МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра компьютерных систем

Направление подготовки: 230100 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Магистерская программа: Безопасность и защита информации

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов

Барчан Константин Андреевич

Тема диссертации утверждена распоряжением по НГУ №531 от «14» декабря 2012г.

Тема диссертации скорректирована распоряжением по НГУ №109 от «20» марта 2014г.

«К защите допущена»

Заведующий кафедрой,

к.т.н., с.н.с. КТИ ВТ СО РАН

Пищик Б. Н. /…….

(фамилия , И., О.) / (подпись, МП)

«……»……2014г.

Научный руководитель

Зам. директора ООО “Системы информационной безопасности”, к.т.н., доцент каф. САПР СибГУТИ

Гончаров С. А./…….....

(фамилия , И., О.) / (подпись, МП)

«……»……2014г.

Дата защиты: «……» июня 2014г.

Автор: Барчан К.А./……...

Новосибирск, 2014 г.

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра компьютерных систем.

Направление подготовки: 230100 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Магистерская программа: Безопасность и защита информации

УТВЕРЖДАЮ

Зав. кафедрой Пищик Б. Н.
(фамилия, И., О.)
…….
(подпись, МП)
«……»……2014г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

Студенту Барчан Константину Андреевичу
Тема: Разработка метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов.
Цель работы: разработать систему обнаружения вторжений для SCADA-систем, обладающую возможностью превентивного обнаружения внедрения вирусного ПО или деятельности злоумышленника.

Структурные части работы: работа включает в себя изучение особенностей архитектуры и функционирования АСУ ТП (SCADA-систем), используемых технологий, специфики обеспечения их ИБ; обзор и выявление недостатков современных методов и технологий по обеспечению ИБ СУ; выявление актуальных проблем ИБ СУ; разработку метода имитационного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов; спецификацию модели акторов с учётом особенносей SCADA; реализацию прототипа системы обнаружения вторжений; анализ полученных результатов.

Научный руководитель

Зам. директора ООО “Системы информационной безопасности”, к.т.н., доцент каф. САПР СибГУТИ

Гончаров С. А. /…….....

«……»……2014г.

Задание принял к исполнению

Барчан К. А. /……..

«……»……2014г.

Оглавление

ВВЕДЕНИЕ

Благодаря всевозрастающей значимости информационных активов в современном деловом мире и государственном секторе, АСУ ТП и, в частности, SCADA-системы уже сейчас занимают существенную позицию в инфраструктуре производственных предприятий и критически важных объектов промышленности. Столь важная структурная единица информационной системы того или иного предприятия не может обходиться без должного внимания с точки зрения информационной безопасности.

Анализ текущего состояния научного и производственного секторов в данной сфере с учётом временной динамики позволил отметить положительную тенденцию к всестороннему увеличению внимания по отношению к проблемам ИБ данной области. Однако, несмотря на некоторые локальные улучшения и исследования в области ИБ АСУ ТП (SCADA-систем), говорить о факте приемлемой защищённости в их отношении не приходится. Причиной этому является ряд нерешённых на данный момент проблем информационной безопасности, а именно:

а) Физическая изоляция (т.н. “воздушный зазор”) сетей перестала быть эффективной мерой информационной безопасности.
б) Уязвимость от атак изнутри корпоративной сети злоумышленным служащим (т.н. “insider”).
в) СЗИ, спроектированная без учёта специфики SCADA, может оказывать негативное влияние на технологический процесс и информационный обмен.

Учитывая вышеизложенные проблемы ИБ АСУ ТП и SCADA-систем, становится возможным сделать вывод о том, что упомянутые ранее подходы в области исследований средств, методов и технологий обеспечения информационной защиты систем управления обладают рядом недостатков, а именно:

а) Отсутствие возможности заблаговременного (превентивного) обнаружения атаки или внедрения вредоносного ПО.
б) Наличие дополнительной вычислительной нагрузки по обеспечению функционирования средств информационной защиты, возлагаемой на важные технологические компоненты инфраструктуры систем управления.

Целью данной работы является разработка СОВ для SCADA-систем, обеспечивающей возможность превентивного обнаружения внедрения вирусного ПО или действий злоумышленника. Прямых аналогов подобной системы на данный момент не существует.

В процессе работы были достигнуты следующие результаты:

а) Изучены особенности архитектуры и функционирования АСУ ТП (SCADA-систем), используемые технологии, специфика обеспечения их информационной безопасности.

б) Проведён обзор и анализ современных методов и технологий по обеспечению ИБ СУ, выявивший актуальные проблемы ИБ объекта исследования и недостатки подходов по её обеспечению.

в) Специфицирована структура модели акторов для работы со SCADA-системами (для обеспечения оперативной оценки состояния).

г) На основе специфицированной модели акторов разработана единая модель SCADA-системы и её СОВ, описано взаимодействие компонентов модели.

д) Разработан метод ИМ эталонного состояния и поведения SCADA-систем для СОВ и его математическая основа.

е) Реализована модель сетевого взаимодействия ПЛК (PLC) и человеко-машинного интерфейса (HMI) по протоколу TCP/IP.

ж) Реализован модуль сбора сетевой статистики.

з) Реализован модуль анализа состояния сети.

и) Реализован модуль анализа настроек оборудования.

1 Обзор предметной области

1.1 Автоматизированная система управления технологическим процессом (АСУ ТП)

Современные АСУ ТП представляют собой совокупность программно-аппаратных средств и технологий, предназначенных для автоматизации управления производственным оборудованием и технологическими процессами на различных объектах промышленного производства, а именно: заводах, фабриках и предприятиях. Зачастую АСУ ТП имеет тесную связь с более общим компонентом производственной инфраструктуры, называемым АСУП или ИСУПП.

В роли АСУ ТП, как правило, выступает некое комплексное решение по обеспечению автоматизации основных технологических процессов и их составляющих на уровне всего производства в целом или отдельной его структурной составляющей.

Необходимо отметить, что наличие на промышленном предприятии АСУ ТП отнюдь не исключает необходимость участия человека в отдельно взятых производственных операциях. Участие человека может осуществляться как в целях обеспечения необходимого уровня контроля над процессом, так и в связи со сложностью, невозможностью или нецелесообразностью проведения автоматизации отдельно выделенных операций.

В общем случае, система управления может быть рассмотрена в качестве набора связанных между собой процессов управления и объектов управления. Базовая цель автоматизации управления заключается в увеличении эффективности эксплуатации допустимых возможностей для объекта управления – технологического процесса. На основании данных фактов, становится возможным выделить ряд целей автоматизации управления технологическим процессом:

а) Предоставление сотруднику данных о процессе для принятия решений.
б) Повышение скорости исполнения отдельных операций, направленных на сбор и обработку данных.
в) Уменьшение числа решений, которые должен принимать сотрудник.
г) Повышение уровня производственного контроля и дисциплины исполнителей.
д) Повышение быстродействия и оперативности управления.
е) Увеличение степени обоснованности предпринимаемых мер и принимаемых решений [1-3].

1.1.1 Классификация и функционирование АСУ ТП

В рамках современных АСУ ТП выделятся несколько характерных подклассов систем, а именно: SCADA-системы, DCS-системы, а также ПЛК-контролируемые системы управления. Рассмотрим особенности каждого подкласса управляющих систем подробнее.

Типичная SCADA-система - это распределённая система, предназначенная для обеспечения управления территориально дифференцированными производственными или хозяйственными ресурсами (объектами). SCADA-системы включают элементы, обеспечивающие централизованное управление и производящих централизованный сбор данных. Данного класса системы используются в рамках распределённых систем (электрическое и водное снабжение, топливная и транспортная системы). На основании данных, полученных от удалённого оборудования, автоматизированное оборудование или оператор отдают управляющие команды периферийным устройствам (открытие и закрытие клапанов, получение данных с датчиков и сенсоров, а также анализ возникновения аварийных ситуаций).

Распределённые системы управления (DCS-системы) применяются в рамках широко распределённых отраслей промышленности и производства, таких как, например, производство продуктов массового потребления, пищевая, топливная и энергетическая промышленность. Подобные системы проектируются таким образом, чтобы их архитектура управления включала в себя некоторый уровень управления, на котором будет производиться анализ данных, получаемых от совокупности дочерних подсистем, обеспечивающих функционирование отдельных этапов производственного процесса.

Для управления производственными процессами и конечным оборудованием в системе устанавливаются ПЛК. Эти контроллеры входят в инфраструктуру SCADA и DCS-систем. На основе ПЛК также может быть создана небольшая локальная система управления производственным процессом (ПЛК-контролируемая подсистема управления), частично выполняющая функции SCADA или DCS-систем. В рамках сложных управляющих систем ПЛК участвуют в управлении отдельными локальными производственными или техническими процессами (например, сборка оборудования на конвейере).

Между перечисленными системами управления существует и ряд определённых различий. DCS-системы и ПЛК-контролируемые подсистемы, как правило, применяются в менее масштабных и централизованных отраслевых (промышленных) предприятиях по сравнению с крупными, территориально распределёнными производствами, использующими SCADA-системы. Система коммуникаций в распределённых и ПЛК-

контролируемых системах обычно строится с использованием локальных сетей (LAN), по причине того, что они обладают большей надёжностью и быстродействием в сравнении с удалёнными системами связи, используемыми в SCADA-системах. SCADA-системы разрабатываются с учётом необходимости работы с удалённым оборудованием и связанными с этим проблемами, такими как временные задержки в передаче информации и возможные потери данных. В отличие от SCADA-систем, остальные имеют централизованное управление процессами, так как задача контроля производства SCADA-системой является более сложной, нежели управление отдаленно взятыми процессами.

С точки зрения информационной безопасности вышеперечисленные отличия управляющих систем в рамках их классов можно считать несущественными, так как общие архитектурные принципы и решения, а также оборудование совпадают. Однако, наиболее сложной, многокомпонентной и масштабной из перечисленных управляющих систем является SCADA-система, поэтому данный класс систем заслуживает более подробного рассмотрения. Таким образом, в рамках данной магистерской работы будут исследоваться особенности систем диспетчерского управления и сбора данных (SCADA-систем) в области информационной безопасности, а также методы, технологии и способы её обеспечения [1-3].

1.2 Система диспетчерского управления и сбора данных (SCADA)

SCADA-система - программно-аппаратный комплекс, предназначенный для целей обеспечения работы в реальном времени систем сбора, анализа, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ, АСУЗ и т. д. SCADA-системы используются практически во всех отраслях промышленности и хозяйства, где существует необходимость обеспечения операторского контроля за технологическими процессами. Программное обеспечение SCADA использует для связи с объектом управления драйверы ввода-вывода или специализированные серверы и протоколы (Modbus, OPC/DDE).

В среде информационных технологий наиболее широко распространена трактовка понятия SCADA как приложения - программного комплекса, обеспечивающего выполнение управляющих функций, а также программно-инструментальных средств для разработки этого программного обеспечения. В рамках производства, телеметрии и информационной безопасности под SCADA-системой подразумевается программно-аппаратный комплекс.

В сферу задач, решаемых современными SCADA-системами, входят следующие пункты, а именно:

а) Обмен данными с “устройствами связи с управляемым объектом” (ПЛК, платы ввода/вывода и др.) и обработка получаемой информации.
б) Логическое управление контролируемым оборудованием.
в) Реализация удобного интерфейса, позволяющего оператору эффективно анализировать системные данные.
г) Ведение системы логирования, отчётов и базы данных с технологической информацией.
д) Синхронизация с системой аварийной сигнализации и управление сообщениями о тревоге и неполадках в системе.
е) Обеспечение коммуникации и интеграции с внешними приложениями (СУБД, система управления предприятием).

Структура современных SCADA-систем позволяют производить разработку АСУ ТП в клиент-серверной или (в зависимости от технологических и производственных требований) распределённой архитектуре [1, 2].

1.2.1 Компоненты управления SCADA

Список основных компонентов типичной SCADA-системы, подробно описанных в таблице А.1, содержит следующие пункты, а именно:

а) Управляющий сервер.

б) SCADA-сервер или Главный сетевой терминал (MTU).

в) Устройство связи с объектом (RTU) (RTU-устройства).

г) Программируемый логический контроллер (PLC, ПЛК).

д) Интеллектуальные электронные устройства (IED).

е) Человеко-машинный интерфейс (HMI).

ж) Журнал данных.

Рассмотрим стандартную схему взаимодействия вышеперечисленных компонентов. Локальные управляющие действия и команды выполняются RTU или ПЛК автоматически. Таким образом, непосредственное управление работой производственного процесса, как правило, обеспечивается средствами RTU или PLC, а вышестоящие элементы SCADA (HMI), управляемые инженером или оператором, регулируют режимы их работы. Цикл управления с обратной связью, как правило, задействует RTU-устройства или ПЛК, тогда как как SCADA-система обеспечивает контроль полного выполнения цикла. Сбор данных начинается на уровне RTU или ПЛК устройств на основе показаний измерительных приборов (датчиков, сенсоров и т.д.). Затем данные обрабатываются и переформировываются таким образом, чтобы оператор или инженер на HMI имел возможность провести анализ текущей ситуации и принять необходимое решение (изменение параметров, прерывание или продолжение управления процессом текущими средствами RTU/ПЛК). Производственные данные могут дополнительно быть записаны в архив для последующей аналитической обработки [3].

1.2.2 Компоненты сетей SCADA

Каждый сетевой уровень в рамках сетевой инфраструктуры SCADA-системы обладает своими собственными характеристиками. Объединение управляющих и корпоративных сетей позволяет сотрудникам проводить мониторинг и управление системами извне управляющей сети. Данная связь предоставляет возможность получения производственной информации для менеджеров высшего звена. Рассмотрим подробнее список основных компонентов SCADA-системы, не зависящий от используемой топологии сети:

а) Промышленная сеть. Промышленная сеть объединяет между собой сенсоры, датчики и элементы оборудования с ПЛК и другими контроллерами. Взаимное соединение устройств осуществляется посредством контроллера промышленной сети с использованием соответствующих протоколов.

б) Управляющая сеть. Управляющая сеть представляет из себя связующее звено между главным управляющим уровнем и низкоуровневыми контрольными модулями.

в) Маршрутизаторы. Маршрутизатор - это устройство для обеспечения коммуникации посредством передачи сигналов между двумя подсетями. Данное устройство используются в целях соединения сетей LAN с сетями WAN, а также соединения MTU-терминалов и RTU-устройств.

г) Межсетевой экран (Файервол). Межсетевой экран обеспечивает защиту сетевых устройств с помощью средств анализа и управления входящим и исходящим сетевым трафиком согласно заданным правилам и фильтрам.

д) Модемы. Модемы - это устройства, назначением которых является перевод цифровых данных в аналоговую форму для передачи по кабельной линии (например, телефонной линии). Модемы могут использоваться в инфраструктуре SCADA-систем для осуществления связи между MTU-терминалами и периферийными устройствами.

е) Точки удаленного доступа. Точки удаленного доступа – это сетевые устройства, а также территории или места управляющей сети, используемые в целях обеспечения удаленного контроля системами управления и доступа к данным о производственных процессах. Примером может служить процесс подключения к точке удалённого доступа ноутбука для удалённого доступа к SCADA-системе (АСУ ТП) [3].

1.2.3 Операционные системы, технологии, и протоколы передачи данных SCADA

Программное обеспечение для АСУ ТП (SCADA-систем) приобретается в составе готовых программных продуктов для систем управления, осваивается, анализируется и, при необходимости, адаптируется под конкретные нужды с использованием предоставляемых универсальных средств и инструментов расширения (API). Всему виной служит возрастание стоимости и сложности разработки прикладного ПО.

В основе большинства из ныне существующих SCADA-систем лежит платформа Microsoft Windows, так как подобного рода системы предлагают наиболее гибкие, полные и расширяемые решения в области HMI. Усиление позиций Microsoft на рынке ОС АСУ ТП влечёт соответствующую реакцию со стороны разработчиков SCADA-систем (например, Siemens) для множества различных платформ, которая заключается в приоритезации дальнейшего развития именно платформы Windows NT. Таким образом, основу глобального рынка программного обеспечения SCADA-систем на данный составляет ОС MS Windows NT, тогда как её предшественники, такие как MS DOS и MS Windows 3.xx/95, стремительно вытесняются. Использованию Windows также способствует отлаженный механизм коммуникаций с драйверами оборудования различных производителей, определяющий универсальность ОС.

Отдельную нишу занимают решения, основанные на операционных системах реального времени, таких как Windows CE, Windows Embedded Compact, QNX и др. ОС реального времени способны обеспечивать требуемое время выполнения задач реального времени, что является особенно актуальным в отношении систем АСУ ТП в плане реализации характеристик доступности и отказоустойчивости. Windows NT имеет ряд ограничений по обеспечению жёсткого реального времени (предпочтении аппаратного прерывания программному, отсутствие приоритетов по обработке процессов в очереди отложенных процедур и др.). Существенным недостатком SCADA-систем, основанных на платформе Windows, является отсутствие поддержки жесткого реального времени. Остальные ОС в рамках SCADA-систем представлены незначительно. Их основными преимуществами являются открытые исходные коды и свободная распространение.

