УДК 004.056

МЕТОДИКА ОЦЕНКИ НЕОБХОДИМОСТИ ПРИМЕНЕНИЯ ОБНОВЛЕНИЙ БЕЗОПАСНОСТИ В ПРОМЫШЛЕННЫХ СИСТЕМАХ

К. В. Семенков*, В. Г. Промыслов**

*,**Институт проблем управления им. В. А. Трапезникова РАН, г. Москва

*✉ semenkov@ipu.ru, **✉ vp@ipu.ru

Аннотация. Рассматривается проблема применения обновлений кибербезопасности (патчей) для программного обеспечения (ПО) автоматизированных систем управления технологическими процессами (АСУ ТП) с длительным жизненным циклом. Проблема исследуется в рамках этапа эксплуатации системы. Основное внимание уделяется: большому числу уязвимостей, обнаруживаемых в ПО; сложности анализа влияния уязвимости на безопасность системы и функции, выполняемые системой; требованиям к тестированию совместимости обновлений и сертификации ПО после внесения изменений. На основе метода анализа отказов (англ. *Failure Mode and Effects Analysis*, FMEA) предложена методика, позволяющая упростить анализ влияния уязвимости на кибербезопасность. Она предполагает рассмотрение не каждой отдельной уязвимости, а меньшего по мощности множества сценариев атак. При анализе сценариев атаки также учитывается действие мер защиты. Методика включает в себя простые критерии применения обновлений безопасности по результатам анализа. Приведен пример анализа уязвимости по предлагаемой методике.

Ключевые слова: уязвимость, обновление, оценка риска, АСУ ТП, кибербезопасность, критерий.

ВВЕДЕНИЕ

Одним из основных путей проведения кибератак является использование уязвимостей в программном обеспечении (ПО). Несмотря на развитие технологий разработки и тестирования ПО, сложность программ не позволяет гарантировать отсутствие уязвимостей в программном обеспечении. Злоумышленники концентрируют свои ресурсы на поиске и использовании уязвимостей, а разработчики и пользователи ПО заинтересованы в поиске, оперативном устранении этих уязвимостей, выпуске и установке соответствующих обновлений безопасности ПО (патчей). В целом под патчем часто понимается весьма широкий спектр изменений ПО [1–4], которые либо характеризуются подходом к патчу как к процессу или объекту, либо связаны с объемом или характером изменений ПО. В неформальном общении специалистов также можно встретить другие схожие термины, например, update, bugfix, hotfix. Для целей настоящей работы примем следующее

Определение. Обновление безопасности (патч) – это модификация установленного программного обеспечения, призванная устранить уязвимости программного обеспечения без изменения других функциональных характеристик программного обеспечения. ♦

В части накопления информации об уязвимостях в мире проделана огромная работа, имеются общедоступные и постоянно пополняемые базы: американская база CVE [5], российский банк данных угроз безопасности информации БДУ ФСТЭК [6], французская CERT-FR [7] и др. В базах для каждой уязвимости из списка обязательно содержатся ее описание, оценка влияния и рекомендации по устранению уязвимости или ослаблению ее негативного влияния. Производители программного обеспечения выпускают обновления безопасности, и установка обновлений зачастую может выполняться в автоматическом режиме.

Наработанный в мире опыт систематизирован в международных и национальных стандартах и методических документах по применению патчей

(см., например, [3, 8, 9]). Указания и рекомендации этих документов в целом сводятся к следующим шагам:

1) постоянно вести мониторинг уязвимостей в применяемом ПО;

2) выполнять анализ вновь открытых уязвимостей и оценку риска для кибербезопасности;

3) в зависимости от результатов оценки риска определить дальнейшие действия: принять риск, устранить риск – применить патч и т. д.;

4) в случае положительного решения по установке обновления:

а) разработать план применения обновления;

б) проверить целостность и достоверность обновления;

в) протестировать обновление;

г) установить обновление;

д) проверить состояние ПО и конфигурацию после установки обновления.

Однако вопросы практического применения имеющихся данных, прежде всего связанные с большим объемом анализируемой информации и ее достоверностью, по мнению авторов, пока до конца не решены.

