Пономарев // Методы менеджмента качества. – 2016. – № 7. – С. 30-35.
69. Поршнев, С.В. Технология семантического анализа дампа трафика информационных потоков в компьютерных сетях / С.В. Поршнев, Д.А. Божалкин // Информационные технологии. – 2014. – № 11. – С. 12-19.
70. Приказ Федеральной службы по техническому и экспортному контролю от 14 марта 2014 г. № 31 "Об утверждении требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды". [Электронный ресурс]. – Режим доступа: https://fstec.ru/normotvorcheskaya/akty/53-prikazy/868-prikaz-fstek-rossii-ot-14-marta-2014-g-n-31
71. Приходькова, И.В. Операционные системы в 2-х частях. Том Часть 1 [Текст]: Учебное пособие / И.В. Приходькова, О.В. Гостевская. – Волгоградский государственный технический университет, 2016. – С.19-39.
72. Радивилова, Т.А. Анализ основных атак на DNS-сервер и методы использования DNSSEC при защите DNS-сервера / Т.А. Радивилова, В.С. Бушманов // Технологический аудит и резервы производства. – 2013. – Т. 2. – № 1 (10). – С. 16-19.
73. Селифанов, В.В. Применение методов автоматизации при определении актуальных угроз безопасности информации в информационных система с применением банка данных угроз ФСТЭК России / В.В. Селифанов, П.А. Звягинцева, Я.В. Юракова, И.С. Слонкина //

Интерэкспо Гео-Сибирь. – 2017. – Т. 8. – С. 202-209.

74. Симаев, А.В. Национальный координационный центр по компьютерным инцидентам / А.В. Симаев, Д.С. Стенькин // Научная дискуссия современной молодежи: актуальные вопросы юридических наук. Материалы II научно-практической конференции. – 2018. – С. 144-149.
75. Синадский, Н.И. Автоматизация тестирования сетевых средств защиты информации на основе применения эволюционно-генетического подхода / Н.И. Синадский, А.В. Агафонов // Математические структуры и моделирование. – 2018. – № 2 (46). – С. 125-134.
76. Сухоедов, К.Б. Подходы России и Японии к кибербезопасности / К.Б. Сухоедов, А.А. Слинько // Мегатренды мировой политики. Сборник научных статей по материалам 5-ой межвузовской научно-практической конференции молодых ученых. – 2018. – С. 125-131.
77. Уткин, М.А. Организация распределенных АСУТП опасных производств / М.А. Уткин, М.С. Федоров, М.Л. Немудрук, С.А. Нибур // Всероссийская научная конференция по проблемам управления в технических системах. – 2015. – № 1. – С. 154-157.
78. Федеральный закон от 26 июля 2017 г. N 187-ФЗ "О безопасности критической информационной инфраструктуры Российской Федерации". [Электронный ресурс]. – Режим доступа: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kriticheskoj-informatsionnoj-infrastruktury/285-zakony/1610-federalnyj-zakon-ot-26-iyulya-2017-g-n-187-fz
79. Финогеев, А.Г. Проблемы безопасности беспроводной сенсорной сети в SCADA-системах АСУ ТП / А.Г. Финогеев, И.С. Нефедова, К.В. Тхай // Известия Волгоградского государственного технического университета. – 2014. – № 6 (133). – С. 66-72.
80. Чернышов, В.Н. Обеспечение информационной безопасности, современные возможности и проблемы / В.Н. Чернышов, А.В. Терехов,

О.М. Дементьев, М.Н. Кочеткова // Учебное электронное издание комплексного распостранения. 2016. – С. 60-83.

81. Чернышов, В.Н. Уголовно-правовая охрана авторских прав в сети Интернет как элемент обеспечения безопасности информационного общества / В.Н. Чернышов, М.Н. Кочеткова // Современное право. – 2018. – № 7-8. – С. 103-109.
82. Шереметьева, С.О. Методы и модели автоматического извлечения ключевых слов / С.О. Шереметьева, П.Г. Осминин // Вестник Южно-Уральского государственного университета. Серия: Лингвистика. – 2015. – Т. 12. – № 1. – С. 76-81.
83. Щипакина А.А. Модель и алгоритм отбора данных для обеспечения информационного процесса оценки достоверности сведений при проведении экспертизы качества медицинской помощи / А.А. Щипакина, И.Ю. Квятковская // Прикаспийский журнал: управление и высокие технологии. – 2015. – № 4 (32). – С. 86-95.
84. Энциклопедия АСУ ТП. Контроллеры для систем автоматизации. [Электронный ресурс]. – Режим доступа: https://www.bookasutp.ru/chapter6_1.aspx
85. Юрьев, Н. Человеческий фактор / Н. Юрьев // Техника и вооружение вчера, сегодня, завтра. – 2003. – № 09. – С. 14-15.
86. Agarwal, P. A Novel Approach for Phishing URLs Detection / P. Agarwal, D. Mangal // International Journal of Science and Research (IJSR). – 2016. – Vol. 5. – Iss. 5. – PP. 1117-1122.
87. Anti-Phishing Working Group reports. [Электронный ресурс]. – Режим доступа: https://apwg.org/
88. Attacks on industrial enterprises using RMS and TeamViewer. [Электронный ресурс]. – Режим доступа: https://securelist.com/attacks-on-industrial-enterprises-using-rms-and-teamviewer/87104/
89. Banik, B. Phishing URL detection system based on URL features using SVM / B. Banik, A. Sarma // International Journal of Electronics and Applied

Research (IJEAR). – 2018. – Vol. 5. – Iss. 2. – PP. 52-53.

90. Bolodurina, I.P. Development of prototype of autonomous self-organizing system for ensuring network security in enterprise based on technology of virtualization network functions / I.P. Bolodurina, D.I. Parfenov, V.A. Torchin, L.V. Legashev, V.M Shardakov. // Proceedings - 2018 Global Smart Industry Conference, GloSIC 2018. – 2018. – PP. 1-8.
91. Cao, Y. Anti-phishing based on automated individual white-list / Y. Cao, W. Han, Y. Le // The 4th ACM Workshop on Digital Identity Management. New York, NY, USA: ACM. – 2008. – PP. 51-60.
92. China-based hacker group targeting Indian firms: FireEye. [Электронный ресурс]. – Режим доступа: https://www.thehindubusine*SSL*ine.com/info-tech/chinabased-hacker-group-targeting-indian-firms-fireeye/article9627468.ece
93. Chou, N. Client-side defense against web-based identity theft / N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, J.C. Mitchell // NDSS 2004. Stanford. – 2004. – PP.2-13.
94. Collin, B.C. The Future of Cyberterrorism / B.C. Collin // Crime and Justice International. – 1997. – Vol. 13. – Iss. 2. – PP.15-18.
95. DDoS Threat Report 2016 Q2. [Электронный ресурс]. – Режим доступа: https://www.nexusguard.com/
96. Dell'Amico, M. Monte Carlo Strength Evaluation: Fast and Reliable Password Checking / M. Dell'Amico, M. Filippone // The 22nd ACM SIGSAC Conference on Computer and Communications Security. – 2015. – PP. 158-169.
97. Fette, I. Learning to detect phishing emails / I. Fette, N. Sadeh, A. Tomasic // The 16th International Conference on World Wide Web. – 2007. – PP. 649-656.
98. Garera, S. A framework for detection and measurement of phishing attacks / S. Garera, N. Provos, M. Chew, A.D. Rubin // The 2007 ACM workshop on Recurring malcode. – 2007. – PP. 1-8.

99. Gastellier-Prevost, S. Decisive heuristics to differentiate legitimate from phishing sites / S. Gastellier-Prevost, G.G. Granadillo, M. Laurent // The conference on network and information systems security (SAR-SSI). – 2011. – PP. 2-7.
100. Gowtham, R. A comprehensive and efficacious architecture for detecting phishing webpages / R. Gowtham, I.A. Krishnamurthi // The Computers & Security. – 2014. – Vol. 40. – PP. 23-37.
101. Gunikhan, S. PhiDMA – A phishing detection model with multi-filter approach / S. Gunikhan, K.S. Kuppusamy // *Journal of King Saud University – Computer and Information Sciences*. – 2017. – PP. 1-14.
102. Jeeva, S.C. Intelligent phishing *URL* detection using association rule mining / S.C. Jeeva, E.B. Rajsingh // Human-centric Computing Information Science. – 2016. – PP. 2-17.
103. Kang, J. Advanced white list approach for preventing access to phishing sites / J. Kang, D. Lee // Convergence Information Technology, 2007 International Conference on. – 2007. – PP. 491-496.
104. Kumaraguru, P. Teaching Johnny not to fall for phish / P. Kumaraguru, S. Sheng, A. Acquisti, L.F. Cranor, J. Hong // ACM Transactions on Internet Technology. – 2010. – PP. 2-25.
105. Liu, Z. Automatic keyphrase extraction via topic decomposition / Z. Liu, W. Huang, Y. Zheng, M. Sun // The 2010 Conference on Empirical Methods in Natural Language Processing. Cambridge, Massachusetts. – 2010. – PP. 366-376.
106. Lu, L., A study of Linux file system evolution / L. Lu, A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau, S. Lu // The 11th USENIX conference on File and Storage Technologies (FAST'13). – 2013. – PP. 31-44.
107. Mashkina, I.V. Development of protection object model - industrial control system using system analysis / I.V. Mashkina, I. Garipov // 2018 International Russian Automation Conference, RusAutoCon 2018. – 2018. – PP. 6-20.

108. Mashkina, I.V. Threats modeling and quantitative risk analysis in industrial control systems / I.V. Mashkina, I. Garipov // 2018 International Russian Automation Conference, RusAutoCon 2018. – 2018. – PP. 9-16.
109. Mityukov, E.A. Phishing detection model using the hybrid approach to data protection in industrial control system / E.A. Mityukov, A.V. Zatonsky, P.V. Plekhov, N.V. Bilfeld // IOP Conference Series: Materials Science and Engineering. International Workshop "Advanced Technologies in Material Science, Mechanical and Automation Engineering – MIP: Engineering – 2019". – 2019. – PP.1-6.
110. Muromtsev, D.Y. Functional modeling of business processes fordevelopment of control and monitoring systems / D.Y. Muromtsev, A.N. Gribkov, V.N. Shamkin, A.G. Divin, A.P Savenkov // Proceedings of the 2017 International Conference "Quality Management, Transport and Information Security, Information Technologies", IT and QM and IS 2017. – 2017. – PP. 440-442.
111. Pan, Y. Anomaly based web phishing page detection / Y. Pan, X. Ding // The 22nd annual computer security applications conference (ACSAC'06). – 2006. – PP. 381-392.
112. Romansky, R. A Survey on Digital World Opportunities and Challenges for User's Privacy / R. Romansky // International Journal on Information Technologies and Security (IJITS). – 2017. – Vol. 9. – Iss. №4. – PP. 103-104.
113. Sag, I.A. Multiword Expressions: A Pain in the Neck for NLP. / I.A. Sag, T. Baldwin, F. Bond, A. Copestake, D. Flickinger.// The Third International Conference on Computational Linguistics and Intelligent Text Processing CICLing '02. – 2002. – PP. 1-15.
114. Tewari, A. Recent survey of various defense mechanisms against phishing attacks / A. Tewari, A.K. Jain, B.B. Gupta // J. Inf. Privacy Sec. – 2016. – Vol. 12. – №1, 3-13. – PP. 2-11.
115. Uralkali – report 2019. [Электронный ресурс]. – Режим доступа:

https://www.uralkali.com/upload/content/Uralkali_AR_2019-en.pdf

116. Vapnik, V.N. Statistical Learning Theory: monograph / V.N. Vapnik; New York: John Wiley & Sons. – 1998. – PP. 730-732.
117. Xiang, G. A hybrid phish detection approach by identity discovery and keywords retrieval / G. Xiang, J.I. Hong. // The 18th international conference on world wide web. – ACM Press. – 2009. – PP. 571-578.
118. Xiang, G. Cantina+: a feature-rich machine learning framework for detecting phishing web sites / G. Xiang, J.I. Hong, C.P. Rose, L. Cranor // ACM Transactions on Information and System Security. – 2011. – PP. 20-22.
119. Yoana, A. I. Assessment of the Probability of Cyberattacks on Transport Management Systems / A. I. Yoana // International Journal on Information Technologies and Security (IJITS). – 2018. – Vol. 10. – Iss. 4. – PP. 99-100.
120. Yu, L. Bio-Inspired Credit Risk Analysis / L. Yu, S. Wang, K.K. Lai, L. Zhou // Computational Intelligence with Support Vector Machines. – 2008. – PP. 244-245.
121. Zhang, J. Highly predictive blacklisting / J. Zhang, P. Porras, J. Ullrich // The 17th conference on security symposium. USA. – 2008. – PP. 107-122.
122. Zhang, Y. CANTINA: a content-based approach to detecting phishing websites / Y. Zhang, J. Hong, L. Cranor // The 16th International World Wide Web Conference (WWW2007). – 2007. – PP. 639-648.
123. Zouina, M. A novel lightweight URL phishing detection system using SVM and similarity index / M. Zouina, B. Outtaj // Human-centric Computing and Information Sciences, – 2017. – PP. 3-13.

Приложение А

Перечень рекомендаций по настройке конфигураций сервисов АСУ промышленными объектами

Совершенствование методов повышения надежности DNS сервисов АСУ промышленными объектами. На операторских уровнях и уровнях автоматического управления АСУ промышленными объектами, описанных в разделе 1.3, зачастую размещают различные инфраструктурные сервисы. К таким сервисам относятся DNS, DHCP, NTP, WSUS, SSH, MS Structured Query Language (SQL) и другие. Эти сервисы с одной стороны позволяют удалённо поддерживать инфраструктуру АСУ промышленными объектами [15] и автоматизировать различные операции, с другой стороны являются потенциальной точкой для развития векторов атак, направленных на АСУ промышленными объектами.

Одним из таких векторов являются атаки на DNS [40]. DNS – это технология, представляется как распределённая система для получения информации о доменах. В большинстве случаев используется для получения IP-адреса по имени хоста, либо для информации об обслуживающих узлах для протоколов в домене (SRV-записи), либо для получения информации о маршрутизации почты (MX-записи) [31]. Вывод из строя сервисов, связанных с данной технологией, существенно влияет на работу любого промышленного объекта в целом, если таковые в ней имеются. Этим и обусловлена популярность атак на DNS.