Наибольшее распространение в среде SCADA получил язык C. Это обуславливается его высокой скоростью работы, независимостью от версии виртуальной машины или фреймворка, а также широкими возможностями для написания драйверов. Меньшее распространение получили такие языки, как C++, C# и Java [4, 5].

1.2.4 Средства расширяемости SCADA

Одной из неотъемлемых возможностей современных SCADA-систем является возможность расширения, позволяющая дополнять (расширять) и адаптировать готовый программно-аппаратный продукт под нужды и условия конкретного производства.

Среди используемых подходов к реализации возможности расширения системы управления можно выделить следующие пункты:

а) Самостоятельная разработка программных модулей**.** Возможность расширения (открытость системы) подразумевает доступность системных спецификаций**.** На практике данный функционал может представлять собой такие процедуры, как доступ к функциям UI, функциям работы с СУБД и т.д.;

б) Драйверы ввода-вывода. В современных SCADA-системах не существует ограничения на выбор аппаратуры нижнего уровня. Данная возможность реализуется благодаря наличию большого количества драйверов или серверов ввода-вывода, а также широких средств и возможностей по созданию собственных программных модулей или драйверов для новых устройств. Обычно для интеграции драйверов ввода-вывода со SCADA-системой используются механизмы стандартного динамического обмена данными (DDE - Dynamic Data Exchange), обмена по внутреннему закрытому (проприетарному) протоколу, включения и встраивания объектов OLE (Object Linking and Embedding) и OPC (OLE for Process Control, OLE для АСУТП).

в) Встраиваемые объекты ActiveX. Встраиваемые объекты ActiveX представляют из себя сущности, в основе которых лежит объектная модель компонентов Microsoft COM (Component Object Model). Технология COM – стандартная технология Microsoft для создания ПО, в основе которой лежит идея взаимодействия компонентов, каждый из которых может быть использован во множестве программах одновременно. В рамках технологии реализована инфраструктура, позволяющая объектам обмениваться данными между прикладными программами.

г) Разработка программных модулей третьей стороной. На рынке присутствует большое количество компаний, которые занимаются разработкой драйверов для различного оборудования, специальных объектов ActiveX и другого специализированного ПО для SCADA-систем [5].

1.2.5 Связь SCADA-систем с остальными системами предприятия

Как правило, в рамках стандартной инфраструктуры промышленного предприятия или объекта хозяйства АСУ ТП не является строго обособленной системой, исключённой из процесса информационного обмена. В целях увеличения уровня производительности, эффективности и автоматизации производственных процессов и рационального управляющего воздействия, а также повышения уровня анализируемости, информативности, доступности технологических данных и простоты использования средств и технологий производства, системы управления (например, SCADA-системы) включаются в состав более крупных систем верхнего уровня, таких как ИСУ ПП, которые объединяют технологические, обслуживающие и корпоративные уровни инфраструктуры предприятия.

ИСУ ПП позволяет решать задачи технологического управления, вовлекать в процесс управления дополнительные типы оборудования и технологий, поддерживает взаимную интеграцию и взаимодействие в рамках системы верхнего уровня центра сбора и обработки технологической информации (ЦСТИ), системы планирования ресурсов предприятия (ERP - Enterprise Resource Planning), системы управления основными фондами предприятия (EAM - Enterprise Asset Management) в составе системы технического обслуживания и ремонта (ТОИР). В рамках ИСУ ПП также отдельно выделяются корпоративная информационная система управления (КИСУ), в состав которой входят ERP и ТОИР; автоматизированная система технологического управления (АСТУ); а также, в рамках технологического уровня, АСУТП, система телемеханики и связи (СТМиС). Как правило, первоначальная модель оборудования строится в рамках КИСУ. На основании модели КИСУ выстраивается модель технологического управления АСТУ. В основании модели измерений АСУТП (СТМиС), в рамках которой функционирует SCADA-система, лежит модель АСТУ. КИСУ и АСУ ТП взаимодействуют на базе общей информационной модели. Интеграция обеспечивается благодаря общей интеграционной шине или связующей сети. Таким образом, благодаря интеграции систем корпоративного, технического и производственного уровней, достигается единое информационное пространство предприятия [6].

1.2.6 Сравнительный обзор SCADA и IT систем

Первоначально АСУ ТП имели мало сходства с IT-системами. Различие заключалось в том, что АСУ ТП являлись изолированными системами, использовавшими свои собственные протоколы управления, специализированное оборудование и программное обеспечение. Сегодня относительно дешёвые и доступные устройства, использующие для связи протокол IP, вытесняют специализированное оборудование. Данный факт ведёт к увеличению вероятности возникновения уязвимостей и инцидентов в области информационной безопасности. АСУ ТП адаптирует решения, используемые в IT-системах, в целях обеспечения связанности компонентов сети и возможности удалённого доступа. Проектирование и внедрение подобных решений в процессе адаптации происходит с использованием стандартных компьютеров, операционных систем и сетевых протоколов, что начинает придавать АСУ ТП сходство с IT-системами. Подобного рода интеграция предоставляет АСУ ТП новые возможности из мира IT-систем, но, в то же время, способствует значительно меньшей изоляции АСУ ТП от внешнего мира, создавая необходимость дополнительной защиты. В случае внедрения в АСУ ТП решений, изначально разработанных для обеспечения безопасности IT-систем, необходимо соблюдать особые меры предосторожности. В некоторых случаях возникает необходимость внедрения решений в области информационной безопасности, учитывающих окружающую обстановку АСУ ТП (т.е. состояние системы и её окружение). АСУ ТП во многом отличаются от традиционных систем IT, включая характерные риски и приоритеты. Некоторые риски включают значительную опасность для здоровья и безопасности людей, серьезный экологический и экономический ущерб (загрязнение окружающей среды, производственные потери, негативное влияние на экономику страны). АСУ ТП отличаются иными требованиями в области производительности, надёжности и отказоустойчивости, а также используют набор специализированных ОС и ПО, нетрадиционных с точки зрения технической поддержки традиционных IT-систем. Кроме того, цели безопасности и отказоустойчивости могут вступать друг с другом в конфликт в процессе проектирования и эксплуатации систем управления. Примером может служить ситуация, в которой парольная аутентификация и авторизация на АСУ ТП, необходимые с точки зрения ИБ, не должны ни в каком виде мешать или препятствовать проведению операций в чрезвычайных ситуациях. В таблице Б.1 перечислены основные различия в области обеспечения информационной безопасности офисных (IT) систем и АСУ ТП [3].

1.3 Информационная безопасность SCADA

1.3.1 Уязвимости и факторы риска современных SCADA

Уязвимости в АСУ ТП могут возникать в результате недостатков, неправильного или ненадлежащего технического обслуживания технических платформ, включая аппаратную часть, операционные системы и приложения, а также благодаря неполным, несоответствующим или попросту несуществующим руководящим документам и политикам в области ИБ. Данные уязвимости могут быть уменьшены благодаря различного рода средств контроля безопасности, таких как проведение обновлений ОС и приложений, контроль физического доступа, специализированное ПО в области информационной безопасности (например, антивирусное ПО). Корпоративная политика безопасности может уменьшить число уязвимостей определяя обязательное использование парольной защиты или, например, регламентируя параметры обслуживания или требования в отношении подключения модемов к компонентам АСУ ТП. Таблица В.1 содержит подробный список и описание основных уязвимостей АСУ ТП [5].

Факторы риска современных АСУ ТП (SCADA-систем) базируются на необходимости поддержки их конкурентоспособности на рынке ПО (расширяемость, адаптируемость и т.д.). Также на факторы риска SCADA-систем влияет опасность осуществления их аудита и реализации необходимых изменений. Таким образом, среди факторов риска современных SCADA-можно выделить следующие: внедрение стандартизованных протоколов и технологий со списком известных уязвимостей, связанность сети системы управления с другими сетями (например, сетью ИТ-системы), широкая доступность и распространение технической информации и документации о системах управления, высокие риски проведения аудита, высокие риски внесения исправлений. Особенности каждого из перечисленных факторов риска подробно описываются в таблице В.2 [5].

1.3.2 Сценарии проведения атак на SCADA

Актуальные сценарии проведения современных атак на SCADA-системы включают атаку при помощи эксплойта**,** атаку злоумышленным служащим, а также управление посредством внедрения вируса.

Сетевая архитектура современных предприятий объединяет компоненты корпоративной сети и компоненты АСУ ТП, что обуславливает уязвимость SCADA-систем перед атаками с использованием эксплойтов. В данной схеме для корпоративной, внутренней и управляющей сети используются свои собственные межсетевые экраны. Однако, зачастую, на практике межсетевые экраны заменяются простыми маршрутизаторами (сетевыми коммутаторами), таблицы маршрутизации которых могут дополнительно быть неправильно настроены с точки зрения ИБ. В некоторых небезопасных случаях все три сети (корпоративная, внутренняя и управляющая) могут быть объединены воедино без использования каких-либо сетевых инструментов. Подробное описание процесса атаки эксплойтом на SCADA-систему рассмотрено в Приложении Г [7].

Одной из проблем практически всех АСУ ТП и SCADA-систем является невысокая защищённость от злонамеренных действий конечных пользователей – управляющего и обслуживающего персонала (например, инженеров, операторов, администраторов и т.д.). Под понятием “инсайдер” подразумевается сотрудник компании, имеющий непосредственный доступ к конфиденциальным данным, системе безопасности, управляющему, сетевому или производственному оборудованию. Негативные действия инсайдера могут иметь как случайный характер, вызванный ошибкой или невнимательностью персонала, так и преднамеренный, обусловленный сознательным желанием вывести из строя АСУ ТП предприятия или создать чрезвычайное происшествие. Подробное описание процесса атаки SCADA-системы инсайдером рассмотрено в Приложении Г [8].

Современные вирусы используют уязвимости ОС, позволяющие производить повышение уровня привилегий до уровня администратора. Они используют специальные методы загрузки ПО, позволяющие не быть замеченными антивирусами, программами анализа поведения и программами для предотвращения вторжений. Вирусное ПО может самостоятельно осуществлять вредоносную деятельность или создавать скрытый канал для последующей атаки системы злоумышленником. Особенности атак SCADA-систем посредством вирусного ПО рассмотрены в Приложении Г [9].

1.3.3 Актуальные исследования в области ИБ SCADA

На сегодняшний день исследования в области ИБ SCADA-систем проводятся по следующим направлениям, а именно:

а) Исследования по внедрению в инфраструктуру SCADA-систем системы обнаружения вторжений (IDS). Примером может служить создание IDS, основанной на принципе моделирования, для сетей Modbus/TCP [10]. Подход основывается на периодичности Modbus трафика в канале HMI - PLC. В результате, каждый HMI-PLC канал может быть смоделирован при помощи собственного конечного детерминированного автомата. Кроме того, к данному направлению исследований относится метод анализа путей распространения информации (трафика) с построением списка доверенных маршрутов, основанный на свойствах сетевых пакетов (адрес отправителя/получателя, порт сервера, транспортный протокол) [11]. Результатом исследований и работы Digital Bond’s в данном направлении является Quickdraw SCADA IDS, предоставляющая обширный список сигнатур уязвимостей оборудования и протоколов АСУ ТП для Sourcefire Snort IDS [12, 13].

б) Исследования по внедрению в инфраструктуру SCADA-систем средств проверки и фильтрации сетевых пакетов по их содержимому “Deep Packet Inspection (DPI)“ (Modbus DPI firewall - Honeywell Modbus Read-only Firewall) [14].

в) Исследования по созданию и улучшению ПО для тестирования на проникновение (PunkSPIDER, Shodan, VPN Hunter, Exploit Search, Nmap-Online, Metasploit, GLEG SCADA+ Pack, OpenVAS, ovaldbru.altx-soft.ru и др.)

г) Исследования по созданию и улучшению ПО для выявления уязвимостей на стыке IT и SCADA посредством анализа трафика (Tenable Passive Vulnerability Scanner™) [15].

д) Исследования по созданию интеллектуальных SCADA-систем. Примером может служить создание агентной распределённой отказоустойчивой системы управления, способной к самостоятельному переконфигурированию в случае выхода из строя отдельного компонента, и её адаптация к SCADA [16].

Внимание специалистов ИБ было вызвано накопившимися проблемами и уязвимостями в области ИБ АСУ ТП и SCADA-систем, которые не были замечены в процессе проектирования, списаны на возможные риски или же просто проигнорированы [17]. Именно этими уязвимостями и воспользовались злоумышленники при проведении своих атак. Сетевой компьютерный вирус-червь Stuxnet – исторический пример данному факту.

1.3.4 Актуальные проблемы в области ИБ SCADA

На основании проведённого исследования предметной области (ИБ АСУ ТП) и объекта защиты (SCADA-системы), возможно выделить следующие актуальные проблемы СУ, а именно:

а) Физическая изоляция сетей АСУ ТП перестала быть эффективной мерой ИБ [9].

б) Объединение АСУ ТП и IT-инфраструктуры влечёт за собой увеличение доступности компонентов SCADA для атак извне [18, 19].

в) Наличие возможности удалённого доступа с высоким уровнем привилегий к компонентам SCADA-систем [7, 9].

г) Уязвимость от атак инсайдером за счёт отсутствия средств мониторинга и контроля деятельности сотрудников, политик безопасности, средств авторизации и аутентификации и др. [7, 9, 18, 20].

д) Негативное влияние СЗИ на информационные процессы АСУ ТП (задержки в обработке команд и запросов, прекращение информационного обмена и др.) [3].

е) Сложность или невозможность постоянного обновления ПО и/или внесения исправлений [21].

ж) Проведение технического аудита (тесты на проникновение, инструментальные проверки и др.) характеризуется высокой степенью риска [22].

з) Отсутствие в архитектуре и реализации СУ компонентов, отвечающих за безопасность и аутентификацию, или трудоёмкость (невозможность) их внедрения [18, 23].

и) Принцип “Безопасность через неясность” становится неактуальным в условиях современного рынка ПО [23].

к) Отсутствие или несовершенство нормативно-правовой базы по принципам проведения аудита безопасности компонентов АСУ ТП [24, 25].

л) Протоколы АСУ ТП и SCADA не имеют детализации на уровне соединения [26].

Таким образом, упомянутые ранее подходы в области исследований средств, методов и технологий обеспечения ИБ СУ обладают рядом недостатков. Во-первых, отсутствует возможность заблаговременного (превентивного) обнаружения атаки или внедрения вредоносного ПО. Во-вторых, отмечается наличие дополнительной вычислительной нагрузки по обеспечению функционирования средств ИБ, возлагаемой на важные технологические компоненты инфраструктуры СУ.

2 Метод имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов

На основании результатов анализа актуальных проблем, исследований, технологий и решений в области информационной безопасности АСУ ТП и SCADA-систем, изложенных в рамках данной статьи, была разработана идея метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем. В рамках данного подхода предполагается создание для ПО SCADA-систем (а также обновлений ПО SCADA-систем) некоторого подобия “цифровой подписи” – эталонной модели, отражающей “правильное” состояние (конфигурацию) и имитирующей “правильную” работу системы. Предлагаемый метод состоит из трёх шагов, представленных на рисунке 1. На первом шаге происходит построение эталонной модели. На втором шаге происходит сравнение показателей модели и реальной системы. На третьем шаге происходит выявление отклонений в работе системы.

Рисунок 1. Этапы метода моделирования эталонного состояния и поведения SCADA-систем.

Расхождения в сравниваемых характеристиках могут быть связаны с заражением оборудования системы вредоносным ПО, нарушением сетевой инфраструктуры, а также всевозможными ошибками и неполадками оборудования. Таким образом, если значение функции отклонения превысит безопасный порог, в модели будет зафиксирован подозрительный инцидент, а управляющий персонал на станции HMI будет об этом проинформирован.

Разрабатываемый метод лишён недостатков, упомянутых в пункте [1.3.4], и обладает следующим функционалом:

а) Предоставляет возможность раннего обнаружения компрометации инфраструктуры SCADA-систем вредоносным ПО.
б) Позволяет обнаруживать действия хакера.
в) Позволяет обнаруживать скрытые каналы [27, 28].
г) Упрощает процедуру расследования инцидентов.
д) Определяет нарушения процедуры управления конфигурациями.