В данной работе на примере автоматизированной системы управления технологическими процессами (АСУ ТП) будет рассмотрена проблема применения обновлений безопасности [100, 11] в системах с длительным жизненным циклом и предложен новый метод ее решения. Под системой с длительным жизненным циклом в работе понимается такая система, стадия эксплуатации и поддержки которой продолжается в течение нескольких лет или десятилетий.

Согласно указаниям нормативных и методических документов [10–13] идентификация, анализ и оценка уязвимостей должны проводиться в течение всего жизненного цикла защищаемой системы. Так как ресурсы для обеспечения кибербезопасности и контекст, в котором рассматривается система, существенно отличаются для различных этапов жизненного цикла, то различаются и проблемы устранения уязвимостей, и методы их решения. Данная работа посвящена вопросам оценки уязвимостей на этапе эксплуатации в предположении, что на момент окончания этапа разработки разработчик закрыл известные ему уязвимости и применил адекватные меры защиты для снижения риска до приемлемого уровня. Для этапа разработки имеется разнообразный набор рекомендаций по разработке безопасного ПО [14].

Внимание к этапу эксплуатации определяется тем, что, как показывает опыт авторов, проблема управления уязвимостями наиболее полно проявляется на этапе эксплуатации и усложняется со временем. Это связано прежде всего с интегральным эффектом нескольких факторов: накопления обнаруженных уязвимостей в используемых компонентах, окончания срока поддержки некоторых их них разработчиком, устаревания технологий обеспечения информационной защиты, заложенных в проекте системы.

В качестве объекта исследования выбраны сложные программные системы, представляющие собой комплексы программ [15], в которых используются как специально разработанные для системы компоненты, так и сторонние компоненты общего применения. Предположим также, что число компонентов в системе достаточно велико, так как для простых систем проблема патчей представляется не слишком серьезной, потому что количество уязвимостей в ней вряд ли будет большим, и они могут оперативно отслеживаться и устраняться в процессе эксплуатации. Практика показывает, что для АСУ ТП «простыми» являются системы, не выходящие за рамки одного компьютера; тогда число активов и связей между ними позволяет описать возникающие отношения безопасности дискреционной моделью, которую можно использовать при оценке риска для активов, связанных с обнаруживаемыми уязвимостями.

Проблемы управления рисками для промышленных объектов и установка обновления для цифровых систем безопасности имеют большое научное и практическое значение. Обзор проблемы оценки риска для АСУ ТП АЭС приведен в работе [16], а подробный обзор публикаций по управлению обновлениями безопасности, вышедших в 2002–2020 гг., приведен в работе [17]. Новизна настоящей работы заключается в том, что в ней детализируются методики верхнего уровня (как, например, [1, 10]), предлагается конкретный, ориентированный на промышленные системы управления набор действий для принятия решений об установке обновлений.

Проблема управления обновлениями безопасности будет рассмотрена в разрезе функций, выполняемых системой, а не в разрезе характера уязвимости отдельного компонента, для которого имеется патч. Так, основная функция АСУ ТП – это управление промышленным объектом. Тогда цель установки патча для операционной системы (ОС) компьютера в составе АСУ ТП – это не защита ОС как таковой, а снижение риска, связанного с невозможностью управлять объектом в случае использования уязвимости. Для решения предложен риск-ориентированный метод на основе метода анализа отказов и их последствий (англ. *Failure Mode and Effects Analysis*, FMEA) [18], который

дает критерии применения обновлений. В дополнение к управлению уязвимостями на основе их типов предлагается явно учитывать влияние мер защиты на реализуемость атак нарушителя, обладающего определенными возможностями. Применение описанного в работе подхода позволяет сопоставить вновь открываемые уязвимости с известными типами уязвимостей некоторого классификатора (см., например, общий перечень недостатков безопасности ПО (англ. *Common Weakness Enumeration*, CWE) [19]), и ответить на вопрос, имеется ли у системы «иммунитет» в виде барьера из совокупности мер защиты от новой уязвимости. Под барьером здесь и далее будем понимать определенный набор мер защиты, который обеспечивает гарантированную защиту от определенного сценария атак.