Ежеквартальный отчет исследователей компании Nexusguard за 2й квартал 2016 года [95], показал значительное развитие вектора атак на DNS. Несмотря на то, что атаки на NTP занимают первую строку рейтинга в соответствии с рисунком А.1, атаки на DNS занимают 44.2% от общего количества атак. Экспертами отмечено, что атаки на NTP и DNS наиболее предпочтительны злоумышленниками для точечных атак на компании.

Следовательно, DNS-сервисы играют одну из ключевых ролей, как в обеспечении работоспособности сервисов компании, которые имеют публикацию в сети Интернет. Так и в предоставлении легитимных данных (соответствие DOMain name и IP-адреса ресурса) пользователям АСУ промышленными объектами компании, которые использую эти сервисы.

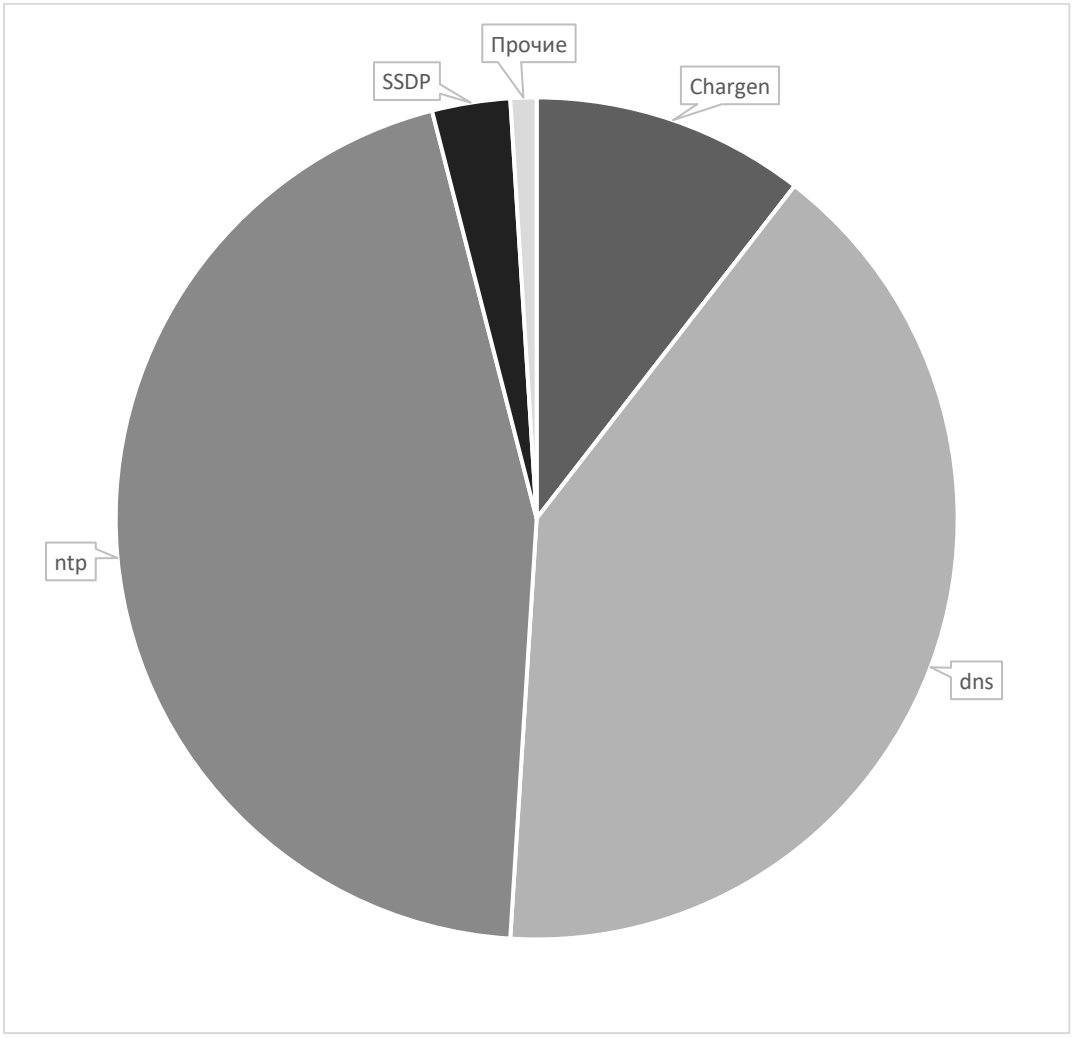


Рисунок А.1 - Атаки по типам

С целью совершенствования методов защиты [75] DNS-сервисов АСУ промышленными объектами, предлагается подход, который бы повышает надежность и живучесть DNS-сервисы АСУ промышленными объектами от атак злоумышленников. Для достижения цели, сформированы следующие задачи:

1. Провести анализ атак на DNS, рассмотреть их реализации на АСУ промышленными объектами;
2. Рассмотреть возможные реализации атак на DNS-сервисы АСУ промышленными объектами;
3. Доработать существующие либо разработать новые методы защиты DNS в соответствии с особенностями АСУ промышленными объектами;
4. Провести практическое исследование.

В данном исследовании предложен подход, который позволяет защитить DNS-сервисы АСУ промышленными объектами от атак с использованием «lame delegation». Это тип ошибки, которая возникает, когда сервер имен назначается как авторитетный сервер для имени домена, для которого у него нет достоверных данных (адрес невозможно найти, не отвечает на запросы, негативно отвечает на запросы). В предлагаемом подходе первоначально проверяется регистрация вторичных DNS-серверов; затем регистрация домена в случае, если DNS-сервер, имеет в собственном имени домен, который ему делегирован; затем проверяется доступность зоны на “Name server” которым она делегирована.

Атаки на DNS в зависимости от последствий хорошо ложатся в концепцию 3-х классических элементов треугольника ИБ: целостности, доступности, конфиденциальности. Их можно разделить на 3 основных типа:

- подмена DNS-сервера или сетевых запросов/ответов;
- перехват сетевого трафика между клиентом и запрашиваемым ресурсом;
- нарушение доступности сервиса или отдельных ресурсов.

Обнаружение атаки, в зависимости от её типа, задача не из простых. Если при нарушении доступности, фактом атаки считается отсутствие неработоспособности сервиса, то в случае с 2 первыми типами атак, все значительно сложнее. Факт перехвата сетевого трафика выявить достаточно сложно, но, если использовать криптостойкие алгоритмы, задача в выявления

перехвата сводится к проверке валидности сертификата второй стороны. В ряде случаев первый тип атак обнаружить не так просто.

Итак, рассмотрим более подробно вариацию реализации первого типа атак с использованием “lame delegation” в соответствии с рисунком А.2, поскольку этот сценарий один из возможных для атаки на АСУ промышленными объектами. Определим угрозы [29] относящиеся к данному сценарию:

УБИ.063: Угроза некорректного использования функционала программного и аппаратного обеспечения. Поскольку используются декларированные возможности программного средства, сама угроза связана со слабостями механизма обработки данных и команд, вводимых пользователями. Источник угрозы: внешний нарушитель со средним потенциалом. Последствия реализации угрозы: нарушение конфиденциальности, целостности, доступности.

УБИ.019: Угроза заражения DNS-кеша. Как следствие УБИ063, использование возможности перенаправления нарушителем сетевого трафика, основываясь на некорректном использовании функционала программного и аппаратного обеспечения, через собственный сетевой узел путём опосредованного изменения таблиц соответствия IP-адресов и доменных имён, хранимых в DNS-сервере, за счёт генерации лавины возможных ответов на запрос DNS-сервера легальному пользователю или за счёт эксплуатации уязвимостей DNS-сервера. Источник угрозы: внешний нарушитель с низким потенциалом/ Последствия реализации угрозы: Нарушение конфиденциальности.

УБИ.175: Угроза «фишинга». Как следствие УБИ019, использование подложного DNS-сервера для неправомерного ознакомления нарушителем с защищаемой информацией пользователя путём убеждения его с помощью методов социальной инженерии, зайти на поддельный сайт (выглядящий одинаково с оригинальным), на котором от дискредитируемого пользователя требуется ввести защищаемую информацию или открыть заражённое вложение в письме. Источники угрозы: внешний нарушитель с низким

потенциалом. Объект воздействия: рабочая станция АСУ промышленными объектами, сетевое программное обеспечение, сетевой трафик.

Анализ возможностей потенциального нарушителя представлен в модели нарушителя. В случае с типом атаки “lame delegation”, нарушители – внешние, атакуют сервисы из-за пределов контролируемой зоны АСУ промышленными объектами. Внешний нарушитель может осуществлять:

- перенаправление сетевого трафика основываясь на некорректном использовании функционала программного и аппаратного обеспечения;
- несанкционированный доступ к информации посредством фишинговых атак;
- отказ в обслуживании DNS-сервисов;
- перехват передаваемой незашифрованной информации;
- останов всех опубликованных в сети Интернет внешних сервисов цели, имеющие собственные доменные имена, которые размещены в зонах зараженных или некорректно делегированных DNS-серверов.

При этом, для реализации атаки нарушитель может использовать доступные в свободной продаже аппаратные средства и программное обеспечение, в том числе программные и аппаратные компоненты криптосредств; специально разработанные технические средства и программное обеспечение; средства перехвата и анализа информационных потоков в каналах связи.

Ниже представлены шаги реальной атаки на действующее крупнейшее производственное предприятие РФ, которая была реализована в последней декаде декабря 2018 года:

1. Злоумышленником велось постоянное наблюдение за изменениями в глобальной базе доменов и их владельцев IANA;
2. Фиксировалось каждое изменение параметров делегирования корневых “Name server” для интересующих его доменов;

3. Проверялась доступность и существование записей доменов “Name server” в IANA, которым делегировались зоны;
4. В случае отсутствия записи о домене в глобальной базе IANA, злоумышленник регистрировал такой домен у любого регистратора;
5. Для нового домена настраивались “Name server” параметры, указывающие IP-адрес заранее подготовленного DNS-сервера. На котором создана DNS-зона, в которой на любой запрос, отправлялся ответ, с IP-адресом сервера злоумышленника. Ключевым моментом здесь были параметры обновления файла зоны и записей, которые выставлялись в максимальное значение.

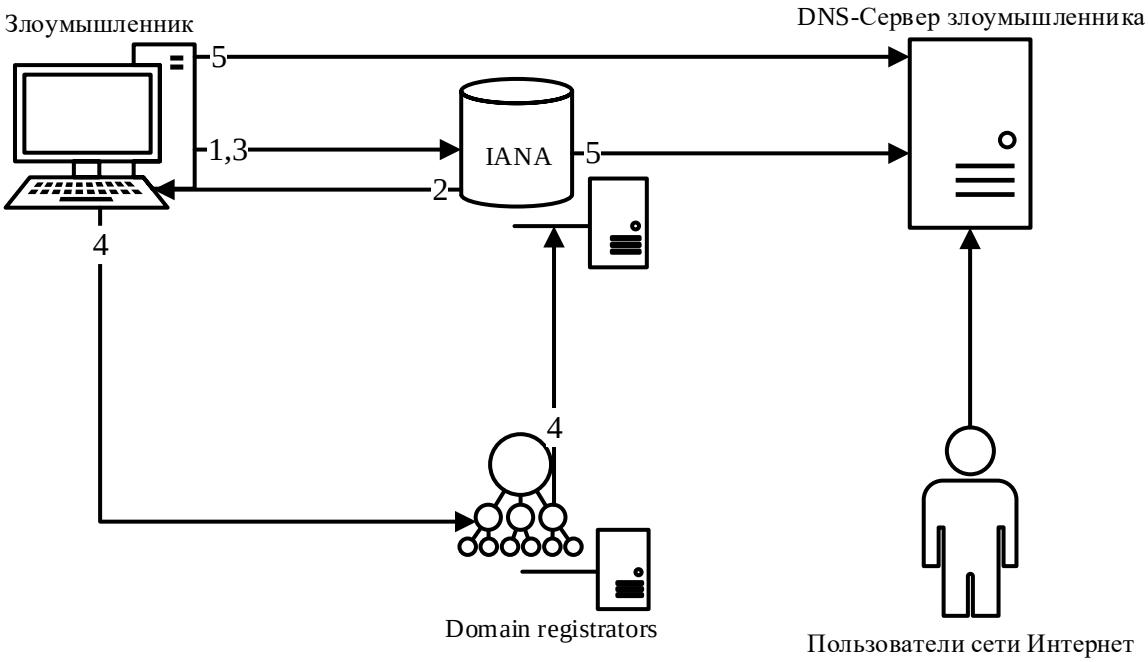


Рисунок А.2 - Общая схема атаки

В результате реализации данной атаки, были выявлены следующие последствия:

- Внешние клиенты всего мира, которые обращающиеся к ресурсам скомпрометированного домена получат нелегитимные DNS-записи

(отравят свой кеш), а кеширующие DNS-сервера поставщиков услуг, их закешируют;

- В случае MX, SRV и других записей в DNS-зоне скомпрометированного домена, некорректно работали соответствующие сервисы (электронная почта, различные порталы, личные кабинеты, видеоконференцсвязь и др.);
- У владельца домена в локальной инфраструктуре, локальные клиенты так же использовали внешние ресурсы скомпрометированного домена. Эти локальные клиенты получили в кеш скомпрометированные записи (отравление своего кеша), что повлекло за собой невозможность управления внутренними ресурсами, в том числе АСУ промышленными объектами.

Дальнейшее развитие атаки и ликвидация последствий не раскрываются.

В данном случае, очень важно определить развитие атаки на раннем этапе, иначе, если большинство поставщиков услуг закешируют записи фейковой зоны, все сервисы компании, относящиеся к этой зоне будут недоступны. С целью анализа времени поступления запросов DNS при смене DNS-сервера для DNS-записей, проведено исследование, которое заключалось в следующем:

1. Создан фейковый DNS-сервер, который отвечал на любой запрос, IP-адресом 127.0.0.1, ttl файла зоны и записей зоны выставлен максимально возможны;
2. Для тестового домена, в IANA изменена настройка параметров “Name server”. Параметры были заведомо заведены с ошибкой, которая перенаправляла часть запросов на легитимный DNS-сервер, а часть на фейковый, который был подготовлен ранее;
3. В такой конфигурации серверы проработали сутки, затем параметры “Name server” были восстановлены в первоначальное состояние.
4. Анализ запросов [69] к фейковому DNS-серверу, после исправления конфигурации.