Возрастающая структурная и технологическая сложность SCADA-систем, растущий уровень их интеграции с другими системами предприятия, а также увеличивающаяся сложность, незаметность, направленность и интеллектуальность вредоносного ПО пагубно отражаются на безопасности SCADA-систем современных АСУ ТП. Отсутствие комплексных программных решений в области обеспечения ИБ SCADA-систем или невозможность применения тех или иных решений (например, в силу отсутствия дополнительных вычислительных ресурсов оборудования) лишь усугубляет упомянутую ранее проблему. В подобных условиях рассматриваемый метод имитационного моделирования является крайне актуальным в области обеспечения безопасности современных SCADA-систем.

Кроме того, данный подход обладает рядом дополнительных достоинств. Во-первых, он не требует значительных дополнительных затрат, ввиду отсутствия необходимости построения тестовой платформы SCADA-системы. Во-вторых, метод, благодаря своей модульной архитектуре, не накладывает строгих ограничений на возраст, версию или тип моделируемого оборудования, а также предоставляет возможности для эффективной синхронизации изменений в системе управления и модели. Процесс синхронизации, в зависимости от структуры изменений, может заключаться в изменении пороговых значений для функций в модели, внедрении новых модулей (в случае использования нового типа оборудования), а также написании программных адаптеров для имеющихся модулей (в случае изменения текущего интерфейса программирования приложений SCADA-системы).

Стоит отметить, что данный подход предполагает правильную настройку (в соответствии с формальными требованиями производителя оборудования) системы управления квалифицированным инженером с целью исключения (при выполнении анализа внедрения вредоносного ПО) ошибок первого рода, связанных с неправильной настройкой работы системы.

Вышеупомянутый метод лежит в основе разрабатываемой в рамках данной работы системы обнаружения вторжений для SCADA-систем.

2.1 Имитационное моделирование как методика разработки

В качестве метода исследования было выбрано ИМ [29, 30]. Выбор в пользу ИМ оправдан следующими факторами, а именно:

1) Невозможность и необоснованность проведения экспериментов над реальными системами. Использование работающей и исправно функционирующей АСУ ТП, контролирующей важные производственные процессы, в качестве тестовой площадки для проведения экспериментов чрезвычайно рискованно и опасно.

2) Ресурсоёмкость макетирования. Создание и настройка тестового стенда, представляющего копию исходной системы управления, является крайне ресурсоёмкой и дорогостоящей задачей.

3) Ресурсоёмкость построения модели информационной системы в виртуальной среде. Задача виртуализации масштабной, многокомпонентной системы управления, включающей распределённые компоненты, может быть сопоставимой по сумме затрат с задачей анализа её защищённости.

4) ИМ предоставляет возможность имитации поведения системы во времени, что предоставляет дополнительные возможности для оценки правильности работы АСУ ТП [29].

5) ИМ позволяет производить построение моделей систем управления любой структурной и архитектурной сложности [29, 30].

Процедура имитации заключается в воспроизведении информационного обмена и параметрах оборудования SCADA системы посредством создания информации о сетевом взаимодействии (“дампов трафика”) и информации об оборудовании. Имитация необходима для предварительной проверки реакции системы обнаружения вторжений на подозрительную активность, а также может быть использована для обучения элементов системы обнаружения вторжений. Таким образом, система обеспечивает имитацию атаки хакера и реакции системы обнаружения вторжений. Время в системе определяется временными метками пакетов, передаваемых в рамках сетевого взаимодействия.

2.1.1 Особенности имитационного моделирования в контексте SCADA

ИМ как методике исследования и анализа функционирования СУ, а также её отдельных структурных составляющих, применяемой в контексте создания СОВ или СЗИ, свойственен ряд отличительных особенностей, заслуживающих отдельного внимания.

Первая из особенностей связана с одним из свойств СУ, определяемой на базовом уровне как “совокупность компонентов, находящихся в определённых отношениях друг с другом и со средой” [29]. Речь идёт о синергетичности информационной системы. Выражаясь точнее, о синергетическом эффекте, получаемом при объединении отдельных компонентов СУ и заключающемся в появлении у системы новых свойств и характеристик, возникающих как результат многовариантного и неоднозначного поведения многофакторных сред или многоэлементных структур [30-32]. Примером подобному явлению в контексте ИБ может служить возможность появления новых недекларированных путей маршрутизации при внедрении в сетевую инфраструктуру новых сетевых протоколов. Подобные синергетические эффекты негативно сказываются на ИБ СУ. ИМ, в свою очередь, способно предоставить возможность заранее обнаруживать причины и возможные последствия возникновения негативных синергетических эффектов и принимать соответствующие меры в отношении СЗИ.

В качестве основы имитационной модели предполагается использовать модель акторов из категории агентного моделирования [31, 33]. Выбор в пользу МА обусловлен следующими фактами, а именно:

1) Современные системы управления (SCADA-системы) являются гетерогенными. Они состоят из компонентов (HMI, PLC, сетевое оборудование, датчики, сенсоры и др.), отличающихся логикой функционирования имитационной модели и обладающих разным набором исследуемых характеристик. МА позволяет эффективно моделировать многокомпонентные гетерогенные системы [32].
2) МА эффективно отражает суть и процедуры информационного обмена в современных компьютерных сетях, а также поддерживает параллелизм, распределённость, масштабируемость и асинхронность на уровне своей архитектуры.
3) Модели, основанные на дискретно-событийном подходе, лишены эффективных средств синхронизации происходящих событий. Модели на основе системной динамики абстрагируются от отдельных объектов и событий в системе. Следовательно, данные модели не позволяют осуществлять детальный анализ сетевой инфраструктуры.

2.2 Модель акторов

МА была впервые предложена в 1973 году Карлом Хьюиттом, Питером Бишопом и Ричардом Штайгером [33] и в дальнейшем исследовалась Гюль Агой [31].

В МА все[1] объекты представляются в виде особой сущности - актора. Подобный подход схож с методологией ООП, в которой структурной единицей является объект. Основное различие между подходами состоит в том, что в ООП исполнение программы происходит последовательно, тогда как в модели акторов оно, как правило, происходит асинхронно, позволяя производить различные вычисления в одно и то же время. Данная возможность обеспечивается разделением отправителя и посланных сообщений – фундаментальной особенностью МА [34].

Актор – это структурная вычислительная единица, которая может одновременно в качестве ответа на полученное сообщение совершить следующие действия:

а) Отправить конечное число сообщений другим акторам.

б) Создать конечное число новых акторов.

в) Определить стратегию обработки следующего сообщения в свой адрес.

Перечисленные действия могут осуществляться параллельно и в произвольном порядке. Идентификация акторов для обеспечения обмена сообщениями происходит по специальному индивидуальному ключу (например, сетевой адрес или адрес в оперативной памяти), называемому “почтовым адресом”. Актор может осуществлять коммуникацию только с теми акторами, адреса которых он имеет. Среди них могут быть адреса акторов, от которых он получал сообщения, которые имел в списке по-умолчанию, а также тех, что он создал. Концепция МА не гарантирует правильную последовательность доставки сообщений, а также не гарантирует срок доставки сообщения адресату. Гарантируется лишь сам факт доставки сообщения [35].

Модель акторов, как и большинство математических теорий, строится на собственной системе аксиом, определяющих базовые понятия и ограничения. Введение базовых аксиоматических понятий необходимо для борьбы с возможными противоречиями или бесконечным регрессом (англ. “infinite regress”) [36]. Примером, иллюстрирующим, бесконечный регресс может быть попытка представить актор как сущность, в состав которой обязательно входит отдельный структурный элемент – очередь входных сообщений (т.н. “почтовый ящик”). Тогда, рассуждая в терминах модели акторов, где все сущности являются акторами, “почтовый ящик” тоже является актором, которому для

[1] В общем смысле, исключением могут являться лишь некоторые базовые сущности, определяемые конкретной аксиоматикой модели акторов.

работы необходим “почтовый ящик” и т.д. [33]. Принимая во внимание данный факт, в качестве базовых понятий выступают события, сообщения, а также способность акторов к их приёму и отправке. Ограничения модели касаются конечности множеств событий и акторов. В любой заданный момент времени одному актору может быть отправлено конечное множество сообщений, тогда как одно сообщение может быть направлено конечному множеству акторов [37]. Введение ограничений необходимо для обоснования того факта, что МА применима на практике. Взаимодействие акторов (пересылка сообщений) осуществляется системой вычислений. Она может быть реализована с использованием правил частичного порядка (подобно денотационной семантике), определяющих причинно-следственную связь между событиями, а также некоторого управляющего компонента (базовой сущности), также определяющего некоторую операционную семантику [37-40].

Рассмотрим основные концепции разработки ПО на основе МА. В отличие от передачи контекста исполнения программы связанным объектам в ООП, в МА акторам передаётся некоторое сообщение. Подобный подход не приводит к созданию глубокого стека вызовов, когда множество объектов взаимодействуют друг с другом посредством вызова программных функций. Обычно каждый актор имеет свой контекст выполнения. Данная возможность достигается за счёт использования особых примитивов многозадачности – волокон (англ. “fibers”). Данный примитив позволяет создавать сотни и тысячи акторов, а следовательно и контекстов выполнения, на один процесс. Кроме того, наличие у каждого актора отдельного контекста выполнения и возможность асинхронного взаимодействия для получения данных позволяет в значительной мере сократить число критических секций, а как следствие, сократить время простоя ядер процессора.

2.2.2 Математические обоснование модели акторов в контексте SCADA

В данной главе будет рассмотрено формальное представление модели акторов, специфицированное для моделирования эталонного состояния и поведения SCADA-систем. Данная спецификация учитывает особенности обеспечения информационной безопасности на АСУ ТП (SCADA-системах), проявляющиеся в необходимости предоставления актуальной информации об уровне безопасности в заданный момент времени. Подобная необходимость накладывает временные ограничения на процесс вычислений и процесс информационного обмена. Следовательно, для осуществления обработки, анализа и предоставления актуальной информации используемая в контексте SCADA-систем модель акторов должна обладать возможностью оценки времени доставки сообщений (по крайней мере, текущего) и, как следствие, доставки сообщений в хронологическом порядке. Однако базовая концепция МА не гарантирует порядок и срок доставки сообщений адресатам, что обуславливает необходимость спецификации МА.

В качестве основы формальной модели акторов была выбрана модель, описанная в статье Jorn W. Janneck [41]. В качестве основы для системы вычислений была выбрана операционная семантика (трансляционная), описанная в статье Lukito Muliadi [39]. Спецификация выбранной формальной модели, реализующей базовую концепцию МА, состоит из двух пунктов. Первый пункт спецификации модели заключается в ведении в определение актора новой базовой сущности – набора проверок (“ассертов” от англ. assert), которые могут определять допустимые временные границы вычислений и срока пересылки сообщений, а также обеспечивать дополнительные возможности по оценке правильности работы актора. Вторая часть спецификации включает введение в управляющий пересылкой сообщений компонент системы вычислений процедуры запуска набора проверок актора, выполняемого после исполнения его основных функций. Следует отметить, что вносимые изменения не затрагивают внутренней логики механизмов функционирования МА, следовательно не требуют повторного доказательства каких-либо положений МА.

Рассмотрим первую часть спецификации МА более подробно.

Определение 1 (Актор, переход). Пусть U – множество всех возможных значений (универсум), а $S = U^*$ - множество всех конечных последовательностей в U. Для любого непустого множества состояний Σ *актором* $m \to n$ *с отправкой* (сокр. актор) называется набор $< \sigma_0, \Sigma, \tau, \mathrm{A} >$, где $\sigma_0 \in \Sigma$ – начальное состояние, $\tau \subseteq \Sigma \times S^m \times S^n \times \Sigma$ – отношение перехода, $\mathrm{A} = \{\alpha_1 \dots \alpha_n\} \mid a_i = p_i(s, s')$, $p(s, s'): S^m \times S^n \longrightarrow \{0,1\}$, $s \in S, s' \in S, n \in N$. Элемент из τ называется *переходом*. Элемент из А называется *проверкой (ассертом)*. Для

любого перехода $(\sigma, s, s', \sigma') \in \tau$ верна запись $\sigma \xrightarrow[t]{s \to s'} \sigma'$, в которой состояние $\sigma(\sigma')$ называется *предшественником (преемником)* состояния $\sigma'(\sigma)$, а $s(s')$ – *входым (выходным) сообщением*. Тогда множество *акторов* $m \to n$ определяется как $\mathcal{A}^{m \to n}$, а множество всех *акторов* как $\mathcal{A} = \bigcup_{m,n \in N} \mathcal{A}^{m \to n}$.

Введение множества проверок A в определение актора является первой частью спецификации MA, обеспечивающей возможность проверки актором временных и алгоритмических условий и ограничений. Для обеспечения необходимого функционала анализа времени входное и выходное сообщение $s(s')$ должно содержать в себе информацию о времени своей отправки и получения, то есть $\tilde{s} = s \circ t$, где s – оригинальное сообщение, $t \in S$ – тэг (временная метка), $\circ$ - отношение композиции.

Далее рассмотрим вторую часть спецификации MA, имеющую отношение к системе вычислений и определяющую процедуру осуществления проверки множества ассертов некоторого актора. Предварительно вводится ряд определений.

Определение 2 (События, сигналы). *Событием* называется элемент множества $E = T \times V$, где T – множество всех меток, V – множество всех значений. *Сигналом* $s \in S$ называется набор событий, где S – множество всех подмножеств E. Кортеж из N сигналов обозначается $s = [s_1, \dots, s_N] \in S^N$. Пустым сигналом λ называется сигнал без событий. Кортеж из N пустых сигналов обозначается Λ^N.

Определение 3 (Индивидуальные метки). Набор $T(s) \subseteq T$ называется набором индивидуальных меток в сигнале s.

Определение 4 (Функция передачи). Функция передачи f принимает набор входных событий и индивидуальную метку (время) и возвращает набор выходных событий и новую функцию передачи, называемую *продолжением*, причём. Множество всех функций передачи $\Gamma = \{ f : S^m \times T \longrightarrow S^n \times \Gamma\}$ для m входов и n выходов, причём $f(\Lambda^M, t) = f(\Lambda^N, f)$ для всех $t \in T$.

Определение 4 (Функция проверки). Функция проверки $asrt$ принимает наборы событий и индивидуальных меток (времён) и возвращает набор выходных событий $asrt : S^m \times T \times S^n \times T \longrightarrow S^k$.

Для взаимодействия акторов определяется операционная семантика. Пусть существует N сигналов, M акторов с функциями передачи $f_1, \dots, f_n$, непустые множества входных $I_1,\dots,I_M$ и выходных $O_1,\dots,O_M$ индексов, $s \in S^N$ – набор присутствующих событий, тогда операционная семантика задаётся следующим образом:

1. while ($s \neq \Lambda^N$) {

2. let $\tau = \min(T(s))$
3. while $(s(\tau) \neq \Lambda^N)$ {
    a. let $j = \min\{k \in \{1, 2, ..., N\} : s_k(\tau) \neq \lambda\}$
    b. let $i \in \{1, 2, ..., M\} : j \in Ii$
    c. let $(s'', f_i) = f_i(\pi_{Ii}(s(\tau)), \tau)$
    d. let $(s''') = as_i(\pi_{Ii}(s(\tau)), s'', \tau, \tau')$
    e. let $s = s - Select_i(s(\tau)) \cup Expand_i(s'') \cup Expand_i(s''')$
4. }}

Сигнал s, представленный в определении цикла на строке 1, выступает в роли глобальной очереди сообщений. Главный цикл продолжается до тех пор, пока не будут обработаны все события. Переменная τ в строке 2 отражает текущее время. На строке 3.a переменная j определяет индекс сигнала с наименьшим рангом, содержащего входные события в момент τ. На строке 3.b переменная i определяет индекс процесса, принимающего сигнал S_j как один из входных сигналов.

На строке 3.c выражение $\pi_{Ii}(s(\tau))$ определяет набор событий во время τ, которые предназначены актору с индексом i. Актор с индексом i выполняет свою работу во время τ. Выходные события хранятся в переменной s''. Строка 3.в определяет вторую часть спецификации МА, осуществляющую выполнение и обработку множества проверок некоторого актора с использованием функций проверок, генерирующих события-предупреждения в случае обнаружения отклонений в работе. Функция Функция $Select_i$ на строке 3.e определяет события, обработанные актором с индексом i. Функция $Expand_i$ определяет события, порождённые актором с индексом i. Глобальная очередь в виде сигнала s обновляется. В конце главного цикла глобальная очередь оказывается пустой.

2.2 Основные концепции функционирования эталонной модели

Функционирование эталонной модели основано на двух критериях. Первый критерий отражает “правильное” состояние (конфигурацию) системы. Второй критерий имитирует “правильную” работу системы.

Возможность выявления эталонных характеристик состояния и поведения обусловлена стабильностью настроенной (в рамках некоторого технологического процесса) SCADA-системы и высокой периодичностью трафика внутри её инфраструктуры, что подтверждается возможностью моделирования каналов передачи данных (HMI-PLC) при помощи собственных конечных детерминированных автоматов [10]. Таким образом, при наличии информации о правильном функционировании периодической системы в прошлом становится возможным предсказание информации о её правильном функционировании в будущем. Различные временные интервалы позволяют сравнивать информацию в рамках работы конкретного оператора (час), смены (день), рабочего графика (месяц), сезонных изменений (год) и т.д.