1. ОСОБЕННОСТИ ПРИМЕНЕНИЯ МОДЕЛИ УГРОЗ ПРИ УСТАНОВКЕ ПАТЧЕЙ

Большинство риск-ориентированных подходов, применяемых в задаче управления патчами, базируется на подходах к оценке риска, сформулированных в стандарте ИСО 27005 [20]. В соответствии с ними анализ модели угроз, включающей уязвимости, угрозы и нарушителя, в основном влияет на решение о приемлемости или неприемлемости риска и, следовательно, на решение об установке патча. Существует много методов описания элементов модели угроз и составления их таксономии; ниже будут рассмотрены те из них, которые авторы считают наиболее подходящими для оценки риска в сложных промышленных системах на этапе эксплуатации. Для этого сначала рассмотрим отдельные составляющие модели угроз.

1.1. Анализ уязвимостей

Будем придерживаться определения уязвимости из стандартов ИСО/МЭК 27000 и методических документов ФСТЭК [12]: уязвимость – это «… слабость актива (программы), которая может быть использована угрозой».

Уязвимости могут иметь различную природу. Они могут быть связаны со свойствами системы, заложенными при разработке (недостатки в архитектуре глубоко эшелонированной защиты или междоменного взаимодействия, ошибки реализации) или же могут появиться из-за неправильного применения мер защиты (например, из-за неправильного применения мер парольной защиты). Анализ уязвимостей призван установить, насколько уязвимости могут повлиять на защищенность системы и на оценку доверия к реализуемым мерам защиты [21]. Патчи должны применяться к системе в случае, если анализ применимости уязвимостей выявит недопустимый уровень риска информационной безопасности для системы (см. руководство [9], *рис. 3.1*). Попробуем показать, какие проблемы возникают при анализе уязвимостей, для чего рассмотрим использование нормативных методических рекомендаций по анализу и применению патчей более подробно.

Первая проблема, которую необходимо выделить, – это масштаб рассматриваемой системы. Как было указано выше, сложная программная система включает большое количество разнородных компонентов и сторонних приложений, и для обеспечения кибербезопасности необходимо вести мониторинг большого числа уязвимостей, связанных как с компонентами, разработанными специально для системы, так и со сторонними продуктами (например, уязвимостей операционной системы, систем управления базами данных, веб-серверов, интерпретаторов и т. п.).

Количество вновь открываемых уязвимостей с каждым годом увеличивается. Для примера приведем количество новых уязвимостей, внесенных в базы CVE и ФСТЭК в 2021–2023 гг. (табл. 1).

Таблица 1

Количество уязвимостей, добавленное в CVE и БДУ ФСТЭК, тыс.

База данных	2021 г.	2022 г.	2023 г.
БДУ ФСТЕК	6,4	7,5	9,1
CVE [22]	20,2	25,0	29,0

Для сложной системы поток уязвимостей может составить десятки и сотни уязвимостей в сутки, и даже первичный анализ новых уязвимостей может потребовать от организации существенных ресурсов и затрат.

Следующая проблема, на которую необходимо обратить внимание, состоит в том, что описание уязвимости во всех базах дается в относительно свободной форме, какого-то единого стандарта описаний не существует. Описания могут быть как весьма краткими, так и излишне детальными, что существенно усложняет их анализ. Приведем несколько примеров неудачных описаний (табл. 2). Описание уязвимости CVE-2018-19932 содержит много технических подробностей, которые могут заинтересовать лишь разработчиков данного ПО, описание уязвимости CVE-2023-36762 слишком общее, а описание уязвимости CVE-2021-30618 не содержит почти никакой информации.