В результате, было выявлено, что большая часть запросов, порядка 70% стали приходить на корректные DNS-сервера, через сутки. С каждым днем ситуация улучшалась и через неделю, практически все клиенты, делали запросы к корректный серверам. За исключением, клиентов компании LLC “Yandex”. Поскольку для своих кеширующих DNS-серверов эта компания, не использует время жизни, указанное в зоне. А использует свое и в ряде случаев, оно может достигать более 2-х недель. В результате последний запрос от внешнего клиента был через 3 недели, затем запросы прекратились. Для того, чтобы клиенты стали получать корректные адреса для имен доменов из скомпрометированной ранее зоны, необходимо почистить кеш DNS. Это возможно сделать только в случае, если имеется доступ к DNS-серверу. В случае с внешними клиентами это невозможно, необходимо вести взаимодействие с технической службой поставщиков услуг, которые очень неохотно идут на контакт.

Дабы избежать последствий, необходимо проводить ряд проверок настройке DNS-сервера и изменении параметров доменных имен. Рассмотрим возможные варианты и меры защиты:

1. При изменении параметров “Name server” в глобальной базе доменных имен, любой “Name server”, как минимум должен быть зарегистрирован в глобальной базе, как вторичный для одного из доменов, которым Вы владеете. Это характерно только когда указан исключительно FQDN (Fully Qualified DOMain Name). Этот механизм реализован у большинства регистраторов доменных имен.
2. В случае, если указаны FQDN + IP-адрес в качестве параметров “Name server“ для домена, никаких проверок регистратором не делается. Подобная конфигурация может понадобиться в случае, если DNS-сервер имеет имя 3 уровня домена для которого указывается параметр (т.е. DNS-сервер ns.test.com указывается для домена test.com). В результате, может возникнуть вышеописанная ситуация с неправильным делегированием (“lame delegation”). Предлагается использовать whois-

сервисы для проверки существования FQDN и корректности их владельцев. Для проверки достаточно использовать командлет whois:

```
# whois test.com
```

Когда домен существует и зарегистрирован, результат выполнения команды выглядит так:

```
   IP Address: 50.23.225.49
   Registrar: Network Solutions, LLC
   Registrar WHOIS Server: whois.networksolutions.com
   Registrar URL: http://networksolutions.com

   DOMain Name: TEST.COM
   Registry DOMain ID: 5429075_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.networksolutions.com
   Registrar URL: http://networksolutions.com
   Updated Date: 2017-04-18T08:54:58Z
   Creation Date: 1997-06-18T04:00:00Z
   Registry Expiry Date: 2019-06-17T04:00:00Z
   Registrar: Network Solutions, LLC
   Registrar IANA ID: 2
   Registrar Abuse Contact Email: abuse@web.com
   Registrar Abuse Contact Phone: +1.8003337680
   DOMain Status: clientTransferProhibited
   Name Server: NS65.WORLDNIC.COM
   Name Server: NS66.WORLDNIC.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form:
https://www.icann.org/wicf/
>>> Last update of whois database: 2019-01-02T18:01:22Z <<<
```

Когда домен не зарегистрирован, например домен «*TEST112.COM*», результат выполнения команды выглядит так:

```
No match for "TEST112.COM".
>>> Last update of whois database: 2019-01-02T18:04:37Z <<<
```

3. В случае корректности параметров “Name server” необходимо провести проверку наличия делегированной зоны на указанных серверах, как для собственных, так и для тех серверов, которыми администратор домена не управляет. Проверку можно осуществить при помощи командлета dig:

```
# dig NS @ns.test.com test.com +norec
```

Корректный вывод, когда все сконфигурировано верно, выглядит так:

```
; <<>> DiG 9.9.4-RedHat-9.9.4-72.el7 <<>> NS @ns.test.com test.com +norec
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57045
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 3
…
;; Query time: 28 msec
;; SERVER: 1.1.1.1#53(1.1.1.11)
;; WHEN: Wed Jan 02 22:52:54 +05 2019
;; MSG SIZE  rcvd: 108
```

Не корректный, когда зона отсутствует или сервер недоступен:

```
# dig: couldn't get address for 'ns1.test.com': not found
```

1. В случае отсутствия зоны на собственных серверах, необходимо отменить делегирование, до тех пор, пока зона не будет соответствующим образом сконфигурирована.
2. В случае отсутствия зоны на серверах, которыми не управляет администратор домена, необходимо отменить делегирование и сообщить администратору домена, на который настроено

делегирование, об ошибке конфигурации. Необходимость отмены делегирования в первую очередь обусловлена тем, что сервисы зоны могут быть недоступны для пользователей, в случае если обращение будет к DNS-серверу, на котором отсутствует требуемая зона.

3. В случае, когда ответ от сервера есть, т.е. зона присутствует, но сервер подготовлен злоумышленником. Сервер может возвращать ответы на любой запрос. В данном случае предлагается проводить проверку соответствия NS-записей зоны и “Name server” параметров в глобальной базе IANA. Затем, проверить соответствие IP-адресации вышеуказанных записей.

   Для проведения проверки предлагается следующий алгоритм:

3.1 Проверить NS записи в зоне, для этого необходимо подключиться к DNS-серверу и использовать следующую команду:

```
# grep NS /var/named/test.com
```

Важно отметить: путь до файла зоны может быть другим. В результате, в случае наличия файла зоны и корректно настроенных NS записей, на экране должны появиться следующие записи:

```
NS      ns65.worldnic.com.
NS      ns66.worldnic.com.
```

3.2 Затем, проверить NS записи, которые возвращают корневые DNS сервера, для этого использовать команду:

```
# nslookup -type=NS test.com 1.1.1.1
```

В результате, на экране должны появиться следующие записи:

```
Server:         1.1.1.1
```

```
Address:        1.1.1.1#53
Non-authoritative answer:
test.com        nameserver = ns65.worldnic.com.
test.com        nameserver = ns66.worldnic.com.
```

3.3Следующим шагом, предлагается проверить соответствие результатов в ручном режиме, либо в автоматическом режиме. Для автоматической проверки необходимо подготовить полученные ранее результаты и сравнить их, сделать это можно выполнив следующий bash-скрипт:

```
#!/bin/bash
localzone=$(grep NS /var/named/test.com | sed 's|.*NS\t||'| sort)
fromInternet=$(nslookup -type=NS test.com 1.1.1.1 | grep nameserver | sed
's|.*nameserver = ||' | sort)

if [ "$localzone" = "$fromInternet" ]; then
  echo " Configuration of the name servers is correctly "
  else
echo " Configuration of the name servers is not correctly "
fi
```

где $localzone переменная, содержащая список DNS-серверов указанных в конфигурации зоны; переменная $fromInternet содержит список DNS-серверов для домена, полученный от корневых DNS-серверов сети Интернет; в результате выполнения, в случае корректной настройке на экране будет надпись “correctly”, в противном случае “not correctly”, что будет означать ошибку в конфигурации.

В результате практического применения предложенного подхода к проверке конфигурации DNS-серверов АСУ промышленными объектами, повышается эффективность настройки данного сервиса, как со стороны глобальной базы IANA, так и локальной конфигурации зон DNS-сервисов.

Совершенствование методов повышения надежности СУБД АСУ промышленными объектами. Следующим вектором атак на сервисы АСУ промышленными объектами - являются атаки, направленные на СУБД.

Наиболее распространённой СУБД, используемой в АСУ промышленными объектами, является MS SQL Server. Выбор именно этой СУБД, связан с простотой администрирования; возможностями WEB-подключений; встроенным инструментарием для тиражирования, синхронизации и анализа данных; наличие средств удалённого доступа и не большой себестоимостью самой СУБД, по отношению к другим (Oracle или DB2).

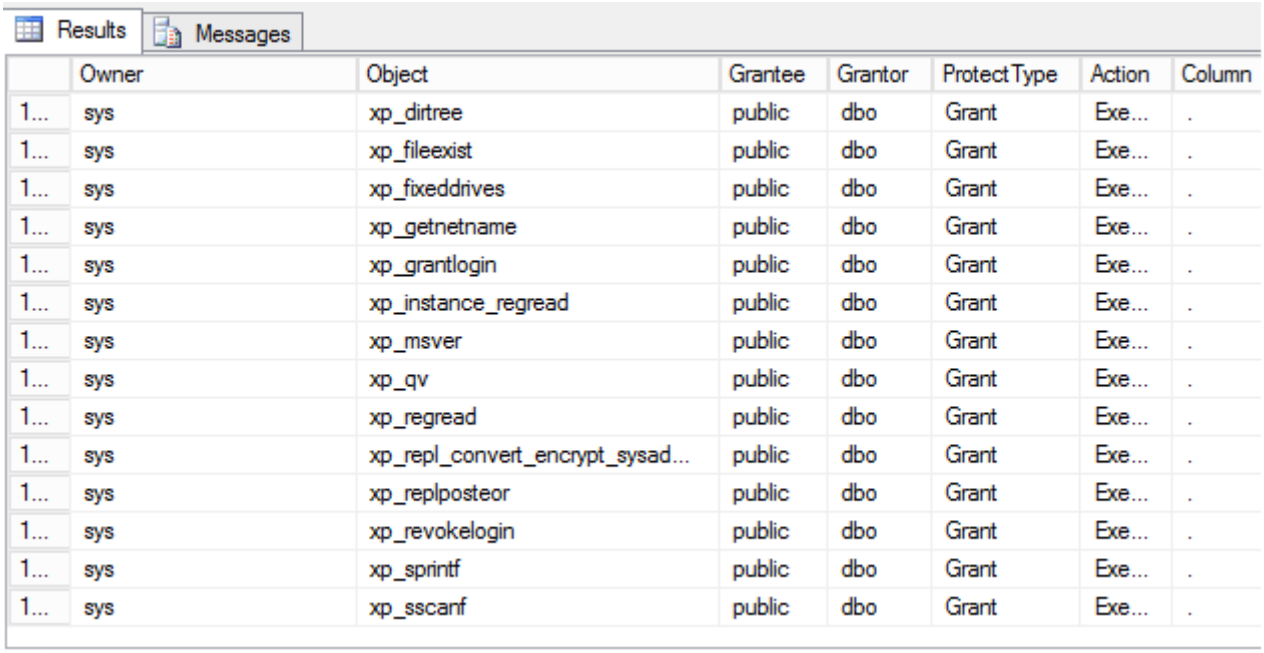
Рассмотрим варианты возможного развития атаки при следующем сценарии реализации атак на СУБД АСУ промышленными объектами. Злоумышленник при помощи фишинговой рассылки получил учетную запись пользователя АСУ промышленными объектами компании. При помощи этой учетной записи подключился по VPN к ресурсам компании и может подключиться к СУБД АСУ промышленными объектами с минимальными привилегиями «по умолчанию». В подобных случаях, злоумышленниками распространено использование хранимых процедур для получения НСД, как к СУБД, так к самой АСУ промышленными объектами. Хранимые процедуры, это объекты БД, которые представляют собой набор SQL-инструкций, которые компилируются однократно и хранятся на сервере самой СУБД. Как правило, производители не уделяют должного внимания при разграничении привилегий для выполнения хранимых процедур.

С целью анализа типовой конфигурации СУБД MS SQL Server версий 2005, 2008 R2, 2012, на предмет использования хранимых процедур обычными пользователями АСУ промышленными объектами СУБД, проведено исследование данных СУБД.

Итак, описание полученных результатов стоит начать в обратном порядке версий. Как правило все пользователи СУБД входят в группу ‘public’. Для получения информации о том, к каким объектам имеет доступ группа ‘public’, как правило используется следующий запрос:

```
sp_helprotect @username = 'public'
```

Выполнив данный запрос, в результирующий список попадают все объекты СУБД, к которым имеет доступ группа ‘public’, в соответствии с рисунком А.3. Хранимые процедуры имеют префикс ‘xp_’, а расширенные хранимые процедуры ‘sp_’.



Results | Messages

	Owner	Object	Grantee	Grantor	Protect Type	Action	Column
1...	sys	xp_dirtree	public	dbo	Grant	Exe...	.
1...	sys	xp_fileexist	public	dbo	Grant	Exe...	.
1...	sys	xp_fixeddrives	public	dbo	Grant	Exe...	.
1...	sys	xp_getnetname	public	dbo	Grant	Exe...	.
1...	sys	xp_grantlogin	public	dbo	Grant	Exe...	.
1...	sys	xp_instance_regread	public	dbo	Grant	Exe...	.
1...	sys	xp_msver	public	dbo	Grant	Exe...	.
1...	sys	xp_qv	public	dbo	Grant	Exe...	.
1...	sys	xp_regread	public	dbo	Grant	Exe...	.
1...	sys	xp_repl_convert_encrypt_sysad...	public	dbo	Grant	Exe...	.
1...	sys	xp_replposteor	public	dbo	Grant	Exe...	.
1...	sys	xp_revokelogin	public	dbo	Grant	Exe...	.
1...	sys	xp_sprintf	public	dbo	Grant	Exe...	.
1...	sys	xp_sscanf	public	dbo	Grant	Exe...	.

Рисунок А.3 - Результат выполнения запроса

Проанализировав каждую процедуру с данным префиксами, был составлен список процедур с их описанием, ограничение доступа к которым наиболее важно. Для MS SQL Server 2012 перечень представлен в таблице А.1.

Важность ограничения доступа к выделенным хранимым процедурам обусловлена их функциональными особенностями. Нужно отметить, что «по умолчанию», в группу ‘public’ входят все пользователи. Например, в случае с публичным доступом к наиболее критичным процедурам, с точки зрения безопасности, xp_regdeletekey, xp_regdeletevalue, xp_regenumvalues, xp_regremovemultistring, xp_regwrite, пользователи группы ‘public’ могут менять ветки реестра ОС, на которой запущен SQL-сервер.