Критерий эталонного поведения системы определяется множеством каналов взаимодействия PLC и HMI и задаётся с помощью конечных автоматов данных каналов. Критерий эталонного состояния системы определяется двумя характеристиками: состоянием сети и настройками оборудования (PLC).

Для каждого критерия определяется частная функция безопасности, оценивающая уровень его компрометации. На следующем этапе определяется общая функция безопасности, объединяющая информацию, получаемую от частных функций безопасности.

Для построения модели будут использоваться сетевые, статистические и диагностические данные, получаемые с использованием протоколов Netflow, а также методов “зеркалирования” портов (англ. “port mirroring”). Трудоёмкость построения эталонной модели не высока благодаря использованию математического аппарата нечёткой логики и зависит только от числа элементов сетевой инфраструктуры. Настройка модели экспертом будет заключаться в определении пороговых значений для функций, отражающих степень отклонения значений реальных параметров от предполагаемых. Таким образом, если значение функции превысит безопасный порог, в модели будет зафиксирован подозрительный инцидент (внедрение вирусного ПО), а управляющий персонал на станции HMI будет об этом проинформирован.

2.2.1 Критерий эталонного состояния

Критерий эталонного (“правильного”) состояния (конфигурации) системы определяется набором важных системных настроек, параметров и свойств, характерных для исправно функционирующей и нескомпрометированной СУ. Структурными элементами критерия являются: состояние сети и состояние настроек оборудования (ПЛК).

Для построения характеристики состояния сети на первом этапе из данных сетевой статистики формируются матрицы сетевой активности, отражающие картину информационного обмена на различных уровнях сетевой модели OSI за определённый промежуток времени. Матрицы могут содержать информацию о факте взаимодействия некоторых компонентов (матрицы сетевой смежности), статистическую информацию о среднем объёме передаваемого трафика, времени и периоде взаимодействия между заданными узлами. Формально данные матрицы задаются следующим образом, а именно: $\mathcal{M}\left(OSI_{level}, char, t_{comp}, \delta\mathrm{T}\right) : N \times N \rightarrow \mathcal{U}$, где OSI_{level} – выбранный уровень модели открытых систем, $char \in Chars = \left\{char_1, \dots, char_p\right\}, p \in \mathbb{N}$ – выбранная характеристика (факт взаимодействия сетевых узлов, средний объём трафика и т.д.), t_{comp} – момент времени, с которого начинается построение матрицы, δT – период времени, начиная от t_{comp}, $N = \{n_1, \dots, n_k\}, k \in \mathbb{N}$ – конечное множество элементов сетевого взаимодействия на уровне OSI_{level} (например MAC-адреса для канального уровня или пара <IP-адрес; порт> для транспортного уровня), $\mathcal{U}$ – множество значений матрицы сетевой активности (может быть множеством {0,1} для матриц смежности или $\mathbb{R}$ для среднего объёма трафика). Столбец данной матрицы соответствует узлам источникам, строка – узлам назначениям. Матрица является квадратной. Для выявления отклонений в работе происходит сравнения двух матриц сетевой активности, одна из которых является эталонной. При это происходит построение матрицы разности следующего вида: $\mathcal{D}(\mathcal{M}_1, \mathcal{M}_2) = \mathcal{M}_1 - \mathcal{M}_2$, где $d_{ij} = |\mu_{ij} - \mu'_{ij}| \in \mathcal{D}$, где $\mu_{ij} \in \mathcal{M}_1, \mu'_{ij} \in \mathcal{M}_2, i, j \in \mathbb{N}$. Если сетевые матрицы имеют различную размерность (данный факт, как правило, означает появление нового узла или соединения), меньшая матрица расширяется до большей размерности, а в качестве значений выступают значения по-умолчанию (например, 0 для матриц смежности). Столбцы и строки матриц должны быть одинаково отсортированы.

Далее определятся множество функций-критериев состояния сети $NetCrits = \{netCrit_1, \dots, netCrit_i\}, i \in \mathbb{N}$. Критерий $netCrit_i\left(t_{comp}, \delta\mathrm{T}, t\right) = f_i(\overline{\mathcal{M}})$, где t_{comp}- начало времени эталонного измерения, δT - интервал эталонного измерения, t – начало реального

измерения, $\overline{\mathcal{D}}$ – набор матриц разности, $\overline{\mathcal{M}}$ - набор матриц сетевой активности, $Dom\ netCrit_i = [0,1]$. Функция f_i вычисляет сетевой критерий компрометации, исходя из информации, содержащейся в матрицах сетевой активности (требует для вычисления критерия хотя бы одну матрицу) с построением матриц разности $\mathcal{D}$. Конкретные критерии из множества NetCrits могут быть следующими:

а) Разница в матрицах смежности на канальном, сетевом, транспортном уровне и уровне Modbus с учётом опасности возникших отклонений. Иначе говоря $f = \frac{\sum_{ij}(w_{ij} * d_{ij})}{\sum_{ij} d_{ij}}$, где $d_{ij} \in \{0,1\}$ – элемент матрицы разности, соответствующей выбранному сетевому уровню, а $w_{ij} \in [0,1]$ – опасность возникновения данного взаимодействия ($\sum_{ij} w_{ij} = 1$). Опасность может определяться, исходя из классификации сетевых элементов по степени доверия и направления взаимодействия (например, из недоверенного узла в доверенный и наоборот).

б) Соответствие Query-Response на уровне Modbus.

в) Соответствие по объёму, времени, допустимости пути (например, до PLC) трафика.

Объединением показаний различных функций-критериев состояния сети задаётся функция всплесков в сети: $\boldsymbol{\varphi_1} = \max_{\text{NetCrits}} netCrit\_i\,, Dom\ \varphi_1 = [0,1]$. Для построения характеристики состояния настроек оборудования на первом этапе формируются таблицы диагностических данных контроллеров $\bar{\mathcal{T}} = \{\mathcal{T}_1 \dots \mathcal{T}_k\}, k \in \mathbb{N}$. Далее из табличных данных аналогично формируется множество критериев $PlcCrits = \{plcCrit_1, \dots, plcCrit_j\}, j \in \mathbb{N}$, где $plcCrit_j(t_{comp}, \delta\mathrm{T}, t) = g_j(\bar{\mathcal{T}})$. Сравниваемые элементы являются векторами, содержащими информацию о параметрах контроллеров за некоторый промежуток времени. Конкретные функции-критерии компрометации настроек оборудования касаются правильности работы логики, конфигурации, оперативной памяти и процессора контроллеров. Примером может служить функция оценки числа запущенных процессов на контроллере $g = \frac{\sum_i (w_i * d_i)}{\sum_i d_i}$, где $w \in [0,1]$ – важность процесса (ОС, производственный, другой), а $d \in \{0,1\}$ – разница в векторе сравнения запущенных процессов. Список возможных диагностических данных для анализа определяется возможностями оборудования и протоколов (например, Diagnostic Subfunctions для Modbus) [42]. Объединением показания различных функций-критериев настроек контроллеров задаётся функция компрометации настроек контроллеров: $\boldsymbol{\varphi_2} = \max_{Plc\text{Crits}} plcCrit\_i\,, Dom\ \varphi_2 = [0,1]$.

2.2.2 Критерий эталонного поведения

Критерий эталонного (т.е.“правильного”) поведения (работы) системы определяется ожидаемыми (прогнозируемыми), согласно текущей доверенной архитектуре и реализации, действиями и событиями, их последовательностью и взаимодействием.

Поведение представлено набором конечных автоматов каналов взаимодействия HMI-PLC: $\boldsymbol{FSM - HMI - PLC} = \{f_1 \ \dots, f_n\}$. Работоспособность канала описывает функция, представленная отношением числа ошибок в канале к общему числу сообщений с учётом стратегической важности канала: $\boldsymbol{FSMWorkState}(t, \delta T, f, w) = \frac{ErrMessCount(t,\Delta T,f)}{AllMessCount(t,\Delta T,f)} * w$, где t - начало времени измерения, δT - интервал измерения, f - канал HMI-PLC, w - стратегическая важность канала ($\sum_i w_i = 1$), Dom **FSM_WorkState** = [0,1]. Интервал измерения не должен быть короче времени одного цикла технологического процесса.

Объединением показаний различных функций-критериев работоспособности каналов HMI-PLC задаётся функция для оценки поведения:

$$\boldsymbol{\varphi_3} = \max_{FSM-HMI-PLC} FSMWorkState(t, \delta T, f, w), Dom\ \varphi_1 = [0,1].$$

Возможность использования конечных автоматов при моделировании каналов PLC-HMI описывает статья N. Goldenberg, A. Wool. “Accurate Modeling of Modbus/TCP for Intrusion Detection in SCADA Systems” [10]. Так как протокол Modbus является лишь протоколом, обеспечивающим транспорт в сети, и прямо не влияет на логику работы каналов HMI-PLC, справедливо утверждение, что сама возможность построения конечного автомата стабильных и периодичных систем не зависит от используемого протокола.

В качестве примера рассмотрим автоматы, представленные в оригинальной статье [10]. Автоматы определяются как $(Q, \Sigma, \delta, q_0, F)$, где Q – конечное множество состояний, Σ – алфавит, $\delta : Q \times \Sigma \rightarrow Q$ - функция перехода, $q_0 \in Q$ – начальное состояние, , $F \subseteq Q$ – набор состояний “принятия” со следующими замечаниями:

а) Множество состояний принятия не требуется. С каждым переходом связывается некоторое Действие (аналог автомата Мура). Отклонения от шаблона функционирования автомата вызывают переход с действием “ошибка”.

б) В качестве алфавита Σ выбирается конкатенация нескольких полей Modbus (1 бит – запрос/ответ; 8 бит- код функции; 16 бит – номер ссылки; 8 бит – число слов). В качестве начального состояния выступает состояние, соответствующее получению первого запроса (query).

Функция перехода сопоставляет паре (Начальное состояние, входной символ) пару (Конечное состояние, Действие). Определяется два множества состояний Q (достигаются после получения сообщения Query) и R (достигаются после получения сообщения Response). Далее определяется текущая позиция S_i и полученный входной символ s_j, а также 4 функции перехода: нормальный переход (известный символ ведёт к следующему состоянию в периодической цепочке $s_j = s_{i+1}$), ретрансляция (появление такого же символа, как и в прошлый раз $s_j = s_i$, состояние не меняется), промах (известный символ s_j появляется в несвойственной ему позиции S_i, то есть $s_j \neq s_{i+1}$, автомат переходит в следующее состояние, соответствующее s_j), неизвестный переход (появление неизвестного символа сбрасывает автомат в начальное состояние). Оценка работоспособности системы определяется как отношение нормальных переходов к их общему числу. Построение автомата происходит на основе анализа сетевого трафика (статистики).

2.2.3 Функция безопасности и оценка компрометации SCADA

Отражением частных функций безопасности, характеризующих степень компрометации отдельных критериев состояния и поведения системы, является общая функция компрометации системы: $F(t) = \max_i(\varphi_i(t))$, где φ_i - частные функции безопасности, t - начало времени сравнения модели и системы.

Показания функции безопасности $F(t)$ интерпретируются с использованием нечёткой логики, позволяющей количественно определить уровень компрометации системы, следующим образом. Пусть $X = \mathrm{Dom}\, F(t) \in [0,1]$ – уровень компрометации системы, универсум U = [0,1], T = {“низкая”, “средняя”, “высокая”} – множество лингвистических переменных, функции принадлежности (рис.2) для $x \in X$:

а) для низкого уровня $\mu_{low}(x,a,b) = \begin{cases} 1, x \le a \\ \frac{b-x}{b-a} \quad a \le x \le b, \\ 0, \ x \ge b. \end{cases}$

б) для среднего уровня $\mu_{mid}(x,a,b,c,d) = \begin{cases} 0 \qquad x \le a \\ \frac{x-a}{b-a}, \ a \le x \le b \\ 1, \quad b \le x \le c \\ \frac{d-x}{d-c}, c \le x \le d \\ 0, \qquad d \le x \end{cases}$,

в) для высокого уровня $\mu_{high}(x) = \begin{cases} 0, x \le c \\ \frac{x-c}{d-c} \\ 1, \ x \ge b. \end{cases}$

Рисунок 2. Функции принадлежности для интерпретации уровня компрометации системы.

2.3 Архитектура программной системы

Программная система на основе метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем имеет в основе модель, представленную на рисунке 3, реализованную в терминах модели акторов. Инфраструктура SCADA-системы представлена акторами PLC и HMI, обменивающихся сообщениями посредством актора Router. Актор Router транслирует пакеты, которыми обмениваются PLC и HMI актору SystemAnalizer, который реализует логику системы обнаружения вторжений. Каждый актор содержит некоторую внутреннюю логику и обладает внешним интерфейсом для взаимодействия с другими акторами. Система может работать в двух режимах, а именно: режиме использования реальных данных и режиме имитации данных. В режиме использования реальных данных информация поступает в систему непосредственно из сетевой инфраструктуры SCADA. В режиме имитации данных информация о сетевом взаимодействии создаётся акторами PLC и HMI. Один HMI может взаимодействовать с несколькими PLC.

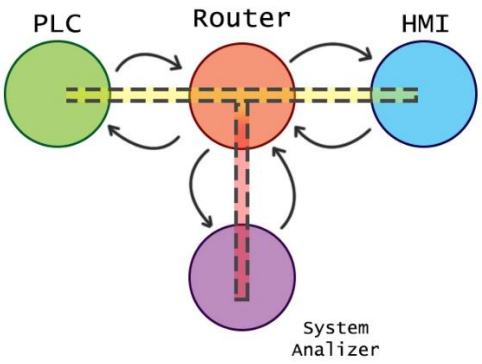

Рисунок 3. Архитектурная реализация программной системы в терминах модели акторов.

Актор PLC представляет из себя образ программируемого логического контроллера, работающего в режиме клиента и реагирующего на команды актора-сервера (например, HMI). Работает в режиме “запрос-ответ”. Данный актор может отправлять ту или иную информацию (поддерживаемую реальным оборудованием) о состоянии ПЛК или состоянии технологического процесса в ответ на запрос со стороны авторизованного актора-сервера, а также выполнять команды сервера, посылая в ответ подтверждение о результате выполнения команды. Дополнительно может следить за временем допустимой задержки между получением запроса и отправкой ответа. В режиме имитации генерирует имитационные данные, а в режиме использования реальных данных выступает в качестве

виртуального актора ПЛК, играющего роль прокси для работы с реальным оборудованием при построении таблиц диагностических данных контроллеров [2.2.1].

Актор HMI представляет из себя образ человеко-машинного интерфейса, работающего в режиме сервера и посылающего управляющие команды акторам-клиентам (PLC). Работает в режиме “запрос-ответ”. Данный актор может запрашивать ту или иную информацию (поддерживаемую реальным оборудованием) о состоянии ПЛК или состоянии технологического процесса, а также получать и обрабатывать ответы на запросы со стороны авторизованных акторов-клиентов. Дополнительно может следить за временем допустимой задержки между получением запроса и отправкой ответа. В режиме имитации генерирует имитационные данные. В режиме использования реальных данных может выступать в качестве виртуального актора человеко-машинного интерфейса. Данный актор также может функционировать как сервис на реальном HMI в качестве модуля анализа нежелательных действий оператора в составе системы обнаружения вторжений, передавая необходимую информацию основному модулю.

Актор Router представляет из себя образ сетевого элемента, который транслирует сетевые пакеты между PLC и HMI, а также посылает информацию о их взаимодействии актору SystemAnalizer (данные и статистику). В режиме использования реальных данных поведение данного актора определяется реальным сетевым оборудованием (маршрутизатором, коммутатором и т.д.), его таблицами маршрутизации. В режиме имитации генерируются и используются имитационные таблицы маршрутизации. Актор также может осуществлять проверку необходимых граничных и алгоритмических условий.

Актор SystemAnalizer представляет из себя комплексную СОВ, которая включает три модуля. Первый модуль отвечает за анализ компрометации сети. Второй модуль отвечает за анализ компрометации настроек и рабочих характеристик оборудования и ОС. Третий модуль отвечает за анализ компрометации каналов взаимодействия HMI-PLC. В основе работы модулей лежит логика вычисления соответствующих критериев эталонного состояния и поведения СУ [2.2]. Модули являются композицией служебных акторов, реализующих соответственно внутреннюю логику оценки компрометации сети, компрометации оборудования, компрометации каналов взаимодействия HMI-PLC [2.2]. Особенности реализации модулей будут рассмотрены далее в работе.