Таблица 2

Примеры неудачных описаний уязвимостей

Уязвимость	Описание
CVE-2018-19932	Проблема была обнаружена в библиотеке дескрипторов двоичных файлов (BFD) (также известной как libbfd), распространявшейся в GNU Binutils до версии 2.31. Макрос IS_CONTAINED_BY_LMA в elf.c вызывает целочисленное переполнение и бесконечный цикл. (англ. An issue was discovered in the Binary File Descriptor (BFD) library (aka libbfd), as distributed in GNU Binutils through 2.31. There is an integer overflow and infinite loop caused by the IS_CONTAINED_BY_LMA macro in elf.c.)
CVE-2023-36762	Уязвимость удаленного выполнения кода Microsoft Word. (англ. Microsoft Word remote code execution vulnerability.)
CVE-2021-30618	Неподходящая реализация в DevTools. (англ. Inappropriate implementation in DevTools.)

Имеются работы, направленные на автоматизацию анализа описания уязвимостей (см., например, [23, 24]), в том числе с использованием алгоритмов машинного обучения. Однако пригодные к практическому применению инструменты, полностью автоматизирующие анализ реальных систем, авторам неизвестны.

Проблема описания усугубляется наличием языкового барьера, имеющего комплексный характер. Во-первых, описания большинства уязвимостей в зарубежных базах составляются на английском языке, которым хорошо владеют далеко не все авторы описания уязвимостей. То есть уже на этапе первоначального описания уязвимости ее суть может быть искажена или неполно изложена. Во-вторых, большой объем информации об уязвимостях переносится из международных открытых баз данных в национальные, где описания обычно модерируются и переводятся на национальный язык. Таким образом, в национальных базах уязвимостей при переводе на официальные языки стран могут как добавиться ошибки и неточности, затрудняющие анализ уязвимости, так и наоборот, описания могут расширяться и уточняться. Однако в последнем случае часто теряется обратная связь с разработчиком компонента, в котором открыта уязвимость, так как сделанные уточнения доступны только на национальном языке.

Можно заключить, что для самостоятельного мониторинга и анализа такого объема уязвимостей организации необходим достаточно большой штат экспертов, которые могут квалифицированно выполнить анализ этой слабоструктурированной информации и оценить риск для защищаемой системы, ассоциированный с уязвимостью.

Возможный способ уменьшить объем анализируемой информации – это приоритизация уязвимостей, которая позволяет сконцентрировать усилия на наиболее критичных из них. Одной из самых известных и распространенных шкал оценки критичности является CVSS (*Common Vulnerability Scoring System*) [25–27]. В ней оценка ведется по трем группам: базовых, временны́х и контекстных метрик. Далее будет использована только базовая метрика, так как две последние, по опыту авторов, или не содержат информации, или информация является специфичной для конкретного сценария применения программы. К базовым метрикам CVSS 3.0 [26] относятся:

- вектор атаки, например, сетевая, локальная, физический доступ, атака на смежный сетевой протокол;
- сложность атаки;
- необходимые права доступа для эксплуатации уязвимости;
- участие «обычного» пользователя для эксплуатации уязвимости;
- возможность выхода последствий атаки за пределы исследуемой системы;
- влияние на свойства доступности, конфиденциальности, целостности информационных ресурсов, контролируемых программой, в которой обнаружена уязвимость.

Базовые метрики главным образом отражают свойства самого программного компонента, факторы среды функционирования учитываются весьма условно. Их значения, рассчитанные экспертным способом, приводятся в базах уязвимостей.

Из описания базовых метрик видно, что они главным образом предназначены для разработчиков и пользователей отдельного программного компонента и что роль компонента в информационной системе в базовых метриках не учитывается. Чтобы провести полный анализ уязвимости по системе CVSS для дальнейшей оценки риска, нужно рассчитывать оставшиеся группы метрик или использовать другие метрики и базы данных уязви-

мостей, учитывающие данные аспекты [2, 28]. Может также возникнуть необходимость пересчета значений базовых метрик с учетом особенностей организации или исследуемой системы.

Переход от отдельных уязвимостей к их классам мог бы снизить трудозатраты на анализ и оценку риска. Классификация уязвимостей может быть проведена по-разному в зависимости от предметной области и уровня детальности системы, но в целом выделяют следующие типы уязвимостей, отражающие природу активов [13]:

– аппаратные уязвимости,

– уязвимости программного обеспечения,

– сетевые уязвимости.

В работе будут рассматриваться только уязвимости программного обеспечения, потому что обновления безопасности в основном сосредоточены на них.