Таблица А.1 - Хранимые процедуры MS SQL Server 2012

Имя процедуры	**Описание процедуры**
xp_availablemedia	возвращает список физических дисков, подключенных к серверу
xp_dirtree	используется для получения списка всех подкаталогов для текущего каталога
xp_enumerrorlogs	отображает журналы ошибок, используемые MS SQL-сервером
xp_enumgroups	предоставляет список локальных групп MS Windows или список глобальных групп, определенных в указанном домене MS Windows
xp_fixeddrives	используется для получения информации о дисковых накопителях, подключенных к серверу, и свободном месте на них
xp_getnetname	возвращает сетевое имя сервера, что может позволить злоумышленнику подобрать имена других машин сети
xp_logevent	регистрирует предназначенные пользователю сообщения в журнале MS SQL Server
xp_loginconfig	возвращает сведения о конфигурации безопасности входа в систему для экземпляра MS SQL Server
xp_msver	возвращает расширенную информацию об MS SQL-сервере
xp_readerrorlog	позволяет просматривать журнал ошибок MS SQL-сервера

Имя процедуры	**Описание процедуры**
xp_servicecontrol	используется для запуска, остановки, приостановки и возобновления работы служб MS Windows
xp_sprintf	используется для преобразования нескольких строк в одну, также может быть использована для создания исполняемых команд
xp_sscanf	используется для извлечения переменных из текстовой строки в произвольном формате, дает возможность злоумышленнику создавать исполняемые команды
xp_subdirs	возвращает все подкаталоги указанного каталога
sp_OACreate	создаёт экземпляр OLE-объекта
sp_OADestroy	удаляет созданный OLE-объект
sp_OAGetErrorInfo	получает данные об ошибке OLE-автоматизации
sp_OAGetProperty	получает значение свойства объекта OLE
sp_OAMethod	вызывает метод OLE-объекта
sp_OASetProperty	устанавливает новое значение свойства OLE-объекта
sp_OAStop	останавливает серверную среду выполнения хранимых процедур OLE-автоматизации
xp_regaddmultistring	позволяет добавлять в системный реестр ключ типа reg_multi_sz
xp_regdeletekey	позволяет удалить ключ системного реестра
xp_regdeletevalue	позволяет удалить установленное значение ключа системного реестра
xp_regenumvalues	возвращает значения указанного ключа системного реестра

Имя процедуры	Описание процедуры
xp_regremovemultistring	позволяет удалить из системного реестра ключ типа reg_multi_sz
xp_regwrite	позволяет изменять значение ключей реестра
xp_grantlogin	используется для предоставления доступа к SQL-серверу пользователям или группам ОС Windows
xp_logininfo	используется для отображения пользователей и групп SQL-сервера
sp_getbindtoken	возвращает уникальный идентификатор для транзакции
xp_revokelogin	позволяет запретить доступ к SQL-серверу пользователям и группам Windows

В случае с процедурой xp_servicecontrol, пользователем возможно отключение служб ОС; с xp_grantlogin, xp_logininfo, xp_revokelogin возможно предоставление/ограничение доступа к SQL-серверу, без административных полномочий. Вследствие чего, возможно нарушение работоспособности SQL-сервера в частности и ОС в целом. Исходя из этого, доступ ко всем вышеописанным хранимым процедурам рекомендуется ограничить, для группы доступа ‘public’ и для любых других кастомных, в которые входят обычные пользователи. Для того, чтобы ограничить доступ к объекту, наиболее приемлемо использование следующего запроса:

```
DENY execute on OBJECT::xp_availablemedia to public;
```

где ‘xp_availablemedia’ – хранимая процедура, а ‘public’ имя группы соответственно. Результатом запроса должно быть сообщение, в соответствии с рисунком А.4, об успешном выполнении.

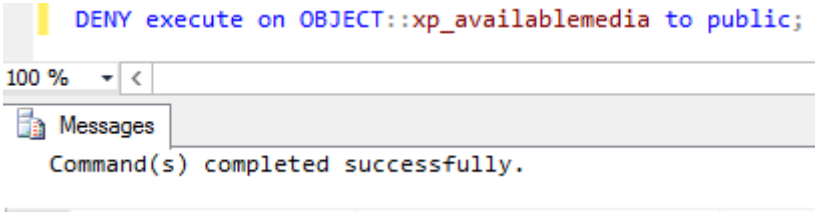


Рисунок А.4 - Результат выполнения запроса

Для проверки доступа к отдельной процедуре, либо для проверки ограничения доступа рекомендуется использовать следующий запрос:

```
sp_helprotect @name= 'xp_availablemedia',@username = 'public'
```

где метод sp_helprotect – показывает привилегии для объекта, ‘xp_availablemedia’ – сам объект (хранимая процедура), а ‘public’ имя группы соответственно. В результате выполнения появится соответствующая таблица с привилегиями, в соответствии с рисунком А.5.

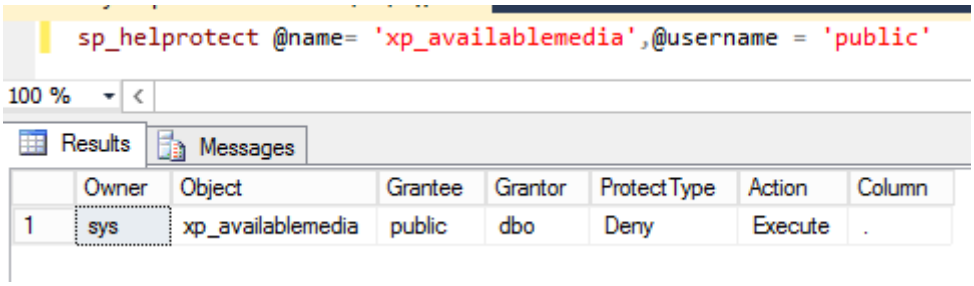


Рисунок А.5 - Таблица привилегий

В свою очередь, исследование MS SQL Server 2008 R2, показало, что все хранимые процедуры MS SQL Server 2012, так же доступны группе ‘public’ и были обнаружены не менее критичные хранимые процедуры, описание которых представлено в таблице А.2.

Аналогично MS SQL Server 2012, рекомендуется ограничивать доступ к вышеописанным процедурам. Запрос на ограничение доступа, в вышеописанном виде, так же применим к версии MS SQL Server 2008 R2.

Таблица А.2 - Хранимые процедуры

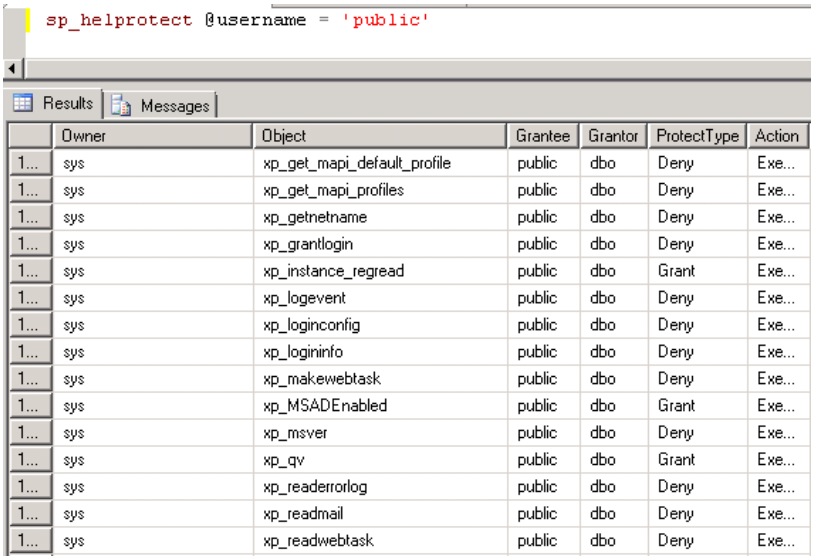
MS SQL Server 2008R2

Имя процедуры	**Описание**
xp_deletemail	Удаляет сообщение из папки входящих сообщений MS SQL Server. Эта процедура используется процедурой sp_processmail для обработки почты в папке входящих сообщений MS SQL Server
xp_findnextmsg	Принимает идентификатор сообщения в качестве входных данных и выдает идентификатор сообщения в качестве выходных данных. Обработка почты в почтовом ящике MS SQL Server осуществляется с помощью хранимых процедур xp_findnextmsg и sp_processmail
xp_get_mapi_default_profile	возвращает профиль MAPI, используемый по умолчанию
xp_get_mapi_profiles	возвращает результирующий список используемых системой профилей MAPI
xp_readmail	Считывает электронное сообщение из входящего почтового ящика службы MS SQL Mail. Эта процедура используется процедурой sp_processmail для обработки всех почтовых сообщений, находящихся во входящего ящике службы MS SQL Mail
xp_sendmail	Отправляет электронные сообщения, которые могут содержать вложение результирующего набора запроса, указанным получателям. Эта

Имя процедуры	Описание
	расширенная хранимая процедура использует службу MS SQL Mail для отправки сообщений
xp_startmail	Запускает сеанс клиента службы MS SQL Mail. При запуске почтового сеанса открываются клиентские компоненты MAPI и выполняется регистрация на сервере электронной почты
xp_stopmail	Останавливает сеанс клиента службы MS SQL Mail. Останавливает почтовый сеанс, открываемый компонентами клиента MAPI, и завершает регистрацию на почтовом сервере

Исследование MS SQL Server 2005, показало, что группе ‘public' доступны хранимые процедуры обнаруженные, как в MS SQL Server 2012, так и в MS SQL Server 2008 R2. Аналогично были обнаружены наиболее «старые» процедуры, к которым имеет доступ группа ‘public’, перечень которых приведён в таблице А.3. Несмотря на то, что версия СУБД достаточно стара и на то, что в официальном описании к продукту описана невозможность использования метода ‘sp_helprotect’ – метод функционирует. Использование данного метода представлено на рисунке А.6. Если функция не работает, есть возможность использования аналогов: представлением каталога sys.database_permissions и функцией fn_builtin_permissions.

В результате практического применения предложенного подхода уменьшается возможный вектор атак злоумышленниками с применением хранимых процедур продуктов MS SQL, за ограничения доступа к хранимым процедурам группам «по умолчанию».



	Owner	Object	Grantee	Grantor	ProtectType	Action
1...	sys	xp_get_mapi_default_profile	public	dbo	Deny	Exe...
1...	sys	xp_get_mapi_profiles	public	dbo	Deny	Exe...
1...	sys	xp_getnetname	public	dbo	Deny	Exe...
1...	sys	xp_grantlogin	public	dbo	Deny	Exe...
1...	sys	xp_instance_regread	public	dbo	Grant	Exe...
1...	sys	xp_logevent	public	dbo	Deny	Exe...
1...	sys	xp_loginconfig	public	dbo	Deny	Exe...
1...	sys	xp_logininfo	public	dbo	Deny	Exe...
1...	sys	xp_makewebtask	public	dbo	Deny	Exe...
1...	sys	xp_MSADEnabled	public	dbo	Grant	Exe...
1...	sys	xp_msver	public	dbo	Deny	Exe...
1...	sys	xp_qv	public	dbo	Grant	Exe...
1...	sys	xp_readerrorlog	public	dbo	Deny	Exe...
1...	sys	xp_readmail	public	dbo	Deny	Exe...
1...	sys	xp_readwebtask	public	dbo	Deny	Exe...

Рисунок А.6 - Использование метода sp_helprotect в MS SQL Server 2005

Таблица А.3 - Хранимые процедуры

MS SQL Server 2005

Имя процедуры	**Описание**
xp_enumcodepages	используется для получения списка всех кодовых страниц, наборов символов и их описаний
xp_cleanupwebtask	предназначена для выполнения произвольных вызовов(очистка) внутри системы MS SQL-сервером
xp_convertwebtask	предназначена для выполнения произвольных вызовов(конвертация) внутри системы MS SQL-сервером
xp_readwebtask	предназначена для выполнения произвольных вызовов(чтение) внутри системы MS SQL-сервером
xp_dropwebtask	необходима для удаления задания, созданного при помощи процедуры xp_makewebtask
xp_makewebtask	еобходима для создания задания, используемого для экспорта данных из табличного вида в HTML-

Имя процедуры	Описание
	файл. Может быть использована для операций с данными посредством сети
xp_runwebtask	необходима для запуска задания, созданного при помощи процедуры xp_makewebtask

Совершенствование методов повышения надежности инструментов удалённого администрирования сервисов обеспечения работоспособности АСУ промышленными объектами. Следующим вектором атак на АСУ промышленными объектами - являются атаки, направленные на средства удалённого администрирования сервисов АСУ промышленными объектами. В случаях, когда средства удалённого администрирования узлов АСУ промышленными объектами опубликованы в сеть Интернет, возникают дополнительные риски проникновения злоумышленниками в сеть АСУ промышленными объектами. Типичная рекомендация не публиковать сервисы удалённого администрирования в сеть Интернет, может не работать в ситуациях, когда других средств администрирования нет, а объект находится далеко от близлежащих населённых пунктов. При этом периметровый интерфейс АСУ промышленными объектами может не иметь статического IP-адреса либо может быть скрыт за NAT поставщика услуг Интернет. В данном разделе речь пойдет о SSH.

Решить данную проблему можно несколькими способами, например перенаправлением портов (NAT). Но NAT может быть крайне сложным в случае, если используется оборудование с несколькими вложенными правилами NAT. В дополнении подобное решение может быть ограничено поставщиком Интернет-услуг: ограничение правилами FireWall; блокировка перенаправления портов или трансляцией NAT адресов. Для экономии внешних IPv4 адресов. Но данные решение не безопасны в случаях прямой публикации средств удалённого администрирования.

Альтернативным решением обсуждаемой проблемы может быть обратное SSH-туннелирование. Механизм работы SSH-туннеля достаточно прост. Необходим хост, который имеет внешний статический IP-адрес, в идеале расположить его на внешнем хостинге или в облачном хостинге. Доступ к серверу предлагается использовать через консоль администрирования браузера и ограничить входящие подключения, связанные с прямым удалённым администрированием. Затем, необходимо настроить реверс SSH-туннель, от целевого оборудования, до хоста со статическим IP-адресом. Т.к. происходит подключение в обратном направлении туннель называется обратным. Подключившись к хосту со статическим IP-адресом по протоколу SSH, целевое оборудование будет доступно до тех пор, пока доступен этот хост.

В качестве примера практической реализации, представим, что имеется удалённый сервер ExtSrv с динамическим внешним адресом и сервер с внешним адресом (8.8.8.8) RelaySrv. Настроим обратный туннель от удаленного сервера ExtSrv к серверу RelaySrv. Настройку выполним таким образом, чтобы можно было получать доступ по протоколу SSH к ExtSrv через RelaySrv с любого оборудования, поддерживающего протокол SSH и находящегося в сети Интернет.