2.4 Реализация программной системы

Как уже было упомянуто ранее в пункте [2.2], основное положение модели акторов говорит о том, что всё вокруг является акторами. Так как модель акторов была ранее использована для описания инфраструктуры SCADA-системы и архитектуры системы обнаружения вторжений, действующей в её рамках, программная реализация системы обнаружения вторжений также использует в своей основе модель акторов.

В качестве языка программирования был выбран C#. Выбор данного языка определяется оборудованием, технологиями и ОС, используемыми при создании тестового стенда для апробации разрабатываемой системы. Среди них можно выделить следующие: MS Windows XP Embedded, MS .NET Framework 4.0, MS Windows Embedded Compact 7, MS .NET Compact Framework 3.5. В качестве программной основы (фреймворка) для реализации модели акторов была выбрана библиотека ReactiveExtensions (RxExtensions). Она предназначена для объединения асинхронного и событийно-ориентированного подходов к построению программ. Библиотека позволяет создавать асинхронные потоки данных, работать с ними, используя запросы LINQ и специальные планировщики. В основе реализации акторов, описываемых в рамках данной работы, лежит интерфейс ISubject библиотеки RxExtensions. ISubject задаёт начальный функционал актора, позволяя ему принимать и отправлять сообщения асинхронно. Данный интерфейс наследуется от двух других интерфейсов – IObserver и IObservable. Интерфейс IObserver определяет логику асинхронного приёма сообщений, а интерфейс IObservable определяет логику асинхронной передачи сообщений. В качестве модели вычислений (операционной семантики), управляющей передачей сообщений между акторами, выступают стандартные планировщики RxExtensions, реализующие интерфейс IScheduler. Он также задействован в реализации проверки граничных условий в рамках технологии ассертов [2.2.2].

Акторы PLC, Router и HMI в режиме работы с реальными данными используют готовую сетевую информацию, получаемую из реальной сети. Таким образом акторы отражают работу реальных устройств. В режиме имитации данные акторы сами генерируют сетевой трафик и другую необходимую информацию.

Актор PLC является составным актором, который играет роль валидатора граничных (например, временных или алгоритмических) условий и прокси (программного сервиса, позволяющего выполнять косвенные запросы к другим сервисам) для работы с реальным ПЛК (рис.4). За генерацию трафика ответственен внутренний актор PlcNetGenerator. За генерацию состояния оборудования ответственен актор PlcConfGenerator. Внутренний

актор PlcWrapper представляет из себя программную обёртку над программным интерфейсом используемого ПЛК.

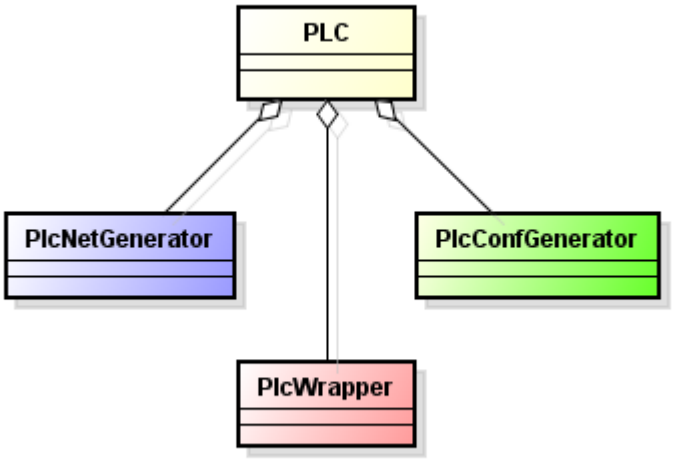

Рисунок 4. Структура программной реализации актора PLC.

Актор HMI является составным актором, который играет роль валидатора граничных условий и прокси для работы с реальным HMI, а также содержит функционал сервиса для передачи информации в систему обнаружения вторжений для оценки нежелательных действий оператора (рис.5). За генерацию трафика ответственен внутренний актор HmiNetGenerator. За работу сервиса по слежению за действиями оператора ответственен актор HmiOperatorListener. Внутренний актор HmiWrapper представляет из себя программную обёртку над программным интерфейсом используемого HMI.

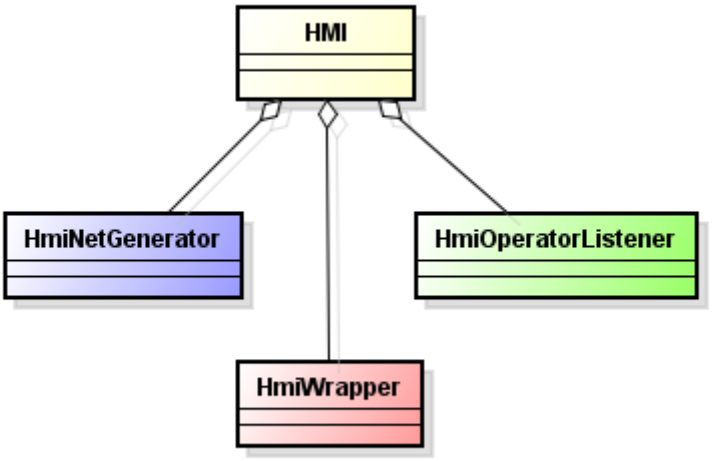

Рисунок 5. Структура программной реализации актора HMI.

Актор Router является составным актором, который выполняет роль валидатора граничных условий и прокси для работы с реальным сетевым оборудованием,

маршрутизирующим сетевые пакеты (рис.6). В режиме имитации генерирует таблицы маршрутизации и самостоятельно маршрутизирует трафик. За генерацию таблиц маршрутизации отвечает внутренний актор RoutingTableGenerator. Внутренний актор RouterWrapper представляет из себя программную обёртку над используемым сетевым оборудованием.

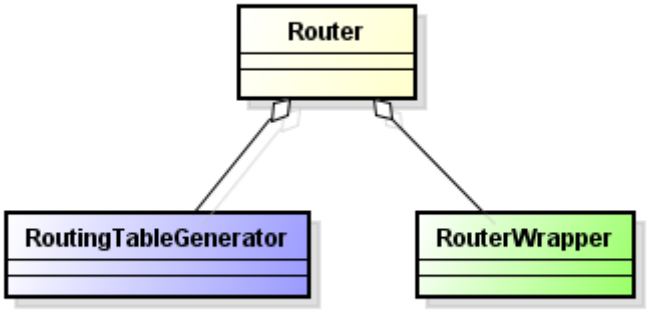

Рисунок 6. Структура программной реализации актора Router.

Актор SystemAnalyzer является составным актором, реализующим систему обнаружения вторжений и проводящим анализ компрометации системы. Его внутренняя структура которого отражена на рисунке 7.

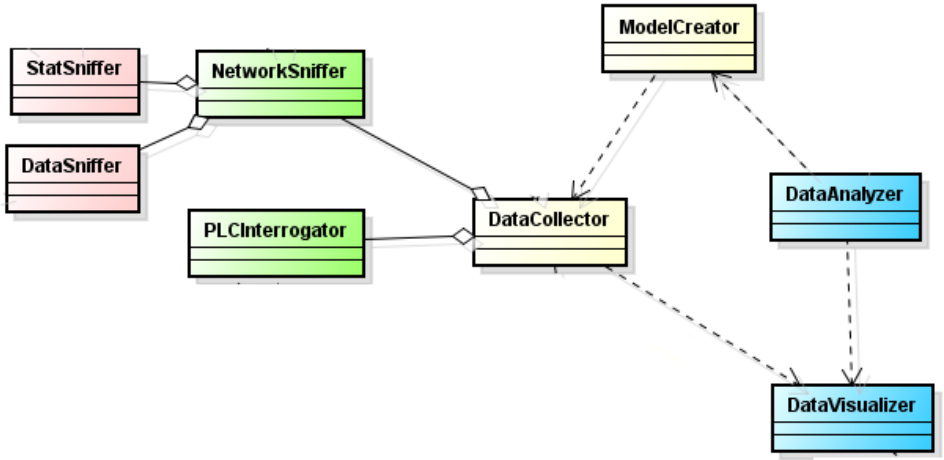

Рисунок 7. Структура программной реализации актора SystemAnalyzer.

Актор StatSniffer производит сбор сетевой статистики по протоколу NetFlow (или его аналогов). Данный протокол поддерживается большинством современных маршрутизаторов. Сбор производится посредством подключения к особому порту роутера Netflow. Если оборудование АСУ ТП не поддерживает протоколы класса NetFlow, то сбор

статистики будет осуществляться из программы на основе данных, полученных от DataSniffer.

Актор DataSniffer производит сбор фактической информации, распространяемой по сети, для её дальнейшего анализа на высоком уровне (например, Modbus). Данный модуль необходим, так как протокол NetFlow не производит сбор статистики для протоколов высокого уровня. Данные для работы данного модуля будут поступать по порту Mirror. Также актор StatSniffer может использовать для получения статистики программные средства (например сканер трафика Snort). Сетевая статистика может храниться локально или в специальной базе данных. Прототип структуры базы данных для хранения сетевой информации представлен на рисунке 8.

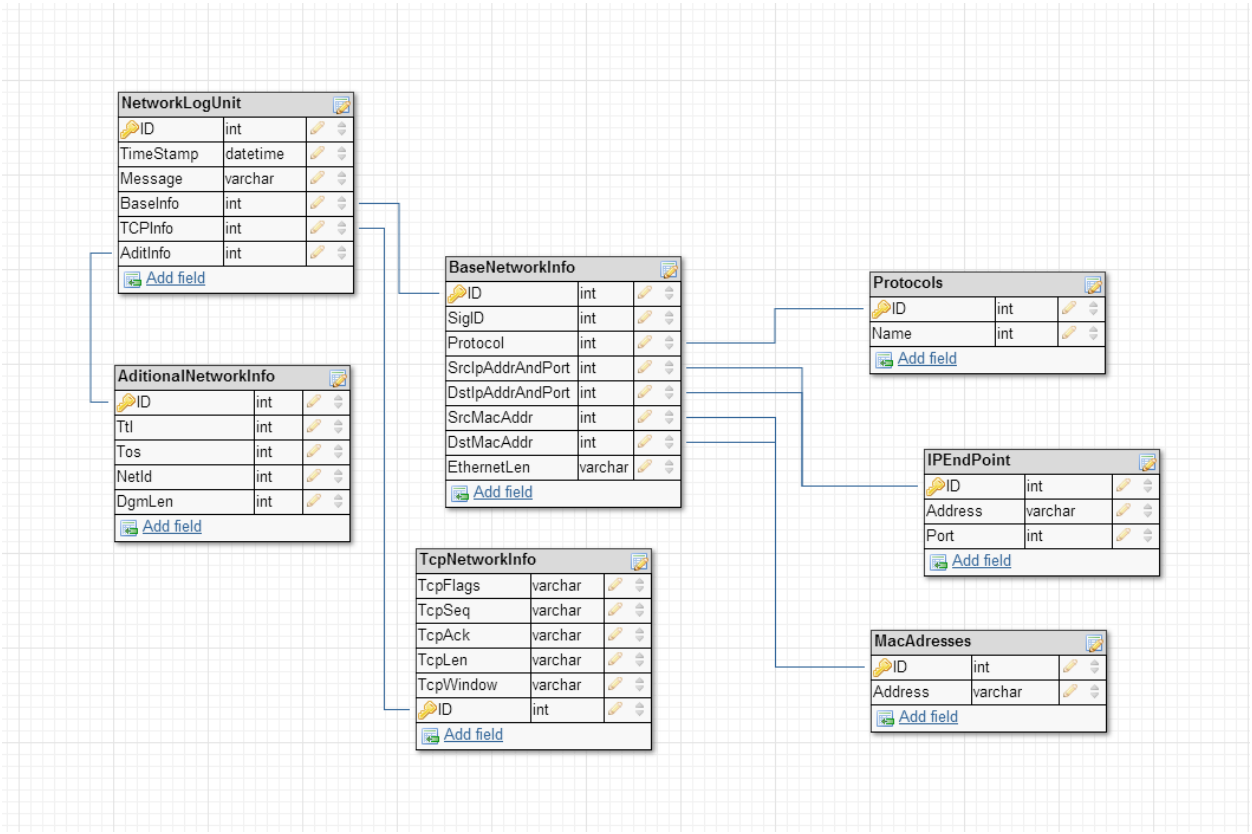

Рисунок 8. Структура базы данных сетевой информации.

Актор NetworkSniffer управляет работой акторов StatSniffer и DataSniffer и является “пассивной” частью коллектора сетевых данных.

Актор PLCInterrogator производит сбор информации о состоянии множества ПЛК. Функционал актора реализуется посредством запросов и обработки диагностических данных от ПЛК. Время отправки запросов может быть синхронизировано со временем проведения технологического перерыва (если данные диагностические функции затрагивают важные технологические процессы ПЛК и/или не могут быть выполнены

параллельно с технологическим процессом). Является активной частью коллектора сетевых данных.

Актор DataCollector собирает, обрабатывает и аккумулирует данные о состоянии сети и состоянии исследуемых характеристик PLC, а также управляет расписанием опроса PLC.

Актор ModelCreator создаёт сетевые матрицы активности и другие элементы эталонной модели, необходимые для вычисления критериев компрометации системы. Актор DataAnalyzer занимается аналитической обработкой и анализом полученных данных. Вычисляет критерии компрометации системы. Актор DataVisualizer занимается отображением собранных данных, а также информации о уровне компрометации SCADA.

На рисунке 9 представлен пример консольного интерфейса модуля анализа компрометации сети в составе системы обнаружения вторжений.

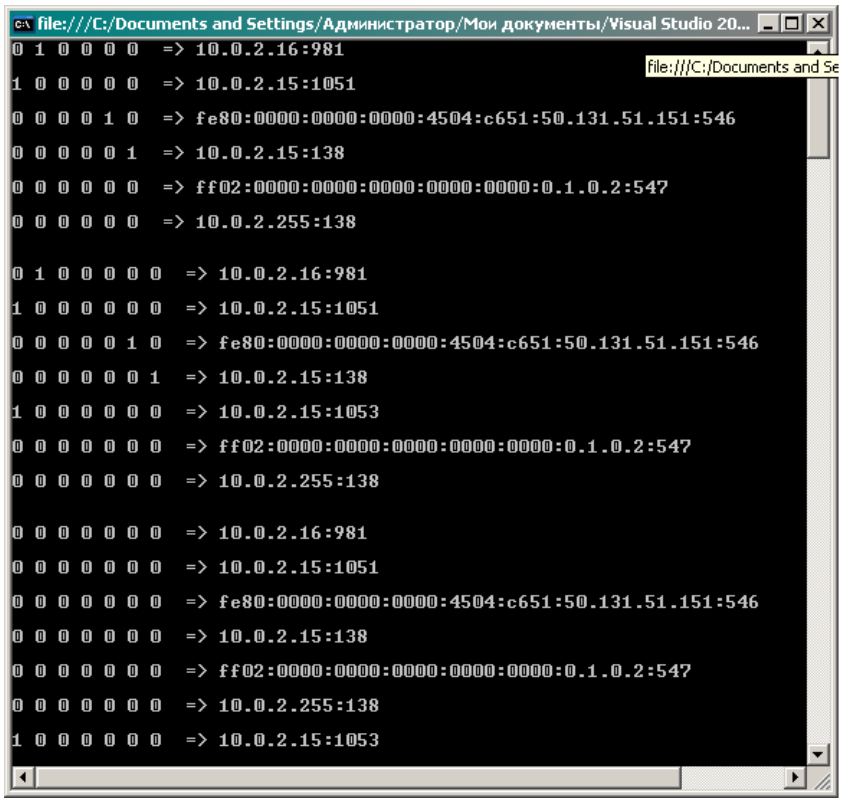

Рисунок 9. Интерфейс модуля анализа компрометации сети.

2.5 Апробация программной системы

Для апробации программной системы был создан виртуальный стенд SCADA для АСУЗ, предназначенной для интеллектуальных зданий (англ. “Smart house”) [43]. АСУТП и АСУЗ имеют схожую трёхуровневую архитектуру, включающую уровень конечных устройств (датчики, сенсоры), уровень управляющих контроллеров и уровень диспетчеризации и администрирования (человеко-машинные интерфейсы), в составе которой функционирует SCADA-система.

В качестве ОС контроллеров была использована система реального времени Windows Embedded Compact 7. Данная ОС широко используется в системах интеллектуальных зданий, построенных с использованием контроллеров Beckhoff серии CP62**. В качестве ОС человеко-машинного интерфейса была использована Windows XP Embedded, также широко используемая в автоматизированных системах управления. Система обнаружения вторжений, реализуемая в рамках данной работы, функционировала на отдельном компьютере под управлением Windows XP Embedded. Все три элемента инфраструктуры функционировали на отдельных компьютерах, объединённых в рамках одной виртуальной сети. Для создания виртуального стенда использовались программные пакеты виртуализации Oracle VirtualBox и Microsoft Virtual PC. Использование Microsoft Virtual PC продиктовано спецификой развёртывания платформ для устройств под управлением Windows Embedded Compact 7 для взаимодействия со средой разработки приложений Visual Studio 2008 SP2. В качестве программного аналога технологиям зеркалирования портов и Netflow для мониторинга сетевого взаимодействия отдельных компонентов сетевой инфраструктуры была использована программа Snort.