Для перехода от отдельных реализаций уязвимостей, которые в большинстве случаев уникальны, к типам применим подход [14], связывающий уязвимости с недостатками программы. В качестве классификатора недостатков по типам возьмем один из наиболее известных – CWE [19].

Этот перечень имеет вид иерархической свободно дополняемой таксономии дефектов программного и аппаратного обеспечения, которая может использоваться в инструментах анализа безопасности.

Классификатор представляет собой древовидную многоуровневую структуру, в которой недостатки распределены по четырем уровням: корневой, базовый, уровень класса, уровень вариантов. Для облегчения работы в CWE выделены так называемые представления – набор записей CWE, предназначенный для решения отдельных типов задач (например, разработки ПО, разработки оборудования, исследований).

Отнесение уязвимостей в CWE производится экспертами вручную, и как показывает практика, во многих случаях мнения экспертов расходятся [29]. Ведутся также исследования относительно полностью автоматической классификации, но однозначных результатов пока нет (см., например, работу [23]).

Каталог CWE можно использовать для создания системы безопасной разработки, но, по мнению авторов, для классификации уязвимости в целях создания системы защиты он малопригоден, так как связь класса CWE с методами атак (например, атаками из классификатора CAPEC [30]) и моделью нарушителя неоднозначна, потому что один и тот же недостаток ПО может использоваться в различных сценариях атак разными нарушителями и приводить к разным последствиям.

1.2. Анализ угроз и характеристик нарушителей

Угроза в рамках кибербезопасности может быть определена как совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации [12].

Виды угрозы обычно соотносятся со способом использования уязвимостей, каждый из которых имеет разные последствия и предпосылки; как правило, виды угрозы можно сопоставить с нарушением свойств кибербезопасности в одной из эталонных моделей (например, в эталонной модели КЦД – конфиденциальность, целостность, доступность).

Отображение свойств кибербезопасности на свойства самой системы индивидуально для каждой конкретной системы, и в зависимости от анализируемой системы каждый вид угрозы может повлиять на любое из свойств системы, как то: надежность, доступность, ремонтопригодность, безопасность. Есть несколько распространенных классификаторов видов угроз:

– банк данных угроз безопасности информации ФСТЭК России [6];

– MITRE ATT&CK [31];

– классификатор угроз STRIDE компании Microsoft [32].

Нарушитель (агент угрозы, злоумышленник) – это активный элемент, субъект в системе, который пытается использовать уязвимость. Примерами возможных нарушителей являются хакеры, компьютерные преступники, террористы, промышленные шпионы, инсайдеры [33]. Подробная классификация злоумышленников содержится в каталогах ФСТЭК, CAPEC, MITRE ATT&CK и др. [12, 30, 31]. Каждый из классификаторов содержит набор атрибутов, по которым могут быть распределены нарушители, например, по типу нарушителей, уровню их компетентности и оснащенности, целям атаки. Нарушители в рамках модели ФСТЭК дополнительно подразделяются на внешних и внутренних. Подходы ИСО/МЭК 27000 и MITRE ATT&CK атрибутируют нарушителя по целям атаки: получение денег, подрыв репутации, получение конкурентного преимущества и т. п.

В качестве примера атрибутирования по документам ФСТЭК приведем выделенные в них типы нарушителей и уровни их компетенции.

Типы нарушителей по ФСТЭК:

- специальные службы иностранных государств;
- террористические, экстремистские группировки;

• преступные группы (криминальные структуры);

• отдельные физические лица (хакеры);

• конкурирующие организации:

• разработчики программных, программно-аппаратных средств;

• лица, обеспечивающие поставку программных, программно-аппаратных средств, обеспечивающих систем;

• поставщики услуг связи, вычислительных услуг;

• лица, привлекаемые для установки, настройки, испытаний, пусконаладочных и иных видов работ;

• лица, обеспечивающие функционирование систем и сетей или обеспечивающих систем оператора (администрация, охрана, уборщики и др.);

• авторизованные пользователи систем и сетей.