На сервере ExtSrv открываем соединение SSH к серверу RelaySrv командой:

```
# ExtSrv~$ ssh -fN -R 20022:localhost:22 user@8.8.8.8
```

где порт 20022 является свободным портом с номером и диапазона 1024–65535. Нужно отметить, что этот порт не должен использоваться программным обеспечением на RelaySrv.

Параметр "-R 20022:localhost:22" определяет реверс SSH-туннель. Параметр делает перенаправление трафик с порта 20022 сервера RelaySrv на порт 22 на сервере ExtSrv. Параметр "-fN" позволяет SSH использовать в

фоновом режиме сразу, как только пройдена успешно проверка подлинности на сервере по протоколу SSH. Этот параметр полезен в случае, если не нужно на удаленном сервере SSH выполнять какие-либо команды, а как в данном случае, только использовать перенаправление портов.

После выполнения вышеуказанной команды произойдет возврат обратно к командной строке сервера RelaySrv. Затем, необходимо подключиться к серверу RelaySrv и убедиться, что 127.0.0.1:20022 привязан к демону SSHD. Для проверки необходимо выполнить следующую команду:

```
# ExtSrv ~$ sudo netstat -nap | grep 20022
```

Результатом выполнения вышеуказанной команды и верной настройки должна быть строка:

```
# tcp      0    0 127.0.0.1:20022    0.0.0.0:*       LISTEN      8493/sshd
```

После настройки, чтобы получить доступ к удаленному серверу ExtSrv достаточно с консоли сервера RelaySrv выполнить нижеуказанную команду:

```
# RelaySrv~$ ssh -p 20022 user@localhost
```

Стоит отметить, что логин/пароль, набираемый для localhost, должен быть для сервера ExtSrv, а не для сервера RelaySrv, т.к. осуществляется вход через SSH-туннель. Выполнив успешно подключение к ExtSrv, станут доступны все команды данного сервера.

В результате практического применения предложенного механизма уменьшается возможный вектор атак злоумышленниками на сервисы удалённого администрирования АСУ промышленными объектами, поскольку сервисы не публикуются, а используется механизм обратного туннелирования.

Совершенствование методов повышения надежности при публикации сервисов АСУ промышленными объектами в сеть Интернет. В связи с

постоянным развитием систем передачи информации, вычислительных средств и информационных технологий в целом ни одна из компаний не обходится без публикации внутренних сервисов АСУ промышленными объектами в сеть Интернет. Небезопасная публикация сервисов АСУ промышленными объектами в сеть Интернет [20,21], может быть использована злоумышленниками для получения НСД к узлам сети АСУ промышленными объектами. Именно небезопасная публикация – рассмотрена, как потенциальный вектор атак на АСУ промышленными объектами, представленный в данном разделе.

Рассмотрим типичную архитектуру для обеспечения минимального уровня защищенности сервиса АСУ промышленными объектами, опубликованного в сети Интернет, на примере публикации WEB-ресурса [49]. Типовая архитектура включает в себя межсетевой экран в любой доступной реализации и сервер приложений (в данном случае backend-сервер), как представлено на рисунке А.7.

Здесь backend-сервер не имеет прямого подключения к сети Интернет, т.к. использован межсетевой экран, позволяющий блокировать нелегитимный трафик и скрыть локальный IP-адрес backend-сервера. Подобная архитектура используется часто, но, в случае компрометации межсетевого экрана, злоумышленник попадает в корпоративную сеть передачи данных и дальше уже ничто не ограничивает развитие его атаки.

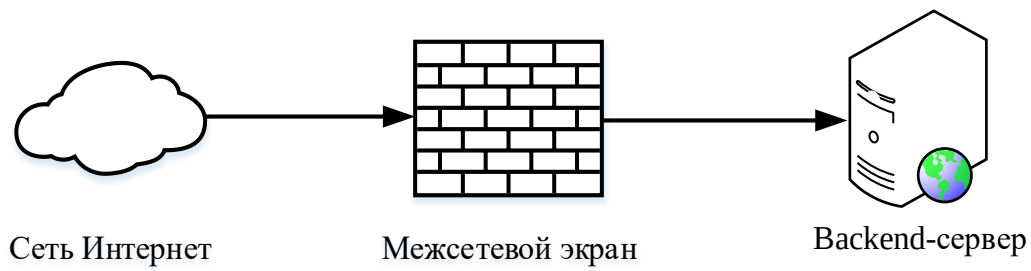


Рисунок А.7 - Типовая архитектура

Более защищенной является архитектура с размещением в демилитаризованную зону сети (DMZ)[1] backend-сервера (или любого другого сервера, с которого необходимо опубликовать сервис) и межсетевого экрана, в соответствии с рисунком А.8.

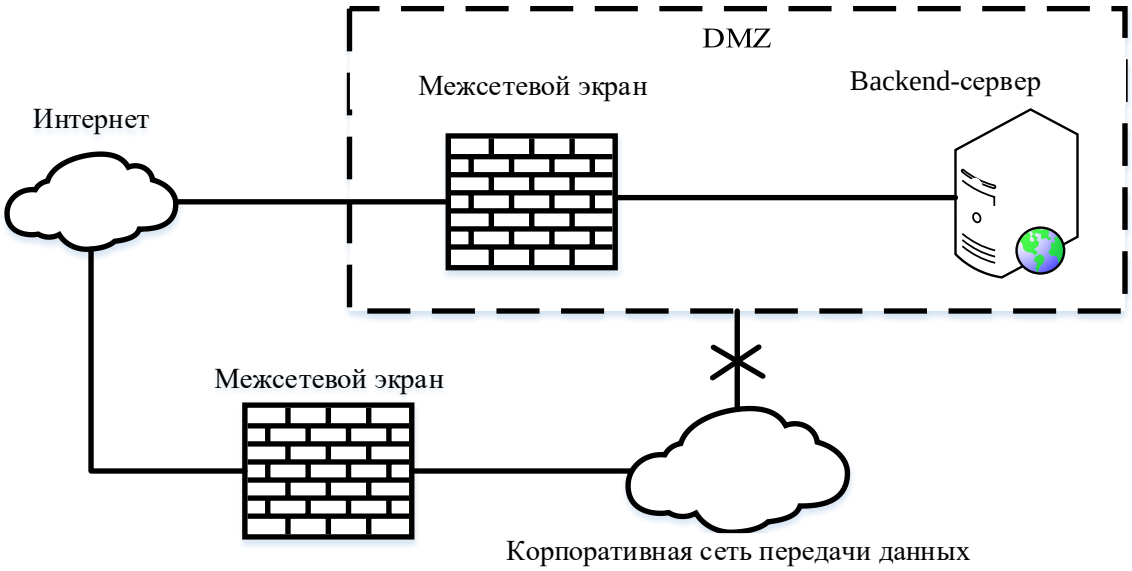


Рисунок А.8 - Архитектура с использованием DMZ

В данном случае, если злоумышленнику удастся получить доступ к межсетевому экрану и, в том числе, backend-серверу, дальнейшее развитие атаки становиться проблематичным, поскольку прямого доступа в сеть предприятия из этого сегмента нет. Но остаётся проблема предоставления доступа к опубликованному ресурсу пользователям изнутри корпоративной сети, у которых отсутствует доступ в сеть Интернет.

Следующая архитектура представляет из себя два backend-сервера (первый с reverse-proxy[2], второй с обычным функционалом, как в 1 и 2 вариантах), два межсетевых экрана (с NAT) без использования DMZ и с

[1] Сегмент сети с белой адресацией, отделённый от сети Интернет и локальной сети организации.
[2] Обратный прокси – промежуточный сервер, который принимает запросы клиентов и ретранслирует их на сервера в корпоративной сети передачи данных, затем передаёт ответы обратно клиентам.

разделением сетевых сегментов на уровне VLAN либо на уровне сетевых диапазонов, представленная на рисунке А.9.

Для простоты реализации, в предложенной архитектуре использовано 4 разных сетевых сегмента с соответствующей IP-адресацией, которая не пересекается между сегментами: Сегмент, состоит из внешнего диапазона адресов, в этом диапазоне находятся IP-адреса, которые не входят в выделенные IANA локальные (немаршрутизируемые в Интернет) диапазоны 10.0.0.0/8; 100.64.0.0/10; 172.16.0.0/12; 192.168.0.0/16;

Диапазон 1, адреса из сети 192.168.1.0/24.

Диапазон 2, адреса из сети 192.168.2.0/24.

Диапазон 3, адреса из сети 192.168.3.0/24.

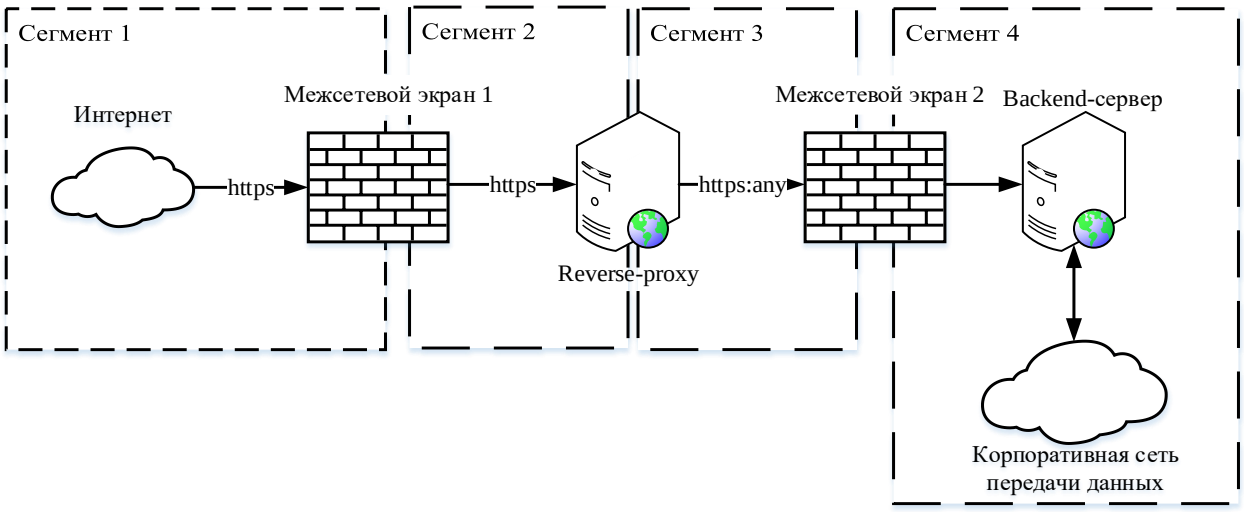


Рисунок А.9 - Архитектура с использованием reverse-proxy

Важно отметить, что оборудование настроено таким образом, что каждый сегмент изолирован друг от друга, а связь между сегментами осуществляется за счет 2 сетевых адаптеров, на каждом из узлов. Допускается пересечение IP-адресов, но в случае пересечения - усложняется логирование и администрирование подобной архитектуры в целом.

В свою очередь, работа данной архитектуры представлена следующим образом:

1. Пользователь открывает браузер и переходит по ссылке, ведущей к заранее опубликованному сервису, например, https://example.com. Получает IP-адрес от DNS-сервера, и в результате формируется запрос к серверу по протоколу https на порт 443.
2. Пришедший к серверу запрос создаёт сессию между клиентом и первым межсетевым экраном. Затем запрос предварительно обрабатывается межсетевым экраном, поскольку на нём настроено правило трансляции адресов (NAT) для всех клиентов, с внешнего IP-адреса и порта 443, на внутренний IP-адрес и порт 443. Как следствие, это NAT-правило срабатывает, и далее трафик запроса попадает в сеть с reverse-proxy.
3. Затем reverse-proxy обрабатывает полученный запрос, попутно делает подмену URL-адреса внутри запроса с https://example.com на https://example.local.comodo.com и пропускает запрос дальше.
4. Следующим обработку начинает второй межсетевой экран, на котором так же используются правила NAT для трансляции адресов из одного сетевого сегмента в другой.
5. И, наконец, запрос доходит до backend-сервера с URL-адресом https://example.local.comodo.com.
6. В обратную сторону для отправки ответа клиенту схема работает точно так же, только reverse-proxy преобразует URL-адрес в первоначальный вид, который получен при запросе пользователем.

Несмотря на явный недостаток в необходимости дополнительного оборудования для реализации данной архитектуры, сама архитектура имеет ряд преимуществ перед другими:

- Уменьшение нагрузки, поскольку большинство реализаций reverse-proxy может равномерно распределять запросы на backend-серверы.
- Обеспечение безопасности, так как reverse-proxy может скрывать существование опрашиваемых им backend-серверов.

- Увеличение производительности, так как при правильно выбранном reverse-proxy имеется возможность управлять статическим контентом (изображениями, CSS и т.п.).
- Простота аудита и логирования с использованием единой точки доступа.

Как правило, обеспечение должного уровня безопасности усложняет жизнь конечному пользователю. Однако, предложенная третья архитектура абсолютно «прозрачна» для пользователей, а потому не создает им проблем.

В результате практического применения предложенной архитектуры уменьшается возможный вектор атак злоумышленниками на сервисы АСУ промышленными объектами, опубликованные в сети Интернет. Следовательно, несмотря на сложность и дополнительные затраты на оборудование, можно рекомендовать для защиты сервисов АСУ промышленными объектами, опубликованных в Интернет именно ее.

Совершенствование методов повышения надежности сервисов АСУ промышленными объектами с использованием инструментов регулярного аудита. Одной из причин появления критичных уязвимостей в АСУ промышленными объектами является быстрый рост информатизации в целом [73]. Недостаточная разработанность мер по должному обеспечению мер ИБ, а также пренебрежение общепринятым принципам ИБ негативно влияют на деятельность множества организаций. Решению задач поиска обеспечения должного технического [16] и документационного [3,55] уровней ИБ уделяется большое внимание как в РФ, так и в за рубежом. Однако проблемы, связанные с поиском уязвимостей именно в АСУ промышленными объектами, проработаны недостаточно полно [53].