В качестве технологического процесса была реализована программная модель процесса аэрации жидкости в цистерне. Рассмотрим основные особенности реализации программной модели технологического процесса. В модели ПЛК (PLC) под управлением ОС Windows Embedded Compact 7 представлена цистерна, имеющая определённый объём (в литрах). Через равные промежутки времени в цистерну поступает некоторый случайный объём жидкости, ограниченный максимальным объёмом поступления в процентах от общего объёма. После того, как объём жидкости в цистерне превысит определённый максимальный порог свободного наполнения (в литрах), будет запущена процедура откачивания жидкости из цистерны. Порог свободного наполнения не превышает максимально допустимого объёма цистерны. Откачивание производится до определённого минимального порога жидкости (в литрах). Через заданные интервалы времени в цистерне запускается процедура аэрации жидкости (насыщения воздухом),

которая продолжается заданное время. Процедуры наполнения, откачивания и аэрации функционируют в отдельных потоках. Откачивание и аэрация производятся асинхронно. Наполнение цистерны производится постоянно. Параметры максимального порога свободного наполнения, минимального порога, скорости наполнения, скорости откачивания могут быть изменены оператором с использованием человеко-машинного интерфейса (HMI), связанного с PLC по сети. Модель HMI осуществляет отображение текущих параметров PLC (текущий объём жидкости в цистерне, текущие настройки) и имеет возможность изменения вышеперечисленных настроек. Сетевое взаимодействие компонентов осуществляется по протоколу TCP/IP. PLC является TCP клиентом, а HMI – TCP сервером.

На рисунке 10 представлен пример консольного интерфейса приложения, управляющего технологическим процессом аэрации, для ПЛК под управлением MS Windows Embedded Compact 7. На рисунке 11 представлен консольный интерфейс приложения, контролирующего технологический процесс аэрации, для человеко-машинного интерфейса под управлением MS Windows XP Embedded.

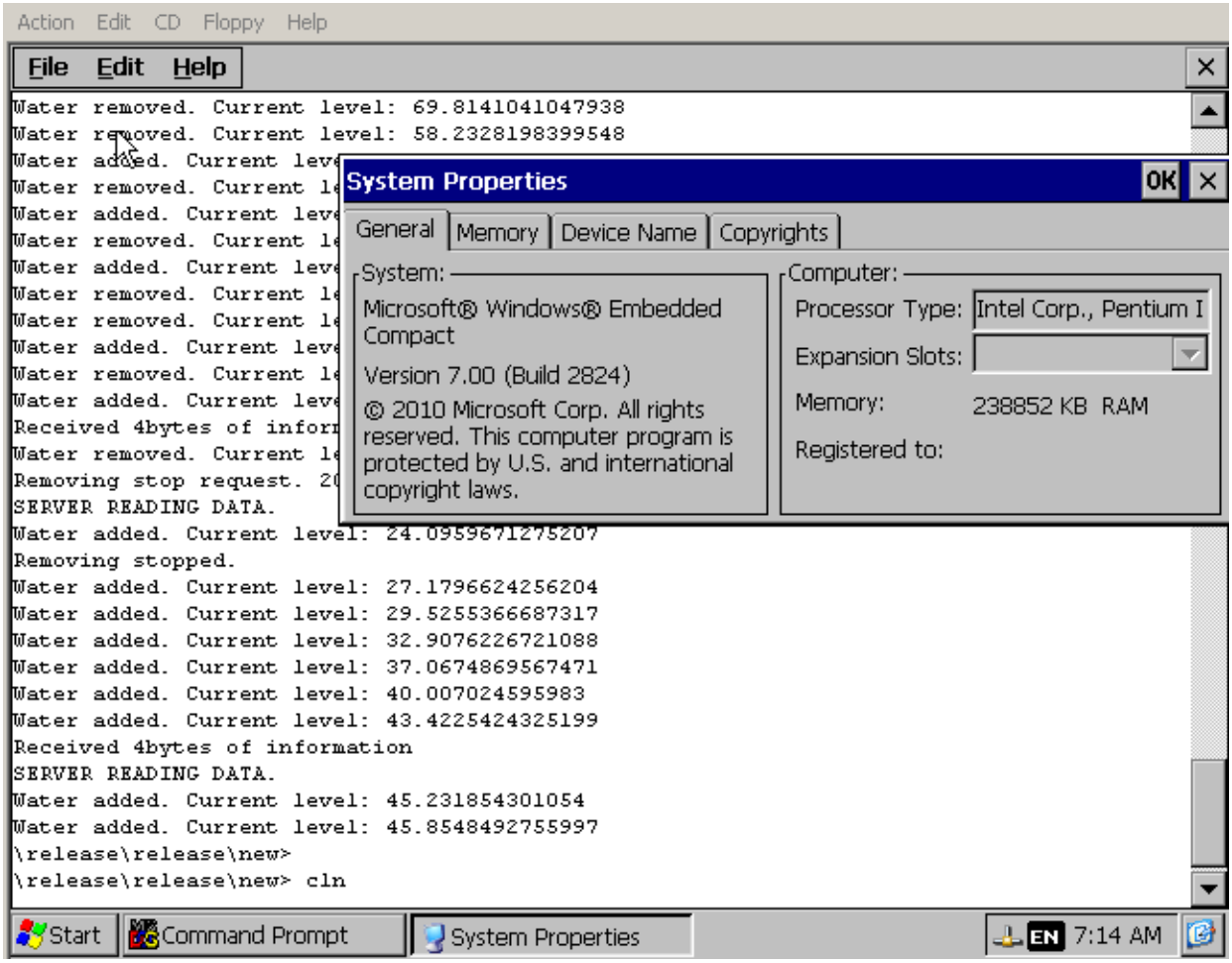

Рисунок 10. Программа для управления процессом аэрации на ПЛК.

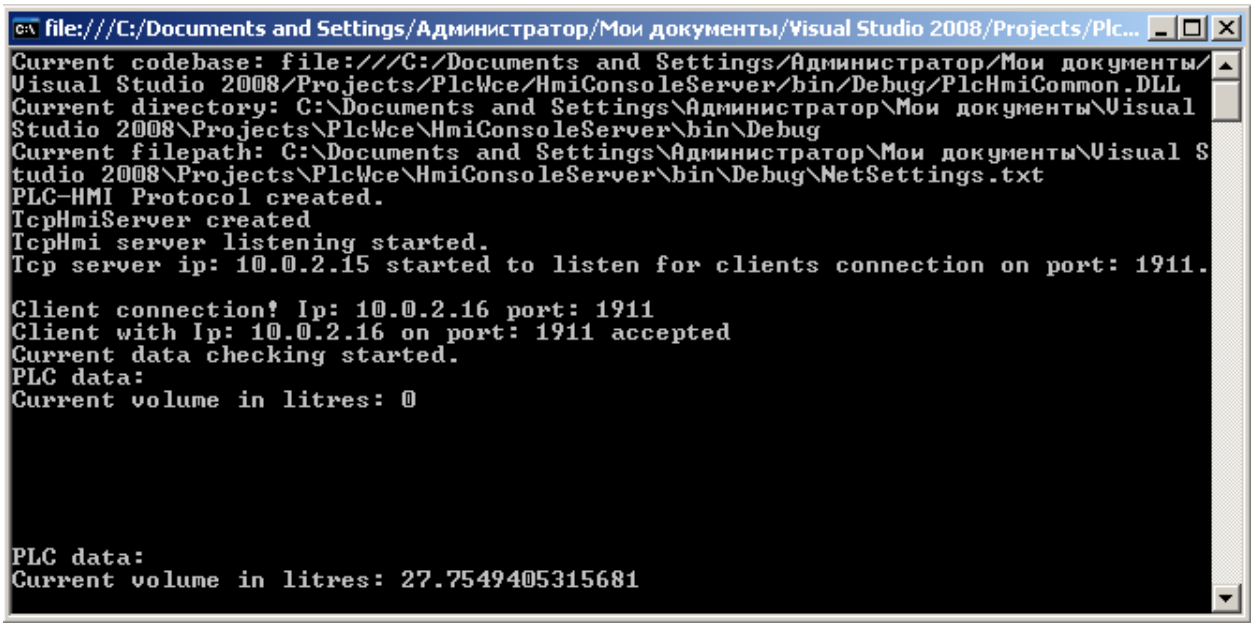

Рисунок 11. Программа для контролирования процесса аэрации на HMI.

2.6 Перспективы развития системы и дальнейшие исследования

Разрабатываемая система имеет широкие перспективы для дальнейшего развития, так как является составным модульным продуктом и не накладывает ограничений на число модулей анализа различных аспектов безопасности SCADA-систем. Основными перспективами развития системы являются следующие, а именно: создание модуля для оценки доверенного состояния после внесения изменений, создание модуля анализа и оценки действий оператора, создание модуля эвристического анализа для противодействия протяжённым во времени атакам, интеграция системы с системой оценки производственных рисков.

Основными направлениями дальнейших исследований являются следующие. Во-первых, необходимо проведение исследования по выявлению особенностей размещения и реализации сервиса слежения за допустимостью действий оператора на человеко-машинном интерфейсе (HMI). Во-вторых, необходимо проведение исследования по созданию обобщённого метода построения конечного автомата канала взаимодействия PLC и HMI. В-третьих, необходимо проведение исследования по особенностям работы системы в условиях обработки больших объёмов данных (т.н. Big Data).

Заключение

На сегодняшний день, вопрос о необходимости проведения активных исследований в области информационной безопасности АСУ ТП и, в частности, SCADA-систем стоит очень остро. Обширный список актуальных проблем информационной безопасности систем управления, описанный в работе, как нельзя лучше доказывает это утверждение.

В рамках данной работы был разработан метод имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем, предоставляющий возможность раннего обнаружения компрометации инфраструктуры SCADA-систем вредоносным ПО, повышая уровень их информационной защищённости.

Были достигнуты следующие результаты:

а) Специфицирована структура модели акторов для работы в SCADA-системах (для обеспечения оперативной оценки состояния).

б) На основе специфицированной модели акторов разработана модель SCADA-системы и её взаимодействия с системой обнаружения вторжений (СОВ).

в) Разработан метод ИМ эталонного состояния и поведения SCADA-систем для СОВ и его математическая основа.

г) Реализована модель сетевого взаимодействия PLC и HMI по протоколу TCP/IP.

д) Реализован модуль сбора сетевой статистики.

е) Реализован модуль анализа состояния сети.

ж) Реализован модуль анализа настроек оборудования.

По результатам работы имеются следующие публикации:

а) Барчан К. А. Разработка метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов // 52-я Международная научная студенческая конференция «Студент и научно-технический прогресс». – Новосибирск : НГУ, 2014.

б) Барчан К. А. Разработка метода имитационного компьютерного моделирования эталонного состояния и поведения SCADA-систем на основе модели акторов // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2014. вып. 1.

Перечень условных обозначений

Таблица 1. Перечень условных обозначений.

Обозначение	Определение
SCADA	Supervisory control and data acquisition, диспетчерское управление и сбор данных.
SCADA-система	Система диспетчерского управления и сбора данных.
ПО	Программное обеспечение.
АСУ ТП	Автоматизированная система управления технологическим процессом.
ИБ	Информационная безопасность.
СУ	Система управления.
СЗИ	Система защиты информации.
СОВ	Система обнаружения вторжений.
PLC	Programmable logic controller, программируемый логический контроллер.
ПЛК	Программируемый логический контроллер.
HMI	Human-machine interface, человеко-машинный интерфейс.
TCP/IP	Transmission Control Protocol (TCP) (Transmission Control Protocol, протокол управления передачей) и Internet Protocol (IP) (Internet Protocol, межсетевой протокол).
АСУП	Автоматизированная система управления предприятием.
ИСУПП	Информационная система управления производственными процессами.
DCS-системы	Distributed control systems, распределенные системы управления.
АСКУЭ	Автоматизированной системы контроля и учёта энергоресурсов.
АСУЗ	Автоматизированная система управления зданием.
СУБД	Система управления базами данных.
MTU	Master terminal unit, главный сетевой терминал.
RTU	Remote terminal unit, устройство связи с объектом.
API	Application programming interface, интерфейс программирования приложений.
ОС	Операционная система.
UI	User interface, пользовательский интерфейс.

Продолжение таблицы 1.

ПАЗ	Противоаварийная защита.
ИМ	Имитационное моделирование.
МА	Модель акторов.
ООП	Объектно-ориентированное программирование.

Приложение А

(рекомендуемое)

Свойства компонентов управления SCADA

Таблица А.1. Свойства компонентов управления SCADA.

Компонент	**Описание**
Управляющий сервер.	Располагает комплектом управляющих алгоритмов и программ для системы управления. Данное ПО осуществляет связь с контрольными устройствами низших уровней. Координирует работу всех контрольных модулей в АСУТП. Может включать в себя сервер ввода-вывода в качестве связующего звена.
SCADA-сервер или главный сетевой терминал (MTU).	SCADA-сервер представляет собой ведущее устройство SCADA-системы. Устройства связи с объектом и PLC-контроллеры, расположенные в удаленных точках, являются подчиненными устройствами.
Устройство связи с объектом (RTU) (RTU-устройства).	Это устройства управления и сбора данных. Они предназначаются для управления удаленными объектами масштабных SCADA-систем. Если PLC используются в качестве периферийного и выполняет роль RTU, тогда этот контроллер также называют RTU.
Программируемый логический контроллер (PLC, ПЛК).	ПЛК представляет из себя промышленный компьютер, предназначенный для логического контролирования и выполнения процесса работы электрического оборудования и аппаратуры (например, реле, клапанов, переключателей, шлюзов, механических таймеров и др.). ПЛК может управлять в том числе и комплексными процессами. В среде SCADA-систем ПЛК чаще являются периферийными устройствами.
Интеллектуальные электронные устройства (IED).	IED - это датчики и исполнительные механизмы, наделенные программными и аппаратными возможностями, необходимыми для обеспечения коммуникаций с другими устройствами и сбора данных, а также управления и исполнения локальных производственных процессов.
Человеко-машинный интерфейс (HMI).	HMI – это программно-аппаратный элемент, позволяющий оператору наблюдать и анализировать статус производственного процесса, изменять его настройки и вносить в его работу корректировки, а также вручную управлять им (в случае ЧП). HMI позволяет изменять ограничения и параметры контроллеров, а также влиять на ход их работы. HMI отображает данные о статусе и истории работы технологического процесса, производственную и техническую информацию. HMI может быть как специальным устройством, так и ноутбуком или интернет-браузером, подключенным к интернету.
Журнал данных.	Журнал данных представляет из себя базу данных, служащую для записи и хранения всей информации о процессах в рамках АСУ ТП (SCADA-системы). Информация, полученная из журнала данных, может быть использована при создании отчётов или статистики, планировании производства на корпоративном уровне, расследовании инцидентов и др [3].

Приложение Б

(рекомендуемое)

Различия ИТ-систем и АСУ ТП

Таблица Б.1. Сводка по различиям ИТ-систем и АСУ ТП.

Категория	Традиционная IT-система	АСУ ТП
Требования по управлению рисками.	Конфиденциальность и целостность данных являются первостепенными факторами. Отказоустойчивость менее важна, так как непродолжительный простой не является основным риском. Таковым является риск задержки бизнес-операций.	Человеческая безопасность является первостепенной. За этим фактором по важности следует защищённость технического процесса. Отказоустойчивость является основным фактором, даже непродолжительный простой может быть недопустим. Основные последствия реализации рисков – это несоблюдение нормативов и законов, негативное воздействие на окружающую среду, гибель людей, поломка оборудования, остановка производства.
Архитектур-ная задача безопасности.	Основная задача – это обеспечивать безопасность информационных активов – информации хранимой и распространяемой в рамках бизнес-процессов. Центральный сервер может требовать большей защиты.	Основная цель – защищать конечные устройства (например, полевые устройства, PLC). Защита центрального сервера также важна.
Непредвиден ные последствия.	Решения в области безопасности проектируются в отношении типичных ИТ-систем.	Инструменты безопасности должны быть протестированы (например, в режиме офлайн на аналогичной АСУ ТП) для доказательства того, что они не нарушают нормального функционирования АСУ ТП.
Критичное по времени взаимодей-ствие.	Экстренное взаимодействие менее критично. Строго ограниченный контроль доступа может быть реализован до степени, необходимой для безопасности.	Реализация отклик и его время на управляющую команду персонала или любое другое экстренное взаимодействие АСУ ТП критично. Доступ к компонентам АСУ ТП должен строго контролироваться, однако не должен препятствовать экстренному взаимодействию персонала и техники.
Операцион-ные системы.	Системы проектируются для использования с стандартными ОС. Обновления производятся простым способом с помощью	Нестандартные операционные системы (возможно, проприетарные), часто без встроенных или предусмотренных

Продолжение таблицы Б.1.