Уровни компетентности нарушителя по ФСТЭК:

• базовые возможности по реализации угроз безопасности информации (Н1);

• повышенные возможности по реализации угроз безопасности информации (Н2);

• средние возможности по реализации угроз безопасности информации (Н3);

• высокие возможности по реализации угроз безопасности информации (Н4).

Указанный перечень видов нарушителей может быть дополнен иными нарушителями с учетом особенностей области деятельности, в которой функционируют системы, и связи анализируемой системы с внешним миром.

1.3. Выводы по анализу составляющих модели угроз

Из вышесказанного становится ясно, что работа по анализу отдельных компонентов модели угроз и оценке риска от использования уязвимостей злоумышленниками требует привлечения на постоянной основе специалистов, владеющих широким диапазоном знаний и навыков программирования, оценки риска информационной безопасности, а также обладающих экспертными знаниями как на системном уровне, так и на уровне отдельных компонентов.

Эта работа весьма трудоемка и не ограничивается функциями, связанными с анализом риска для уязвимости, ведь если выявлена необходимость устранения уязвимости с применением патча, то это приводит к дополнительным работам и проблемам для собственника системы.

2. ПРОБЛЕМЫ УПРАВЛЕНИЯ ПАТЧАМИ

В § 1 были приведены основные шаги, предшествующие решению об установке патча, и связанные с ними проблемы. Однако трудности этим не исчерпываются, и в случае положительного решения об установке патча собственник системы сталкивается с целым рядом новых проблем, обусловленных сложностью рассматриваемого класса систем:

• Для компонентов в составе системы обновление безопасности программного компонента часто не выпускается отдельно, а включается в состав новой версии этого компонента, тогда функциональность новой версии компонента может потребовать дополнительного тестирования компонента в составе системы. Замена существующей версии программного компонента может привести к отказу некоторых функций АСУ ТП и к необходимости доработки ПО АСУ ТП.

• Программные компоненты в составе комплекса программ связаны друг с другом цепочкой зависимостей. Зависимость может иметь как горизонтальный характер, например, на уровне прикладных компонентов программы, так и вертикальный – на уровне компонентов операционной системы. Замена любого из ключевых компонентов может повлечь за собой замену остальных и в худшем случае – замену всех компонентов в цепочке зависимостей. Для системы с длительным жизненным циклом может возникнуть такая ситуация, что часть компонентов, входящих в состав цепочки зависимостей, более не поддерживается разработчиком, для них не существует новых версий, и обновление в этом случае нельзя провести простым переходом на новую версию.

• Для АСУ ТП характерны длительный жизненный цикл (период эксплуатации может составлять десятки лет) и жесткие требования к порядку их разработки и тестирования. Учитывая высокую скорость обнаружения новых уязвимостей, можно предположить, что за время между выпуском версии ПО АСУ ТП и запуском системы в эксплуатацию после пусконаладки в программном окружении АСУ ТП найдут новые уязвимости. Необходимо также учитывать, что производители сторонних компонентов могут прекратить поддержку устаревших версий своих продуктов, и для новых уязвимостей будут отсутствовать обновления безопасности.

• Программное обеспечение АСУ ТП тестируется и сертифицируется для работы на определен-

ных технических средствах в заданном программном окружении, и применение патча в зависимости от условий действия сертификата на систему может привести к необходимости дорогостоящей и длительной процедуры повторной сертификации системы.

Немаловажным фактором является вопрос доверия к источнику обновлений и к разработчикам программного обеспечения. Как показывает практика, риск того, что полученное обновление может содержать намеренно встроенные уязвимости, существует [2, 34, 35]. Причем доверие к источнику обновлений может меняться с течением времени от широкого применения программ до их запрета только на основании оценок риска, связанного с социальной и политической ситуацией [36, 37].

Коснемся вопроса тестирования обновлений. Установка и тестирование обновлений могут потребовать испытаний работы всей АСУ ТП. Одним из решений, позволяющих в большинстве случаев провести полноценное тестирование, сравнимое с тестированием на реальном объекте, является использование гибридного цифрового двойника, в котором совместно с чисто цифровыми компонентами используются элементы реального оборудования АСУ ТП [38].