С целью совершенствования методов защиты Web-интерфейсов АСУ промышленными объектами [20,21], в том числе, основанных на аппаратной базе Simatic S7, путём поиска потенциальных векторов атак на АСУ промышленными объектами, сформирован перечень возможных действий по исследованию их уязвимостей и мерам защиты. Для достижения цели возможно использование следующих методов:

1. Периметровый пентест. Осуществление несанкционированного доступа (НСД) во внутреннюю сеть со стороны сети Интернет с последующим развитием атаки во внутренней сети и анализ защищенности беспроводных сетей. В зависимости от требований заказчика может быть принято решение об уведомлении ответственных за объект исследования администраторов.
2. Внутренний пентест. Исследование системы с автоматизированного рабочего места (АРМ) среднестатистического пользователя сети. Учитывая требования заказчика, может быть выбран один или несколько сегментов сети.
3. Тестирование отдельных компонентов АСУ промышленными объектами. Получение доступа к определенным ресурсам и определенным данным.
4. Оценка осведомлённости сотрудников компании в вопросах ИБ. Различные анкетирования сотрудников, заполнение опросных листов, заочные обследования человеческого фактора, возможно, посредством видеонаблюдения. Как правило, этот подход менее интересен заказчику ввиду того, что ИБ с точки зрения человеческого фактора должны обеспечивать соответствующие службы (служба безопасности).

В большинстве случаев, должны быть использованы один или несколько вышеуказанных методов [64].

Итак, проникновение «извне» в сеть предприятия считается одной из самых сложных задач [5,6,61,90]. Это обусловлено высоким классом защиты оборудования, находящегося на передовой линии защиты (межсетевые экраны и др.) и подготовкой специалистов обеспечивающих ИБ на предприятии в целом. Следовательно, для осуществления проникновения необходимо большое количество опыта и наличие немалого количества ресурсов у атакующих. На стадии планирования обязательно должен быть сформирован потенциальный перечень исследуемых ресурсов. Этими ресурсами могут быть как корпоративный сайт, так почтовые и другие внешние сервисы. Как

правило, согласно этой информации, начинается дальнейшее планирование действий по проникновению:

1. Поиск хостов, находящихся в тех же подсетях, что и ресурсы, указанные заказчиком;
2. Получение информации с найденных активных хостов (версии, названия сервисов, открытые порты и тд.);
3. Использование сканеров уязвимостей;
4. Поиск эксплойтов и их использование;
5. Поиск учетных записей со стандартной парой логин-пароль, грубый перебор этой пары с использованием словарей;
6. Поиск перебором подозрительных директорий и файлов (для Web-серверов) и др.

Если ничего найти не удалось, и все вышеописанные операции ни к чему не привели, возможны следующие варианты:

- Синжерство или фишинг: отправка зловреда на корпоративный email. Получатель, перейдя по ссылке, скачивает зловреда, который сам просканирует внутреннюю сеть и предоставит удаленный доступ к корпоративной сети передачи данных (КСПД) через инфицированную рабочую станцию пользователя;
- При помощи поисковых машин (google, yandex, yahoo и др.) или специализированных поисковых систем [58] (shodan) собрать список поддоменов ключевого ресурса компании (site.com). Это позволит получить историю компании. Если компания ранее пользовалась различными услугами (Web-хостинг и пр.) других компаний, то есть возможность найти эту информацию, а затем развивать атаку уже согласно найденной истории, начиная с пункта 1 общего плана. Выполнив все пункты общего плана по поддоменам, в случае удачи и получения доступа к одному из них, формируется большая база учетных записей с паролями, которые можно попробовать применить к панелям управления действующих ресурсов. В большинстве случаев один из

найденных ресурсов будет в локальной сети либо будет иметь «следы», ведущие к ней.

Воспользовавшись вышеуказанными действиями, можно сформировать перечень возможных векторов атак направленных на получение НСД к КСПД или АСУ промышленными объектами, сервисам организации, опубликованным в сети Интернет. Далее следующая стадия тестирования - «Внутренний пентест», с использованием инструментов для внутреннего тестирования. Как правило, при внутреннем пентесте, доступ к КСПД, в которой находится целевая АСУ промышленными объектами или оборудование предоставлен по умолчанию. Укрупненный план тестирования выглядит следующим образом:

1. Сканирование диапазона адресов на наличие активного оборудования (компьютеры, оргтехника, сетевое оборудование, контроллеры и др.);
2. Получение информации о найденном оборудовании (имя, версия ОС, MAC и пр.);
3. По списку найденного оборудования сканируются открытые/используемые порты разных диапазонов, стандартные - до 1023, остальные до 65535, производится поиск доступных сервисов, их версий;
4. Использование сканеров уязвимостей;
5. Выбор и использование эксплойтов к целевому оборудованию, согласно полученным данным на 2 и 3 шаге, также здесь можно провести.

Говоря о сканировании сети, нужно отметить, что существует множество сетевых сканеров, умеющих решать задачи, поставленные перед сетевыми сканерами. Были исследованы только самые актуальные и часто используемые.

IP tools – представляет собой набор утилит для работы с стеком TCP/IP в одной программе. Поддерживает версии ОС семейства MS Windows. Интерфейс ПО представлен на рисунке А.10.

Исследование заключалось в сканировании всего пула /24 IP-адресов (254 адресов) локальной сети тестового стенда, состоящего из 4 активных хостов, на которых были запущены ОС MS Windows и Linux/Unix системы. Исследование IP tools показало, что утилиты, включенные в состав данного ПО, работают крайне медленно (отчет приходится собирать «руками»), в сравнении с Linux дистрибутивами, но, тем не менее, необходимые задачи в плане сетевого сканирования данное ПО решает.

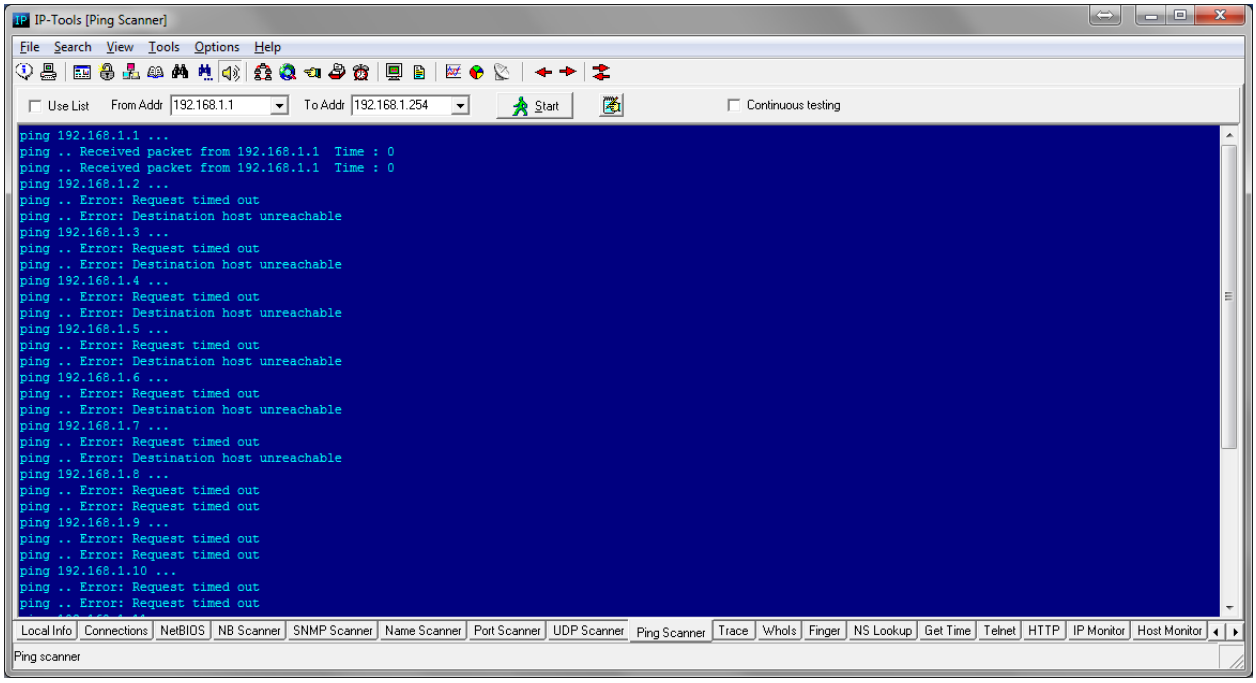


Рисунок А.10 - Сканирование сети при помощи IP tools

NMAP (Network Mapper) - бесплатный аналог IP Tools с открытым исходным кодом, поддерживается большинством Linux/Unix дистрибутивов. Работает значительно быстрее предыдущего ПО и быстро развивается. Данный инструмент использует сырые IP-пакеты оригинальными способами, чтобы определить какие хосты доступны в сети, какие службы (название приложения и версию) они предлагают, какие операционные системы (и версии ОС) они используют, какие используются типы пакетных фильтров/брандмауэров и еще множество других характеристик [30]. К

несвойственным обычным сетевым сканерам возможностям можно отнести наличие скрипта для проверки узла на наличие червя Stuxnet. В тех же условиях, при которых тестировался IP tools, NMAP показал длительность сканирования в 7.34 секунды, результат представлен на рисунке А.11. При этом были получены мини-отчеты о хостах, содержащие информацию о задержке, открытых портах, службах, которые используют этот порт и MAC-адресах и кратком комментарии о состоянии системы.

Одним из самых больших плюсов NMAP является возможность использования в автоматическом тестировании. Т.е. NMAP проанализирует сеть, получит список хостов, предоставит информацию о необходимом хосте, открытых на нем портах и установит наличие доступных для обращения сервисов, а Metasploit Framework который определит возможные эксплойты и позволит их использовать.

Для автоматизированного и более «удобного» использования уязвимостей используются эксплойты, которым передаются только параметры для использования брешей в безопасности. В случае необходимости применения различных эксплойтов к большому количеству оборудования, ПО, скриптов и т.д., ручное использование становится трудоемким и нерациональным. Поэтому чаще всего приходится использовать специальные инструменты автоматизации со значительной базой имеющихся эксплойтов:

The Metasploit Framework – инструмент, позволяющий использовать большое количество эксплойтов, исходя из класса целевого сервиса (SSH, FTP, HTTP и др.) или автоматизировать их выполнение. Как было отмечено ранее, данный инструмент хорошо работает в связке с NMAP и, аналогично ему, поддерживается большинством Linux/Unix дистрибутивов. Для Metasploit есть GUI (Armitage) именно при помощи него и выполнялось исследование. Получив список активных хостов в сети и информацию о них при помощи NMAP, выбрав среди указанного списка хостов «целевой», можно начать поиск и дальнейшее применение эксплойтов (соответствующие самому

целевому хосту или его сервисам). Интерфейс с найденными вариантами атак для целевого хоста представлен на рисунке А.12.

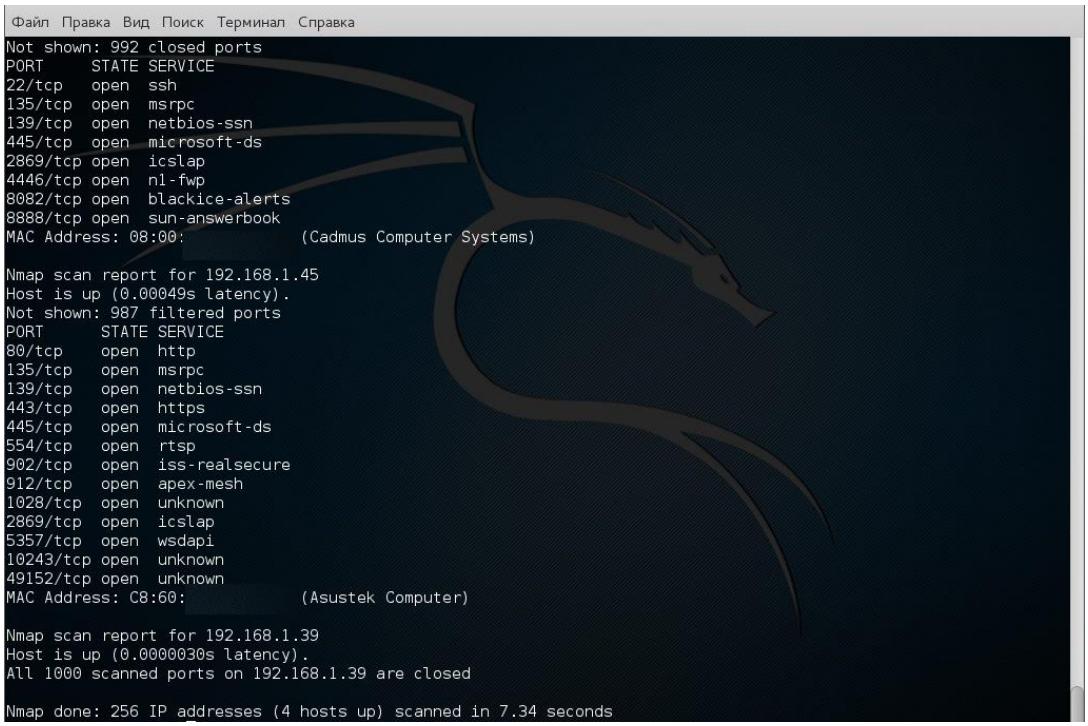


Рисунок А.11 - Сканирование сети с NMAP

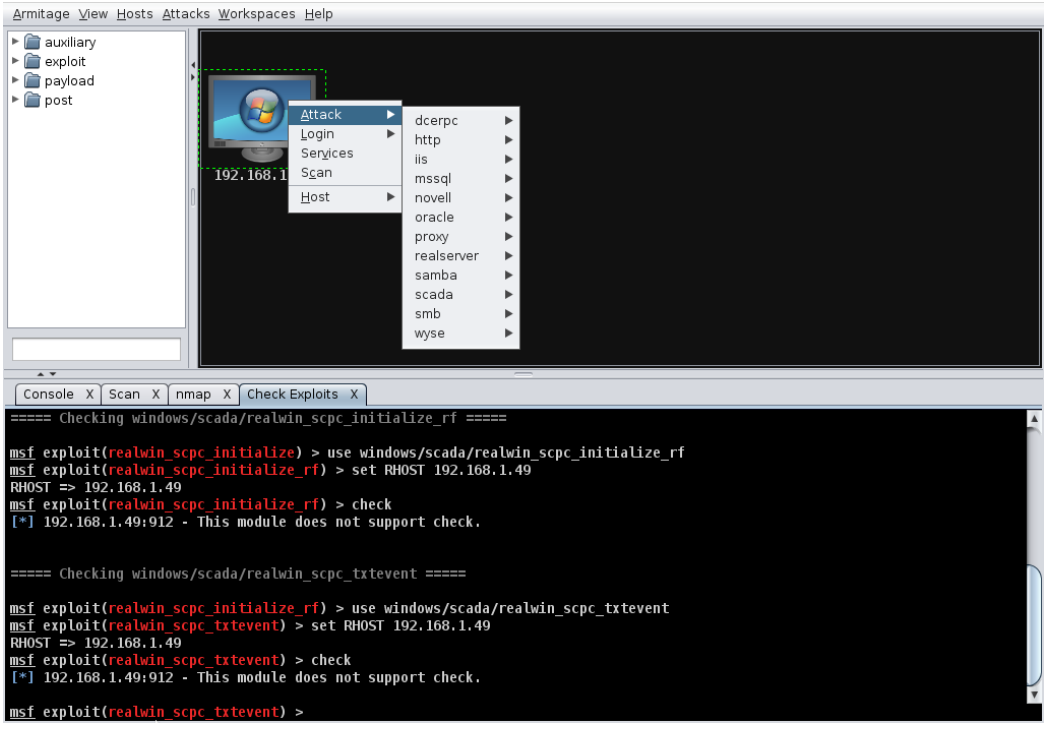


Рисунок А.12 - Применение эксплойтов в Armitage

В результате исследования целевого хоста и попыток применения эксплойтов к нему с платформенной ОС MS Windows Server 2008 и развернутым WinCC Web Navigator'ом был получен отчет представленный на рисунке А.12 -. Данный отчет содержал информацию о том, подвержен ли действительно выбранным эксплойтам целевой хост. Так же имеется возможность «грубой» проверки сразу всех эксплойтов.