	доступных средств автоматического развёртывания.	возможностей для информационной безопасности. Изменения в программном обеспечении должны проводиться осторожно, зачастую с помощью специалистов официального поставщика оборудования, ввиду специализированных алгоритмов управления и, возможно, необходимости модификации аппаратной или программной части.
Ограничения ресурсов.	Системы отличаются наличием достаточного количества вычислительных ресурсов для поддержки дополнительных сторонних приложений, таких как программные решения в области ИБ.	Системы были спроектированы для поддержки исключительно технического процесса и могут не иметь достаточного количества памяти и/или вычислительных ресурсов для поддержки дополнительных средств безопасности.
Связь и коммуника-ции.	Стандартные протоколы связи. Используются, в первую очередь, средства проводной связи с некоторым количеством локальных возможностей в области беспроводной связи. Используются стандартные практики при построении архитектуры и реализации сетевой инфраструктуры.	Используется множество протоколов передачи данных - как стандартных, так и проприетарных. Используется несколько видов средств передачи информации, включая специализированные проводные и беспроводные (радио и спутниковые). Инфраструктура сетей отличается сложностью и иногда требует экспертных оценок со стороны квалифицированных инженеров.
Управление изменениями.	Изменения программного обеспечения производятся регулярно согласно текущей политике безопасности. Процедуры обычно автоматизированы.	Изменения программного обеспечения должны быть тщательно тестироваться и внедряться в систему поэтапно (“инкрементно”), дабы гарантировать целостность и правильную работу АСУ ТП. Плановые работы на АСУ ТП должны планироваться заранее за дни/недели до их проведения. В АСУ ТП могут использоваться ОС, которые более не поддерживаются.
Управляемая поддержка.	Допускается использование разнообразных практик поддержки разными исполнителями.	Сервисная поддержка обычно осуществляется средствами одного поставщика оборудования.
Жизненный цикл компонентов.	Жизненный цикл компонентов, как правило, занимает от 3 до 5 лет.	Жизненный цикл компонентов, как правило, занимает от 15 до 20 лет.

Продолжение таблицы Б.1.

Доступ к компонентам.	Компоненты обычно являются локальными и легкодоступными.	Компоненты могут быть изолированы, территориально удалены, и требовать больших физических усилий для получения к ним доступа [3].

Приложение В

(рекомендуемое)

Основные уязвимости и факторы риска АСУ ТП и SCADA-систем

Таблица В.1. Основные уязвимости АСУ ТП.

Уязвимость	**Описание**
Уязвимости политик и процедур	1) Не отвечающая требованиям политика безопасности для АСУ ТП. 2) Отсутствие формального обучения в области безопасности АСУ ТП. 3) Не отвечающая требованиям архитектура и дизайн системы безопасности. 4) Не было разработано специализированных или документированных процедур в рамках политики безопасности для АСУ ТП. 5) Не было разработано специализированных или документированных процедур в рамках политики безопасности для АСУ ТП. 6) Отсутствие или недостаточность руководящих документов по особенностям работы с оборудованием в рамках АСУ ТП. 7) Отсутствие механизма административных действий в области безопасности. 8) Отсутствие или малое количество действий по проведению аудита АСУ ТП. 9) Отсутствие на АСУ ТП документированного плана или набора процедур для осуществления защиты и восстановления инфраструктуры в случае катастрофы. 10) Отсутствие на АСУ ТП специфицированной конфигурации по управлению изменениями.
Аппаратные **у**язвимости платформы.	1) Обновления ОС и специализированного программного обеспечения от поставщика оборудования могут не проводиться до того момента, как уязвимость уже была найдена. 2) Не поддерживаются обновления безопасности ОС и приложений. 3) Проведение обновления для операционных систем и приложений без исчерпывающего тестирования. 4) Использование стандартных конфигураций. 5) Критические конфигурации не хранятся или не подвергаются резервному копированию.

Продолжение таблицы В.1.

	6) Незащищённость данных на портативных устройствах. 7) Отсутствие адекватной политики в области парольной защиты. 8) Пароли не используются. 9) Раскрытие пароля, а также подбор или “угадывание” паролей 10) Применение не отвечающих требованиям средств управления доступом. 11) Не отвечающее требованиям тестирование изменений в области безопасности. 12) Не соответствующая требованиям физическая защита критически важных систем. 13) Посторонние лица среди персонала имеют физический доступ к оборудованию. 14) Небезопасный доступ к компонентам АСУ ТП. 15) Использование сетевых плат с двумя портами. 16) Недокументированные средства. 17) Радиочастотные или электромагнитные импульсы. 18) Отсутствие резервного питания. 19) Потеря контроля над средой. 20) Отсутствие дублирующих устройств для критически важных компонентов.
Программные уязвимости платформы.	1) Переполнение буфера 2) Установленные возможности для обеспечения безопасности не включены по умолчанию. 3) Отказ в обслуживании (DoS). 4) Неправильное реагирование на неопределённые, плохо определённые или неправильные условия. 5) Технология связывания и внедрения объектов в другие документы OLE для управления процессами (OPC) основывается на использовании удалённого вызова процедур (RPC) и технологию распределённой компонентной объектной модели DCOM. 6) Использование небезопасных индустриальных протоколов АСУ ТП. 7) Использование незашифрованных данных в рамках протокола. 8) Функционирование ненужных служб.

Продолжение таблицы В.1.

	9) Использование проприетарного программного обеспечения, обсуждаемого на конференциях и в журналах. 10) Не соответствующие требованиям механизмы аутентификации и контроля доступа в программном обеспечении для конфигурирования и программирования. 11) Программное обеспечение по обнаружению/предотвращению вторжений не установлено. 12) Ведение системных журналов (т.н. “логирование”) не осуществляется. 13) Инциденты не определяются. 14) Не установлено программное обеспечение по защите от вредоносных программ. 15) Программное обеспечение по защите от вредоносного ПО устарело или имеет устаревшие сигнатуры вирусов (уязвимостей, атак). 16) Программное обеспечение по защите от вредоносного ПО было внедрено без тщательного тестирования.
Сетевые уязвимости.	1) Слабая архитектура безопасности сети. 2) Не реализована система контроля информационных потоков. 3) Неправильно настроенное оборудование для обеспечения безопасности. 4) Настройки сетевого оборудования не хранятся и не подвергаются резервному копированию. 5) Пароли не шифруются на этапе передачи. 6) Пароли существуют неопределённый срок на сетевых устройствах. 7) Применяется не удовлетворяющая стандартам система контроля доступа. 8) Не удовлетворяющая стандартам физическая защита сетевого оборудования. 9) Небезопасные физические порты. 10) Потеря контроля над средой. 11) Неуполномоченные сотрудники имеют доступ к оборудованию и сетевым соединениям. 12) Отсутствие дублирующих устройств у критически важных сетевых компонентов.

Продолжение таблицы В.1.

	13) Не обозначен периметр безопасности. 14) Межсетевые экраны не используются или настроены неправильно. 15) Управляющие сети используются для передачи сетевого (“неуправляющий”) трафика. 16) Управляющие сервисы сети запущены не в рамках управляющей сети. 17) Не удовлетворяющие требованиям журналы межсетевого экрана и маршрутизатора. 18) Не производится мониторинга сети АСУ ТП. 19) Не определены критические пути мониторинга и контроля. 20) Стандартные, хорошо документированные протоколы используются открытым текстом. 21) Аутентификация пользователей, данных, устройств является некачественной или не существует. 22) Отсутствие для связи проверки целостности. 23) Не удовлетворяющая требованиям аутентификация между клиентами и точкой доступа. 24) Не удовлетворяющая требованиям защита данных между клиентами и точками доступа [3].

Таблица В.2. Основные факторы риска АСУ ТП.

Фактор риска	Описание
Внедрение стандартизованных протоколов и технологий со списком известных уязвимостей.	Детали реализации проприетарных протоколов и их спецификации публикуются для поддержки возможности для сторонних производителей создавать совместимые расширения. Отмечается тенденция перехода с проприетарных систем на стандартизированные технологии (ОС Microsoft Windows и Unix-подобные операционные системы, а также общие сетевые протоколы (TCP/IP, HTTP). Целью перехода является сокращение расходов, повышение производительности, обеспечение расширяемости. Использование открытых стандартов предоставляет экономические и технологические

Продолжение таблицы В.2.

	преимущества, однако увеличивает восприимчивость АСУ ТП по отношению к угрозам и инцидентам киберпреступности.
Связанность сети системы управления с другими сетями (например, сетью ИТ-системы).	Обуславливается необходимостью наличия удалённого доступа к АСУ ТП с целью мониторинга системы управления извне управляющей сети. Корпоративные сети и сети АСУ ТП объединяются для получения доступа к данным о текущем статусе производственных систем и отправки инструкций касательно производства или распространения продукции. Интеграция сетей увеличивает доступность СУ для реализации возможных уязвимостей. Данные уязвимости могут подвергнуть все уровни архитектуры сети АСУ ТП рискам появления ошибок, связанных со сложностью, рискам атаки со стороны злоумышленников и ряду других кибер-угроз (внедрение вирусного и вредоносного ПО и т.д.).
Широкая доступность и распространение технической информации и документации о системах управления.	Информация по архитектуре и дизайну АСУ ТП, их обслуживанию, структуре и коммуникациям, находящаяся в открытом доступе в сети Интернет, необходима для поддержки конкуренции среди продуктов ПО и возможности использования открытых стандартов. В открытом доступе находятся инструменты, помогающие разрабатывать ПО, реализующее различные стандарты из области АСУ ТП. Подобные данные и средства упрощают получение несанкционированного доступа к SCADA-системе.
Высокие риски проведения аудита.	Аудит АСУ ТП, предполагающий попытки взлома системы (например, тестирование по методу “чёрного” или “серого” ящика в рамках аудита на возможность проникновения) должен осуществляться опытными системными инженерами. Неосторожные действия могут привести к нарушению работоспособности системы, простоям и компрометации её безопасности.

Продолжение таблицы В.2.

	Инструменты и методы аудита должны тестироваться до их применения на действующей АСУ ТП (например, на автономных сопоставимых макетах АСУ ТП). Однако, создание подобного макета действующего оборудования может быть невозможно ввиду его дороговизны или отсутствия. В таком случае риск проведения аудита может превышать риск возможных последствий.
Высокие риски внесения исправлений.	Большинство обновлений программного обеспечения АСУ ТП и SCADA требуют выключения и перезагрузки оборудования, осуществляющего технологический процесс. Некоторые из них могут вывести из строя или удалить некоторый программный функционал, на основе которого функционировала система управления. К примеру, одной из уязвимостей, используемых вирусом Stuxnet, был пароль в базе данных Siemens' WinCC SQL, занесённый в исходные коды приложения в незашифрованном виде [3].

Приложение Г

(справочное)

Описание процедур проведения основных атак на SCADA-системы

Первым шагом злоумышленника при атаке *эксплойтом*, как правило, является взлом и взятие под контроль некоторого элемента корпоративной сети, с которого впоследствии производится последующая атака элемента внутренней сети. Примером может служить атака через сервер системы системного анализа и разработки программ (SAP) к серверу логирования, расположенному во внутренней сети. Взлом может быть осуществлён во время приёма/передачи данных системных журналов, дневной статистики, данных о текущих заказах, данных о текущем спросе и др. Имея в распоряжении сервер логирования (как правило, устройство класса Windows Server), злоумышленник может вывести его из строя, скрыть предыдущие атаки, получить доступ к чтению и редактированию конфиденциальных данных. Следующим шагом атаки злоумышленника является попытка взлома одного из элементов управляющей сети (SCADA-системы), выполняемая с использованием ранее захваченного элемента внутренней сети. Атакующий продвигается от сервера логирования к станции HMI, расположенной внутри управляющей сети. Проникновение может быть осуществлено во время обмена данными системных журналов и системной статистики, а также посредством протокола OPC и сервисов домена (domain services). Захват станции HMI (как правило, устройство класса Windows Workstation) позволяет злоумышленнику управлять настройками, манипулировать данными о текущем технологическом процессе в целях обмана сотрудников. Необходимо отметить, что для создания мнимой картины функционирования технологического процесса необходим захват и синхронизированное управление всех станций HMI, что предполагает наличие межпрограммного взаимодействия. Далее атака может идти в направлении сервера приложений, обслуживающего управляющие рабочие станции, посредством информационного обмена в рамках чтения/записи текущих параметров процесса и настроек, аварийных оповещений, диагностики управляющей шины. Управление сервером приложений (как правило, устройством класса Windows Server) позволяет осуществлять фальсификацию данных о технологическом процессе для вышестоящих компонентов (например, сервера логирования), нарушать синхронизацию отдельных компонентов SCADA-системы или продолжить атаку в направлении управляющих рабочих станций или PLC. Захват управляющей рабочей станции (как правило, устройства класса Windows Server или Windows Workstation) позволяет злоумышленнику получить доступ к PLC. Доступность

PLC позволяет управлять его работой, просматривать, модифицировать и обновлять ПО для PLC [7].

Наиболее вероятный сценарий запланированной атаки *инсайдером* на АСУ ТП включает в несколько этапов. На первом этапе происходит создание дополнительного контура управления системой для перехвата управления и вызова аварийной ситуации. Как правило, подобные действия совершаются от имени подставного пользовательского (операторского) аккаунта в целях сокрытия следов действий инсайдера. После проведения атаки управляющий контур самоуничтожается, стирая максимально возможно количество информации в системных журналах. Выполнение данных действий может включать необходимость изменения ПО контроллеров (например, контроллера ПАЗ), что может быть осуществлено при помощи программного скрипта, выполняющего записанные ранее действия. На втором этапе ключевые элементы АСУ ТП заражаются вредоносным ПО, позволяющим в нужный момент вывести оборудование из строя или нарушить работоспособность компьютера, имитирую атаку устройства злоумышленником. Дальнейшие действия инсайдера могут развиваться по следующей схеме. В запланированное сотрудником-инсайдером время начинается имитация атаки компьютеров АСУ ТП злоумышленниками. Нормальная работа станций управления нарушается, сотрудники предприятия оказываются в замешательстве. Критически важное управляющее оборудование самостоятельно отключается, аварийные блокировки не срабатывают, производственное оборудование остаётся без управления. Выполнение технологического процесса нарушается, возникает существенный риск создания ЧП. Во время процедуры расследования подобного инцидента будет выявлено заражение станций вредоносным ПО, отмечены действия операторов по созданию аварийной ситуации, а также выявлен отказ системы противоаварийной защиты. Таким образом, вина в произошедшем событии ложится на плечи неизвестного злоумышленника, заразившего систему вирусом и атаковавшего её. Злоумышленник-инсайдер остаётся вне подозрений [8].

Для примера атаки *вирусом-червём* рассмотрим классического представителя данной категории вирусов – Stuxnet. В качестве первого из аргументов в пользу его сложности необходимо отметить факт того, что вирус способен распространяться тремя совершенно разными путями, а именно:

а) Посредством инфицированных отчуждаемых носителей данных (например, USB-флеш-накопитель).
б) Посредством трафика внутри локальной сети.

в) Посредством инфицированных файлов проекта Siemens.

Вирус инфицирует компьютеры посредством USB-флеш-накопителей (даже в случае выключенного автозапуска) путём ранее неизвестной уязвимости (MS10-046), связанной с ярлыками (файлами с расширением *.lnk). Версии Stuxnet, выпущенные до марта 2010, распространялись при помощи USB-флеш-накопителей путём уязвимости, связанной с автозапуском, нежели с расширением *.lnk. Вирус распространяется по локальной сети на компьютеры с наличием сетевых ресурсов в общем доступе путем регистрации всех учетных записей пользователей компьютера и домена. Затем программа пытается использовать все доступные сетевые ресурсы для того, чтобы скопировать и выполнить себя на удалённом ресурсе, тем самым заражая удалённый компьютер. Вирус распространяется по локальной сети, предоставляя сервис печати с помощью уязвимости нулевого дня в Windows Print Spooler (MS10-061). Вирус распространяется по локальной сети посредством уязвимости MS08-067 Windows Server Service Vulnerability (MS08-067). Вирус инфицирует компьютеры, использующие базу данных Siemens WinCC, с помощью внутренних неизменяемых системных паролей, подключаясь к SQL-серверу в целях передачи и исполнения копии вируса. Вирус распространяется, копируя себя в любые найденные файлы проектов Siemens STEP 7 (файлы с расширением *.S7P, *.MCP и *.TMP), а затем исполняясь автоматически при открытии проекта [9].