Таким образом, практика последовательного анализа отдельных уязвимостей в компонентах и установки патчей этих компонентов может использоваться только на простых системах. Если же речь идет о больших, сложных системах с длительным периодом эксплуатации, то необходимо искать другие решения проблемы обеспечения кибербезопасности.

Отметим, что в обзоре публикаций [17] составлен перечень описанных в литературе проблем, и, по мнению авторов, наиболее важными являются следующие из них:

– задачи применения патчей для обеспечения информационной безопасности и текущие задачи организации (например, непрерывности бизнеса или технологических процессов) могут быть несовместимы ([17], *п. 2 табл. 5*);

– процесс применения патчей требует дополнительных ресурсов и экспертных знаний, которых нет в компаниях, эксплуатирующих ПО ([17], *п. 5, п. 6 табл. 5*);

– автоматическое тестирование патчей крайне затруднительно, большинство патчей тестируется вручную и качество тестирования зачастую неудовлетворительно ([17], *п. 11, п. 12 табл. 5*);

– проверка корректности применения патча и устранения последствий ошибок развертывания – довольно трудная задача ([17], *п. 13, п. 14 табл. 5*).

Ниже опишем подход, который позволяет, оставаясь в рамках риск-ориентированного подхода, существенно упростить анализ уязвимостей.

3. РИСК-ОРИЕНТИРОВАННЫЙ ПОДХОД К АНАЛИЗУ УЯЗВИМОСТЕЙ С УЧЕТОМ МЕР ЗАЩИТЫ

Предлагаемый подход разработан с учетом его применимости к системам, обладающим следующими характеристиками:

– программная среда системы рассматривается как набор отдельных компонентов, представленных в виде черного ящика и взаимодействующих друг с другом посредством известного набора интерфейсов;

– система создавалась с применением архитектурных принципов инкапсуляции и разделения на домены [39].

Будем рассматривать систему с точки зрения выполнения определенных функций и учитывать вклад каждого компонента в функцию системы.

Основная идея предлагаемого подхода заключается в том, что анализ уязвимостей ПО, составляющего систему, должен дать ответ на вопрос, насколько уязвимость опасна для той или иной функции системы, а не насколько она критична для того или иного программного компонента.

Предлагаемый в данной работе метод анализа уязвимостей основывается на методе анализа видов и последствий отказов (FMEA), который предназначен для выявления слабых мест в архитектуре системы, чтобы повысить таким образом ее надежность и безопасность [18]. Метод FMEA был разработан в 1940-х гг. [40] и первоначально был нацелен исключительно на анализ надежности или безопасности технических систем и оборудования. Позднее были предложены модификации метода для анализа задач кибербезопасности систем с цифровыми компонентами (SFMEA) [41, 42]. В настоящей работе приводятся общие сведения о методе FMEA, подробности читатель может узнать из обширной специальной литературы. В рамках FMEA система должна обладать следующими свойствами:

– иметь определенные цели, представленные в виде требований к ее функциям;

– иметь установленные условия функционирования;

– иметь определенные границы;

– иметь иерархическую структуру.

Вместе с блок-схемой иерархической структуры элементов в FMEA используются блок-схемы, отражающие иерархию выполняемых элементами

системы функций и функциональные взаимосвязи между элементами, что обеспечивает прослеживаемость функциональных отказов системы.

Для каждого компонента системы анализируются (рис. 1):

– причины, которые могут привести данному отказу;

– функция, отказ, который может произойти (вид отказа);

– особенности последствий в случае отказа;

– тяжесть отказа (является ли отказ безвредным или наносит ущерб);

– критичность отказа (как и когда отказ можно обнаружить).