OpenVAS (OpenSource Vulnerability Scanner) – бесплатный инструмент работающий с клиент-серверной архитектурой, где все операции выполняются серверной частью, поддерживается большинством Linux/Unix дистрибутивов. Данный инструмент использует движок Nessus 2 и часть плагинов этого проекта. Интерфейс сканера представлен на рисунке А.13.

Vulnerability		Severity	QoD	Host	Location	Actions
MS15-034 HTTP.sys Remote Code Execution Vulnerability (remote check)		10.0 (High)	95%	192.168.1.49	80/tcp	
OpenSSL CCS Man in the Middle Security Bypass Vulnerability		6.8 (Medium)	99%	192.168.1.49	443/tcp	
DCE Services Enumeration		5.0 (Medium)	75%	192.168.1.49	135/tcp	
DCE Services Enumeration		5.0 (Medium)	75%	192.168.1.49	135/tcp	
SSL Certification Expired		5.0 (Medium)	75%	192.168.1.49	443/tcp	
Microsoft IIS Default Welcome Page Information Disclosure Vulnerability	Mitigation	4.6 (Medium)	80%	192.168.1.49	80/tcp	
Deprecated SSLv2 and SSLv3 Protocol Detection		4.3 (Medium)	99%	192.168.1.49	443/tcp	
POODLE SSLv3 Protocol CBC ciphers Information Disclosure Vulnerability		4.3 (Medium)	75%	192.168.1.49	443/tcp	
TCP timestamps		2.6 (Low)	75%	192.168.1.49	general/tcp	
OS fingerprinting		0.0 (Log)	70%	192.168.1.49	general/tcp	
arachni (NASL wrapper)		0.0 (Log)	75%	192.168.1.49	general/tcp	
Traceroute		0.0 (Log)	75%	192.168.1.49	general/tcp	
Microsoft SMB Signing Disabled		0.0 (Log)	75%	192.168.1.49	general/tcp	
CPE Inventory		0.0 (Log)	75%	192.168.1.49	general/CPE-T	
SMB Test		0.0 (Log)	75%	192.168.1.49	general/SMBClient	
Windows SharePoint Services detection		0.0 (Log)	80%	192.168.1.49	80/tcp	
HTTP Server type and version		0.0 (Log)	75%	192.168.1.49	80/tcp	
DIRB (NASL wrapper)		0.0 (Log)	75%	192.168.1.49	80/tcp	

Рисунок А.13 - Стандартный отчет OpenVAS

Помимо плагинов для анализа защищенности, полученных от проекта Nessus, в OpenVAS интегрировано немало утилит: Nikto для поиска уязвимых CGI-сценариев, NMAP, ike-scan для обнаружения IPSEC VPN узлов, amap для идентификации сервисов на портах и многое другое [30]. Проведя

исследование данного инструмента, было выявлено, что наличие Web-интерфейса на серверной части значительно упрощает работу с данным инструментом. В зависимости от поставленных задач, инсталляция серверной части может быть произведена как на сервере в Интернет, при необходимости проникновения извне, так и на сервере в локальной сети. Проведя единожды инсталляцию серверной части, зная IP-адрес сервера можно свободно пользоваться браузером для различного рода тестирований или автоматизировать тестирования. Исследование начиналось с добавления хоста или списка хостов для сканирования (можно воспользоваться не интегрированной версией NMAP) сразу после добавления начинается быстрое сканирование целевого хоста.

В результате сканирования на уязвимости целевого хоста, был получен отчет представленный на рисунке А.13, содержащий название уязвимости, уровень критичности, действиях, которые выполнялись для выявления уязвимости, и ссылку на исправление критичных уязвимостей от производителя.

Nessus – инструмент, ключевой особенностью которого является то, что любой тест на проникновение не зашивается наглухо внутрь программы, а оформляется в виде подключаемого плагина. Плагины распределены на сорок два различных множество различных типов: для проведения тестирования, для активации как отдельных плагинов, так и всех плагинов одновременно определенного типа – например, все проверки определенного семейства ОС. При этом отсутствуют ограничения по написанию собственных пентестов, т.к. в Nessus реализован специальный скриптовый язык NASL (Nessus Attack ScrIPting Language) [30]. Любой тест начинается с создания правил, которых сканер будет придерживаться все время сканирования. При формировании этих правил выбираются виды сканируемых портов, количество одновременных подключений, в том числе, безопасное сканирование (исключаются плагины, которые могут нанести вред системе) и другие,

типичные для данной утилиты опции. Web- интерфейс сканера представлен на рисунке А.14.

Каждый из исследованных инструментов умеет решать огромный спектр задач, при этом большинство из них распространяются на коммерческой основе. Стоит отметить, что большинство утилит используемых для тестирования написаны под Linux/Unix системы. Говоря о бесплатных инструментах, стоит отметить OpenVAS, единственный минус инструмента - ограничение в периодичности обновлений баз. OpenVAS не найдет все бреши в защите, но сможет указать на большую известную их часть.

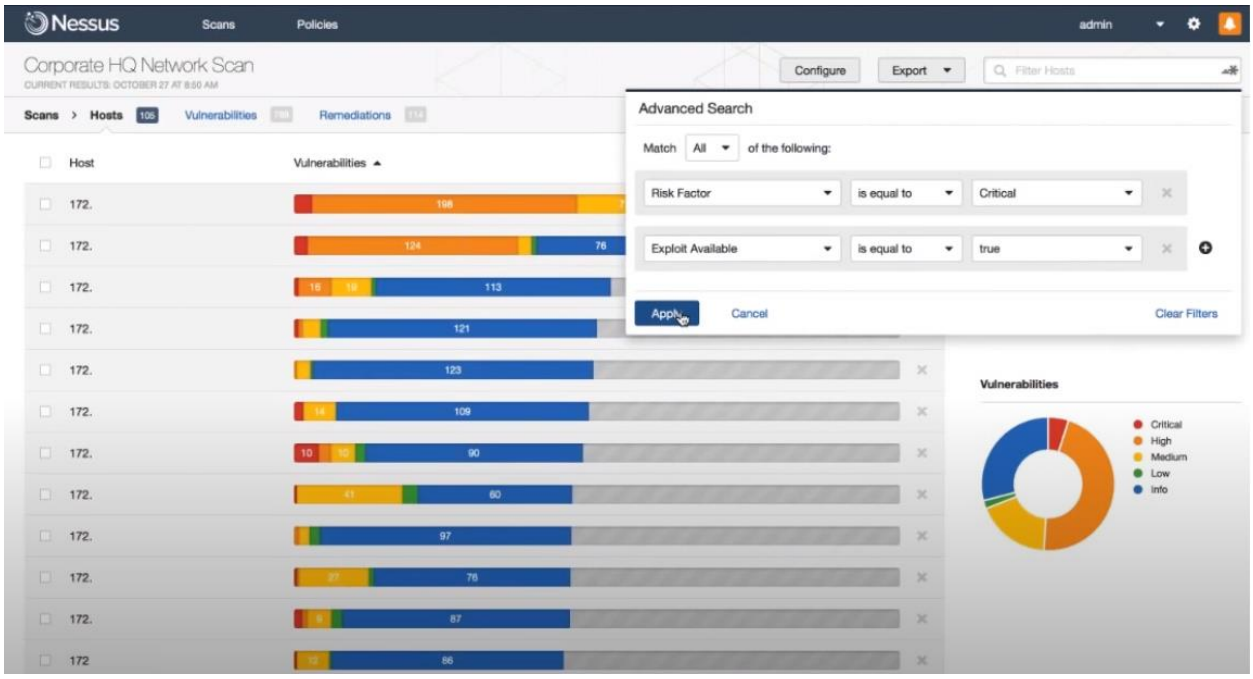


Рисунок А.14 - Стандартный отчет Nessus

Говоря об ОС семейство MS Windows, нужно отметить их собственное решение Microsoft Baseline Security Analyzer. Это анализатор безопасности, проверяющий все хосты с системами ОС семейства MS Windows в сети на соответствие требованиям к ИБ компании установленные в соответствии с лучшими практиками компании Microsoft. Сканирование проводится исключительно в режиме «белого ящика». Одним из основных критериев

проверки является наличие всех установленных обновлений в системе. Перед началом сканирования утилита проверяет наличие обновлений для своих баз и затем исследует целевой хост. В результате будет получен сводный отчет, представленный на рисунке А.15. Отчет показывает отсутствие установленных критических обновлений на активных хостах сети, тем самым отражает возможность применения эксплойтов к ним. Также в отчете отражена ссылка на необходимые обновления.

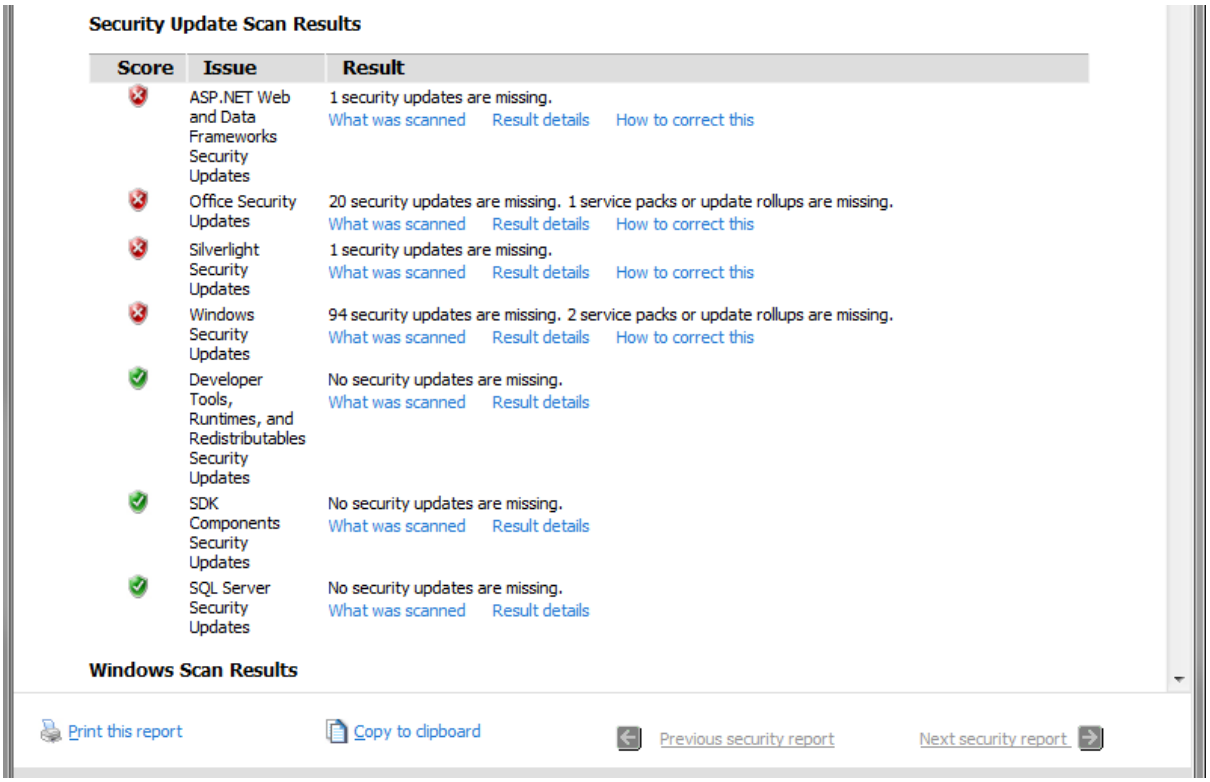


Рисунок А.15 - Отчет Microsoft Baseline Security Analyzer

Итак, нужно отметить, что получение доступа к сервисам WEB, VNC, telnet, SSH, FTP, TFTP, SFTP на контроллерах и рабочих станциях по умолчанию не имеет ограничения по количеству попыток ввода пароля. Часто в качестве паролей доступа персонал использует стандартные или широко распространенные пароли.

Рассмотрим подробнее авторизацию на WEB-интерфейсы, на примере Siemens Simatic S7. Использование данного сервиса на этих контроллерах может быть различным: Установка на рабочую станцию (ПК или сервер) WinCC Web Navigator; Включение опции самотестирования на контроллере HMI MiniWeb on; VNC - аналогичный Web Navigator'у доступ к управлению SCADA-системой, за исключением некоторых особенностей.

Для работы с Web Navigator'ом авторизация проходит через механизм базовой авторизации (Basic Auth). Это наиболее простой вид идентификации. Этот вид идентификации один из самых уязвимых, т.к. форма не содержит никакой дополнительной информации для проверки легитимности отправляемых запросов (не читаются заголовки запросов и дополнительная информация в виде версии браузера, ос и т.д.), а сами запросы от пользователя отправляются в незашифрованном виде (на уровне рабочей станции). Исследовав попытки не легитимной авторизации при условии использования Basic Auth на промышленных системах, выявили, что сами попытки проходят очень быстро.