Приложение Д

(обязательное)

Особенности актуальных проблем ИБ SCADA-систем

Таблица Д.1. Особенности актуальных проблем ИБ SCADA-систем.

Проблема ИБ	Особенности
Физическая изоляция (т.н. “воздушный зазор”) сетей перестала быть эффективной мерой информационной безопасности.	Изоляция компонентов SCADA от остальной сети, как всех вместе, так и в отдельности, не обеспечивает их безопасность, не защищает от угроз заражения вредоносным кодом. Инцидент заражения может произойти посредством физического контакта с оборудованием в процессе обслуживания, например, при проведении обновления посредством таких устройств как USB-флеш-накопитель, дисковый носитель информации (к примеру, CD или DVD) и т.д.) [9].
Объединение АСУ ТП и IT-инфраструктуры предприятия влечёт за собой увеличение доступности компонентов SCADA-системы для атак извне.	Слияние воедино на сетевом уровне отдельных систем предприятия, имеющих различные требования в области информационной безопасности, увеличивает связанность сетевой инфраструктуры, порождает новые маршруты для взаимной достижимости отдельных компонентов объединённой сети [18]. Данный факт, в свою очередь, увеличивает риски проявления ряда возможных уязвимостей, связанных с внешними атаками, ориентированными на SCADA-системы [7].
Наличие возможности удалённого доступа с высоким уровнем привилегий к компонентам SCADA-систем.	Наглядным примером данной проблемы может служить наличие у инженера-настройщика PLC удалённого доступа с высоким уровнем привилегий. Необходимость высокого уровня привилегий, как правило, обусловлена особенностями в работе операционной системы и/или SDK аппаратной платформы при проведении обслуживания (настройки, обновления и т.д.) оборудования. Необходимость реализации возможности удалённого доступа зачастую бывает продиктована такими обстоятельствами, как, например, отсутствие квалифицированных инженеров в штате предприятия и использование услуг сторонних

Продолжение таблицы Д.1.

	специалистов, затруднённость физического доступа к оборудованию благодаря его удалённости и др. [7, 9].
Уязвимость от атак изнутри корпоративной сети злоумышленным служащим (т.н. “insider”) за счёт частичного или полного отсутствия в инфраструктуре SCADA специализированных средств мониторинга и контроля деятельности сотрудников, политик безопасности, средств авторизации и аутентификации и др. [18].	Данный факт обусловлен невозможностью внедрить вышеперечисленные средства обеспечения информационной безопасности благодаря повышенным рискам нарушения работоспособности системы [7, 9, 20].
Негативное влияние системы защиты информации (СЗИ) на информационные процессы АСУ ТП (задержки в обработке команд и запросов, прекращение информационного обмена и др.).	Одним из основных требований к работе любой SCADA-системы является повышенная отказоустойчивость и возможность получить данные или исполнить команду за гарантированное время. В рамках данной особенности дополнительная вычислительная нагрузка на элементы SCADA-системы может быть недопустимой [3].
Сложность или невозможность постоянного обновления ПО и/или внесения исправлений.	Данная проблема, как правило, обусловлена техническими или организационными сложностями, такими как отсутствие квалифицированного персонала, невозможность или дороговизна остановки технического процесса для проведения перезагрузки ОС [21].
Проведение технического аудита (тесты на проникновение, инструментальные проверки и др.) характеризуется высокой степенью риска.	Проблема обусловлена многими факторами. В первую очередь, это отсутствие гарантий сохранения работоспособности реальной системы при проведении технического аудита с целью моделирования действий хакера или злоумышленного служащего. В целях снижения рисков выхода из строя при тестировании, реальная система может быть заменена аналогом посредством аппаратного (тестовые стенды) или программного (виртуализация и моделирование) макетирования. Однако, и в случае с макетированием не удаётся полностью избежать возможных проблем.

Продолжение таблицы Д.1.

	В отношении аппаратного макетирования актуальными становятся финансовые проблемы, проявляющиеся в необходимости фактического дублирования оборудования тестируемой системы. Виртуализация отличается сложностью своей настройки и имитации специального оборудования (PLC, RTU и др.). Создание низкоуровневой модели всех структурных компонентов и их взаимодействия отличается крайней сложностью реализации и высокими трудозатратами. Высокоуровневое моделирование, в свою очередь, отличается проблемами, связанными с неполнотой рассматриваемой модели за счёт её искусственного упрощения. Проблему рисков проведения аудита зачастую дополнительно усугубляет факт отсутствия у аудиторов (например, при проведении тестов на проникновение) опыта и навыков работы со специализированным оборудованием (например, различными видами PLC, RTU и др.) [22].
Отсутствие в архитектуре и реализации системы управления компонентов, отвечающих за безопасность и аутентификацию, или трудоёмкость (невозможность) их внедрения.	Данная проблема особенно актуальна для относительно старых АСУ ТП, разрабатывавшихся без учёта необходимости интеграции с IT-инфраструктурой предприятия в будущем [18, 23].
Принцип “Безопасность через неясность” (англ. “Security through obscurity”) становится неактуальным в условиях современного рынка программного обеспечения.	Открытость современных SCADA-систем в целях расширяемости и конкурентоспособности влечёт за собой доступность всевозможных системных спецификаций, что значительно облегчает процесс изучения системы в целях создания вредоносного кода направленного действия [23].
Отсутствие или несовершенство нормативно-правовой базы.	Нормативно-правовые документы по принципам проведения аудита безопасности компонентов АСУ ТП, устранения выявленных в процессе аудита ошибок и уязвимостей, немедленного реагирования в случае обнаружения атаки, выявлению и расследованию инцидентов

Продолжение таблицы Д.1.

	информационной безопасности и др. отсутствуют или недоработаны [24, 25].
Протоколы АСУ ТП и SCADA не имеют детализации на уровне соединения.	С точки зрения номера используемого порта, сообщение, направленное на чтение данных, выгладит абсолютно так же, как и сообщение, направленное на обновление программной составляющей устройства. Таким образом, в случае наличия разрешающего правила на обмен командами (сообщениями) чтения данных от HMI до PLC, разрешающее правило на обмен командами конфигурирования и/или программирования присутствует автоматически. Одним из возможных решений данной проблемы является внедрение в инфраструктуру SCADA средств “Deep Packet Inspection (DPI)” (например, Modbus DPI firewall), контролирующих трафик PLC [26].

Литература

1. Рождественский Д. А. Автоматизированные комплексы распределённого управления: Учебное пособие / - Б.м., ТМЦДО, 2002.
2. Матвейкин В. Г., Фролов С. В., Шехтман М. Б. Применение SCADA-систем при автоматизации технологических процессов. – М.: Машиностроение, 2000.
3. Stouffer K., Falco J., Scarfone K. Guide to Industrial Control Systems (ICS) Security. [Электронный ресурс]. National Institute of Standards and Technology Gaithersburg. Gaithersburg, Maryland, USA. 2011. URL: http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf (дата обращения: 30.12.2013).
4. Torre G. SIMATIC IT Reference Book. [Электронный ресурс]. Siemens AG Industry Sector Industry Automation. 2012. URL: https://mes-simaticit.siemens.com/res/html/go2market/Documents/References_book_web.pdf (дата обращения: 30.12.2013).
5. Куцевич Н. А. SCADA-системы и муки выбора. // Мир компьютерной автоматизации. №1. 1999. С.72-78.
6. Максимов В. В. Интеграция и организация единого информационного пространства систем корпоративного и технологического управления. В кн.: Материалы III профессионального форума “Информационные технологии и измерение в электроэнергетике”. М.: ЦМТ, 2008. [Электронный ресурс]. URL: http://it.e-m.ru/08/prezentation/maksimov.pdf (дата обращения: 30.12.2013).
7. Meixell B., Forner E. Out of Control: Demonstrating SCADA Exploitation // Black Hat 2013. [Электронный ресурс]. Black Hat Conference. Las Vegas, Nevada, USA. 2013. URL: https://media.blackhat.com/us-13/US-13-Forner-Out-of-Control-Demonstrating-SCADA-Slides.pdf (дата обращения: 30.12.2013).
8. Глебов О. А. Защита автоматизированных систем управления промышленных предприятий. // Мобильные телекоммуникации. №6. 2012. С.20-21.
9. Byres E., Howard S. Analysis of the Siemens WinCC / PCS7 “Stuxnet” Malware for Industrial Control System Professionals. [Электронный ресурс]. Tofino Security. Lantzville, BC, Canada. 2010. October 14. URL: http://www.scadahacker.com/library/Documents/ICS_Events/Analysis%20of%20Siemens%20Malware%20Attacks%20v3.1%20(Tofino%20Security).pdf (дата обращения: 30.12.2013).

10. Goldenberg N., Wool A. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems // International Journal of Critical Infrastructure Protection. 2013. Vol. 6. Issue 2. P. 63–75.
11. Rafael Ramos Regis Barbosa, Sadre R., Pras A. Flow whitelisting in SCADA networks // International Journal of Critical Infrastructure Protection. 2013. Vol. 6. Issue 3 – 4. P. 150 – 158.
12. Weaver P. SNORT IDS for SCADA Systems / RedHat 5 Enterprise Installation Guide Featuring SCADA ICCP Signatures. [Электронный ресурс]. URL: http://www.snort.org/assets/114/Snort_RH5_SCADA.pdf (дата обращения: 30.12.2013).
13. Zhu B., Sastry S. SCADA-specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy. [Электронный ресурс]. URL: http://www.cse.psu.edu/~smclaugh/cse598e-f11/papers/zhu.pdf (дата обращения: 30.12.2013).
14. Byres J. Honeywell selects Tofino™ Modbus Read-only Firewall to Secure Critical Safety Systems. [Электронный ресурс]. The University of British Columbia. Canada. 2011. January 6. URL: http://www.tofinosecurity.com/sites/default/files/pr_hon_modbus_read-only_firewall_01_06_11.pdf (дата обращения: 30.12.2013).
15. The Untouchables: Protecting Sensitive Technology Systems with Tenable's Passive Vulnerability Scanner // Tenable Network Security. [Электронный ресурс]. Tenable Network Security, Inc. Columbia, USA. 2012. January 9. URL:http://static.tenable.com/whitepapers/Tenable-TheUntouchables.pdf (дата обращения: 30.12.2013).
16. Zeng W. Secure Distributed Control Methodologies with Built-in Defense in Distributed Networked Control Systems. [Электронный ресурс]. North Carolina State University. Raleigh, North Carolina, USA. 2013. August 16. URL: http://www.lib.ncsu.edu/resolver/1840.16/8911 (дата обращения: 30.12.2013).
17. Васенин В. А. Критическая энергетическая инфраструктура: кибертеррористическая угроза // Информационные технологии. – 2009. – № 9.
18. Кубышкин А. С. Разработка модели разграничения прав доступа для автоматизированных систем технологического управления // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2012. Т. 10, вып. 3.
19. Meixell B., Forner E. Out of Control: Demonstrating SCADA Exploitation // Black Hat 2013. [Электронный ресурс]. Black Hat Conference. Las Vegas, Nevada, USA. 2013. URL: https://media.blackhat.com/us-13/US-13-Forner-Out-of-Control-Demonstrating-SCADA-Slides.pdf (дата обращения: 30.12.2013).

20. Federal Grand Jury Indicts Arlington Security Guard for Hacking into Hospital's Computer System // Official FBI Press Release 2009. [Электронный ресурс]. U.S. Attorney's Office. Dallas, Texas, USA. 2009. July 23. URL: http://www.fbi.gov/dallas/press-releases/2009/dl072309a.htm (дата обращения: 30.12.2013).
21. Byres E. Patching for Control Systems – A Broken Model? [Электронный ресурс]. Tofino Security. Lantzville, BC, Canada. 2013. URL: https://www.tofinosecurity.com/downloads/691 (дата обращения: 30.12.2013).
22. Seymour B., Kabay E. IS Auditing Procedure – Security Assessment – Penetration Testing and Vulnerability Analyses. Document P8 / ISACA. [Электронный ресурс]. ISACA (Information System Audit and Control Association). Rolling Meadows, Illinois, USA. 2004. URL: http://trygstad.rice.iit.edu:8000/Audits/Audit%20Checklists/ISAuditingP8PenetrationTesting-ISACA.pdf (дата обращения: 30.12.2013).
23. Gritsai G., Timorin A., Goltsev Y., Ilin R., Gordeychik S., Karpin A. SCADA Safety In Numbers V1.1*. [Электронный ресурс]. Positive Technologies. Moscow, Russian Federation. 2012. URL: http://www.ptsecurity.com/download/SCADA_analytics_english.pdf (дата обращения: 30.12.2013).
24. Официальный документ Совета Безопасности РФ "Основные направления государственной политики в области обеспечения безопасности автоматизированных систем управления производственными и технологическими процессами критически важных объектов инфраструктуры Российской Федерации" от 08.08.2013 г. [Электронный ресурс]. URL: http://www.scrf.gov.ru/documents/6/113.html (дата обращения: 30.12.2013).
25. Ревнивых А. В., Федотов А. М. Обзор политик информационной безопасности // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2012. Т. 10, вып. 3.
26. Modbus Application Protocol Specification V1.1b. [Электронный ресурс]. Modbus Organization. Hopkinton, Massachusetts, USA. 2006. URL: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf (дата обращения: 30.12.2013).
27. Витяев Е. Е., Ковалерчук Б. Я., Федотов А. М., Барахнин В. Б., Дурдин Д. С., Белов С. Д., Демин А. В. Обнаружение закономерностей и распознавание аномальных событий в потоке данных сетевого трафика // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2008. Т. 6, вып. 2.

28. Безукладников И.И., Кон Е.Л. Проблема скрытых каналов в промышленных информационно-управляющих и инфокоммуникационных сетях // Промышленные АСУ и контроллеры. – 2011. – № 7.
29. Хемди А. Таха. Имитационное моделирование // Введение в исследование операций = Operations Research: An Introduction. – 7-е изд. – М.: Вильямс, 2007. – с. 667-705.
30. Михеева Т. В. Информационные технологии имитационного моделирования в организации корпоративной производственной системы // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2009. Т. 7, вып. 2.
31. Agha G. Actors: A Model of Concurrent Computation in Distributed Systems. Cambridge: MIT Press Series in Artificial Intelligence, 1986.
32. Xiaojun Liu, Jie Liu, Eker J., Lee E. A. Heterogeneous Modeling and Design of Control Systems. [Электронный ресурс]. Department of Electrical Engineering and Computer Sciences University of California. Berkeley, California, USA. 2001. URL: http://sec.eecs.berkeley.edu/papers/01/controlsys/controlsys.pdf (дата обращения: 10.03.2014).
33. Hewitt C., Bishop P., Steiger R. A universal modular ACTOR formalism for artificial intelligence // Proceeding IJCAI'73 Proceedings of the 3rd international joint conference on Artificial intelligence. 1973. P. 235 – 245.
34. Hewitt C. Viewing Control Structures as Patterns of Passing Messages // Journal of Artificial Intelligence. 1977. P. 323 – 364.
35. Hewitt C. Actor Model of Computation: Scalable Robust Information Systems. [Электронный ресурс]. 2014. URL: http://arxiv.org/ftp/arxiv/papers/1008/1008.1459.pdf (дата обращения: 10.03.2014).
36. Byrne P. H. Analysis & Science in Aristotle. // SUNY Series in Ancient Greek Philosophy. State University of New York Press. 1997.
37. Hewitt C., Baker H. Actors and Continuous Functionals. // Massachusetts Institute of Technology. Laboratory for Computer Science. 1977.
38. Cohen E. S. Semantic models for parallel systems. [Электронный ресурс]. Carnegie Mellon University. Computer Science Department. Pittsburgh, Pennsylvania, USA. 1975. URL: http://repository.cmu.edu/compsci/1726 (дата обращения: 10.03.2014).
39. Muliadi L. Discrete event modeling in Ptolemy II. [Электронный ресурс]. Department of EECS University of California. Berkeley, California, USA. 1999. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.2313&rep=rep1&type=pdf (дата обращения: 10.03.2014).

40. Milner R. Processes: A Mathematical Model for Computing Agents. // Studies in Logic and Foundations in Math. 1975. Vol. 80. P. 157–174.
41. Janneck J. W. Actors and their composition. // Formal Aspects of Computing. 2003. Vol. 15. Issue 4. P. 349-369.
42. Modicon Modbus Protocol Reference Guide. [Электронный ресурс]. MODICON, Inc., Industrial Automation Systems. North Andover, Massachusetts, USA. 1996. URL: http://web.eecs.umich.edu/~modbus/documents/PI_MBUS_300.pdf (дата обращения: 10.03.2014).
43. Gerhart J. Home Automation and Wiring. McGraw-Hill Professional. 1999. pp. 322.