Исследование заключалось в получении НСД к тестовой машине (клон снятый с реального производства), базирующейся на ПО WinCC Web Navigator v7.0 и платформе MS Windows Server 2012. Для чистоты эксперимента воспользовались NMAP. Получив список активных хостов и доступных портов сети, была найдена тестируемая машина с открытым 80 портом. Затем, методом получения НСД был выбран - подбор связки «логин-пароль» по загруженному в специализированное ПО словарю (2 стандартных логина и 100 различных, часто применяемых паролей). В результате исследования получили связки логин/пароль, при помощи которых возможен НСД к исследуемой системе, так же нужно отметить, что всего использовано 200 попыток авторизоваться, попытки использовались в 2-х потоках, общее время проведения эксперимента 16 секунд.

В свою очередь, для работы с системой самотестирования HMI MiniWeb on, авторизация проходит несколько иным способом. Идентификация также

базируется на HTTP, за основу берется механизм Basic Auth, но при этом, учетные данные вводятся не в отдельном окне, а на самой форме. Соответственно, изменяется содержание POST/GET запросов. Исходя из того, что реализация этой самой формы может быть различна, тем самым усложняются нелегитимные попытки авторизации. Но, к сожалению, усложнение попыток авторизации не всегда возможно, так как стандартные средства защиты, такие как капча (Captcha), столь часто используемые в идентификации Интернет-ресурсов, не принято применять для промышленных систем. Виной тому использование стандартных форм авторизации (HTTP, Basic Auth и др) с невозможностью их изменения, т.к. они зашиты в прошивку или в саму службу. Разумеется, не все вендоры используют формы, разработанные под своё ПО. Те, кто позволяют форму изменить, сталкиваются с другой проблемой. Зачастую вся сеть АСУ промышленными объектами, оказывается изолированной от интернета, за исключением демилитаризованной зоны (хотя допускается изоляция DMZ тоже), вследствие ограниченности значительно затрудняется использование капчи и поддержание её актуальных баз. Проведённые нами исследования показывают, что попытки нелегитимной авторизации при условии использования механизма авторизации HTTP на промышленных системах проходят относительно не быстро.

Исследование заключалось в получении НСД к WEB-интерфейсу тестовой SCADA/HMI-системы самотестирования (служба была включена на приборе, собранном в тестовом стенде). Аналогично первому исследованию, была найден IP-адрес исследуемой системы, а для получения НСД использовался аналогичный первому метод и словарь, за исключением некоторых деталей. Изменены тексты GET/POST запросов, в соответствии с используемыми для HTTP механизмами авторизации. В результате исследования получили связки логин/пароль, при помощи которых возможен НСД к исследуемой системе, в целом было использовано 200 попыток

авторизации, отметим общее время на все попытки 130 секунд, при использовании 2-х потоков.

Аутентификация вида VNC посредством использования порта 5800 (Siemens Simatic S7) и значительно отличается от предыдущих. Это связано с использованием java-машины, за счет которой проходит авторизация и управление системой. Здесь для авторизации необходимо подобрать всего лишь пароль – логин не требуется. Но в связи с тем, что протокол HTTP используется только для запуска java-апплета и присутствует шифрование трафика, подбор значительно усложняется.

Исследование механизма VNC заключалось в попытке использовании стандартного пароля от единой учетной записи к SCADA/HMI-системе самотестирования, исследуемой во втором эксперименте и двум другим, найденным в Интернет. Найденные системы были протестированы в «ручном» режиме. Результат превзошел все ожидания. Подобрав нужную версию java-машины, к 2м системам из 3-х был получен НСД с первой попытки со стандартным паролем [28].

Для противодействия обнаруженным векторам, рекомендуется применение следующих мер [17]:

- использование правильно настроенного firewall, используемого как шлюз, не дающего сделать более 5 get/post запросов за определённый период времени с одного IP/MAC-адреса, это значительно усложнит нелегитимные действия;
- настройка групповых политик безопасности АСУ промышленными объектами в целом при адекватной настройке firewall, рекомендуется менять пароли раз в 30-60 дней, это обусловлено тем, что расшифровка хэшей более продолжительная;
- использование криптостойких паролей со спецсимволами и своевременный их аудит (исследователями из университета Глазго были использованы интеллектуальные алгоритмы, которые предварительно были обучены на базе данных, представляющей 10 млн паролей,

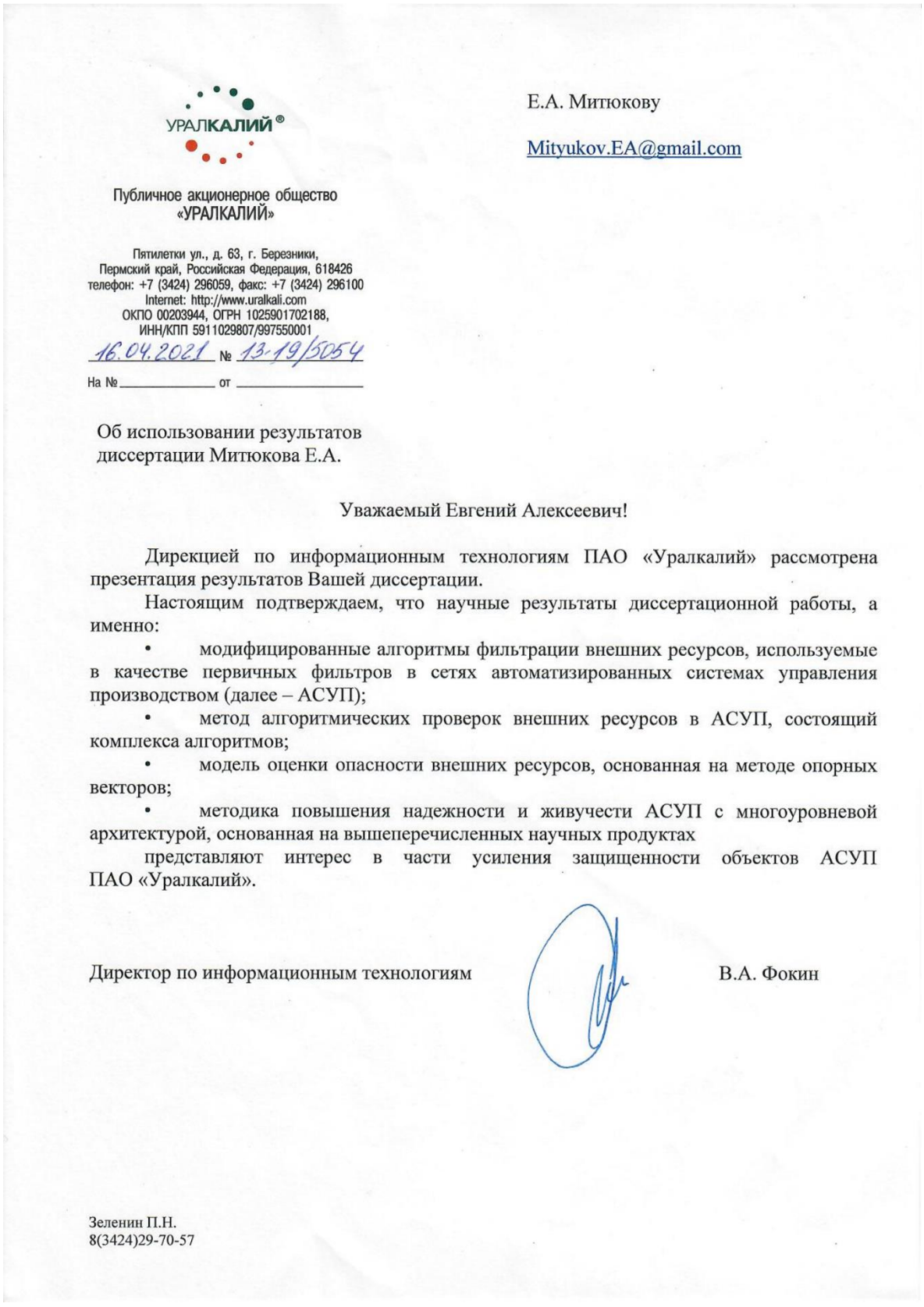
имеющихся в сети в открытом виде. Далее они проверили эффективность алгоритмов на 32 млн других паролей. Выяснилось, что цифры и символы верхнего регистра не позволяют усложнить пароль. Такого эффекта можно достичь удлинением пароля или использованием специальных символов) [96];

- использование двухфакторной аутентификации (стандартная связка логин+пароль и привязка сотового телефона, на который высылается смс).

В результате практического применения предложенных подходов и инструментов, в значительной степени снижаются потенциальные векторы атак на сервисы АСУ промышленными объектами, как в КСПД, так и опубликованные в сети Интернет.

Приложение В

Копия письма о внедрении результатов диссертационной работы



УРАЛКАЛИЙ®

Публичное акционерное общество
«УРАЛКАЛИЙ»

Пятилетки ул., д. 63, г. Березники,
Пермский край, Российская Федерация, 618426
телефон: +7 (3424) 296059, факс: +7 (3424) 296100
Internet: http://www.uralkali.com
ОКПО 00203944, ОГРН 1025901702188,
ИНН/КПП 5911029807/997550001

16.04.2021 № 13-19/5054

На № _____ от _____

Е.А. Митюкову

Mityukov.EA@gmail.com

Об использовании результатов
диссертации Митюкова Е.А.

Уважаемый Евгений Алексеевич!

Дирекцией по информационным технологиям ПАО «Уралкалий» рассмотрена презентация результатов Вашей диссертации.

Настоящим подтверждаем, что научные результаты диссертационной работы, а именно:

- модифицированные алгоритмы фильтрации внешних ресурсов, используемые в качестве первичных фильтров в сетях автоматизированных системах управления производством (далее – АСУП);
- метод алгоритмических проверок внешних ресурсов в АСУП, состоящий комплекса алгоритмов;
- модель оценки опасности внешних ресурсов, основанная на методе опорных векторов;
- методика повышения надежности и живучести АСУП с многоуровневой архитектурой, основанная на вышеперечисленных научных продуктах

представляют интерес в части усиления защищенности объектов АСУП ПАО «Уралкалий».

Директор по информационным технологиям В.А. Фокин

Зеленин П.Н.
8(3424)29-70-57

Приложение С

Копия письма о внедрении результатов диссертационной работы

BioNT
ВАША ИНФОРМАЦИЯ
ПОД НАДЕЖНОЙ ЗАЩИТОЙ

ЗАО «Бионт»

Краснова, 24, корпус 1, Пермь, 614015
тел. (342) 206-38-00, факс (342) 237-73-35, info@biont.ru, biont.ru
ОКПО 26604586, ОГРН 1025900889167, ИНН/КПП 5906000464/590401001

26 мая 2020 г. Исх. № 222

ФГБОУ ВО "Тамбовский государственный технический университет
Ученому секретарю диссертационного совета Д 212.260.01
Шишкиной Г.В.
392000, г. Тамбов, ул. Советская, д. 106

20 мая 2021 года на техническом совете ЗАО "Бионт" была рассмотрена презентация результатов диссертации Митюкова Евгения Алексеевича.

Настоящим подтверждаю, что научные результаты диссертационной работы, полученные Митюковым Е.А., а именно:

- модифицированные алгоритмы фильтрации внешних ресурсов, используемые в качестве первичных фильтров в сетях промышленных автоматизированных систем управления (далее – ПАСУ);
- метод алгоритмических проверок внешних ресурсов, состоящий комплекса алгоритмов;
- модель оценки опасности внешних ресурсов, основанная на методе опорных векторов;
- методика повышения надежности и живучести ПАСУ с многоуровневой архитектурой, основанная на вышеперечисленных научных продуктах

представляют интерес в части усиления защищенности объектов ПАСУ, безвозмездно переданы исполнителем и приняты к использованию в ЗАО «Бионт».

Генеральный директор Н.А. Григоров

ЗАКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО БИОНТ г.Пермь

Приложение D

Копия АКТА внедрения результатов диссертационной работы в учебный процесс БФ ПНИПУ

УТВЕРЖДАЮ
Директор БФ ПНИПУ
_____ Косвинцев О.К.
"18" июня 2021 г.

АКТ

внедрения результатов диссертации Митюкова Евгения Алексеевича «Повышение надежности автоматизированных систем управления промышленными объектами путем совершенствования уровня их информационной безопасности» в Березниковском филиале Федерального государственного автономного образовательного учреждения высшего образования «Пермский национальный исследовательский политехнический университет»

Мы, нижеподписавшиеся профессор кафедры «Автоматизации технологических процессов», к.т.н. Беккер В.Ф. и к.т.н., доцент кафедры «Автоматизации технологических процессов» Плехов П.В., настоящим актом удостоверяем внедрение материалов диссертации Митюкова Е.А. «Повышение надежности автоматизированных систем управления промышленными объектами путем совершенствования уровня их информационной безопасности» в научной деятельности и учебном процессе в дисциплинах: «Хранение и защита компьютерной информации» направления подготовки 15.03.04 «Автоматизация технологических процессов и производств» и «Безопасность и защита информации в распределенных автоматизированных системах» направления подготовки 09.04.01 «Информатика и вычислительная техника» кафедры «Автоматизации технологических процессов» в Березниковском филиале Федерального государственного автономного образовательного учреждения высшего образования «Пермский национальный исследовательский политехнический университет».

В ходе выполнения диссертации Митюков Е.А. являлся постановщиком задач и руководителем разработки программного обеспечения, реализующего методы и алгоритмы информационной защиты автоматизированных систем управления предприятиями, в рамках студенческой научно-исследовательской лаборатории кафедры «СНИЛ ИТ», в результате чего было разработано программное обеспечение, используемое в учебном процессе, опубликовано несколько статей студентов, в том числе, в рецензируемых журналах РИНЦ.

Существенные теоретические результаты, практические примеры, алгоритмы и программное обеспечение, полученные в ходе работы над диссертацией, используются при преподавании дисциплины «Защита информации» для направления «Информатика и вычислительная техника» (бакалавриат).

Заместитель заведующего кафедрой АТП, к.т.н., профессор Беккер В.Ф.

Доцент кафедры АТП, к.т.н. Плехов П.В.