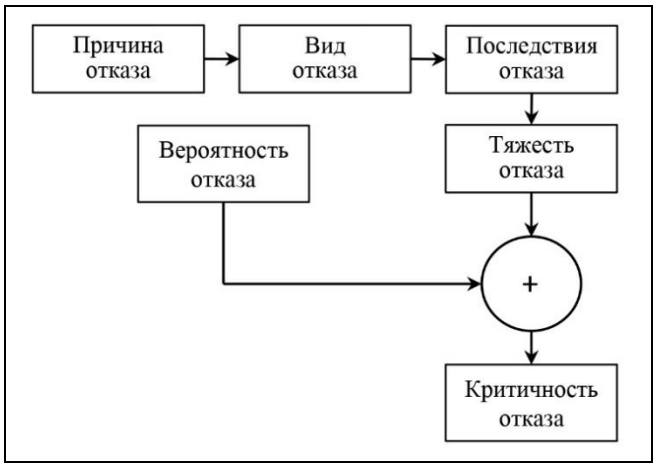

Рис. 1. Одна из возможных последовательностей шагов при оценке критичности отказа согласно стандарту [18]

Применяя подходы FMEA для оценки кибербезопасности, необходимо описать этапы FMEA в терминах кибербезопасности и связать причину отказа с уязвимостью и способностью нарушителя ее использовать.

Предлагается адаптировать методику FMEA для оценки влияния уязвимости и уменьшить таким образом объем анализируемой информации (рис. 2).

Методика учитывает, что уязвимость сама по себе не является причиной отказа: для того, чтобы уязвимость привела к отказу, необходимо, чтобы нарушитель, обладающей достаточной компетенцией, использовал ее для реализации определенной угрозы (см. блок 1 на рис. 2). Нарушитель и его компетенция могут быть типизированы в соответствии с принятой шкалой: например, с использованием классификатора ФСТЭК [12]. Таким образом, отказ – это результат успешной атаки на систему со стороны нарушителя, обладающего достаточной компетенцией и возможностями для эксплуатации уязвимости.

В предлагаемой методике для уменьшения пространства анализа постулируется независимость уязвимостей, что аналогично предположению о независимости отказов в методике FMEA ([18], *п. 4.1*).

Предлагаемая схема анализа, в отличие от типичного рассмотрения отдельных технологий, лежащих в основе уязвимости (как, например, в каталоге CWE), акцентируется на атрибутах CVSS для соотнесения уязвимости и ассоциированной с ней угрозы. Дополнительно выполняется переход от анализа отдельных угроз и нарушителей к анализу типовых сценариев атак, связанных с классами угроз и типовыми возможностями нарушителя, что снижает количество анализируемых угроз и сценариев атак. Примеры типовых сценариев атак для АСУ ТП АЭС описаны в работе [43].

Отказ функции в этом случае связан не с конкретной уязвимостью (см. рис. 2), а с совокупностью условий и факторов, которые привели к отказу компонента и преодолению нарушителем барьера защиты. Необходимо отметить, что отказ может быть непосредственно не связан с (кибер)атакой и с использованием уязвимости, а, например, являться результатом исчерпания ресурса. Однако режимы работы АСУ ТП критических объектов проектируются таким образом, чтобы исключить исчерпание ресурсов.

Вторым важным дополнением к применению подходов FMEA к задачам кибербезопасности является учет наличия барьера, который отражает действие уже реализованного набора мер защиты. Барьер может блокировать влияние отказа компонента на функции всей системы и, следовательно, свести к нулю риски, связанные с атакой на систему. Предполагается, что барьер обладает высокой степенью доверия и способен противодействовать эксплуатации одного или нескольких типов уязвимостей. Таким образом, при анализе сценариев атаки барьер считается абсолютным. Это модельное предположение позволяет существенно уменьшить объем анализа.

Наличие нарушителя, у которого есть мотивация и цель, означает, что отказ в кибербезопасности в общем случае не является случайным событием. Поэтому применение статистических подходов для анализа отказов, как правило, невозможно,

но возможно применение риск-ориентированного подхода и логических правил.

Если допускается вероятностный характер воздействия нарушителя на систему, то влияние отказа на всю систему и связанные с этим риски (см. блок 2 рис. 2) оцениваются с применением методов анализа функциональной зависимости FMEA и разработанных теоретико-вероятностных подходов [18, 44].

Описанный подход можно свести в метод интегрированной защиты и оценки уязвимостей, названный авторами Vulnerability Inspection Control Strategy (VICS), схематически он приведен на рис. 3.

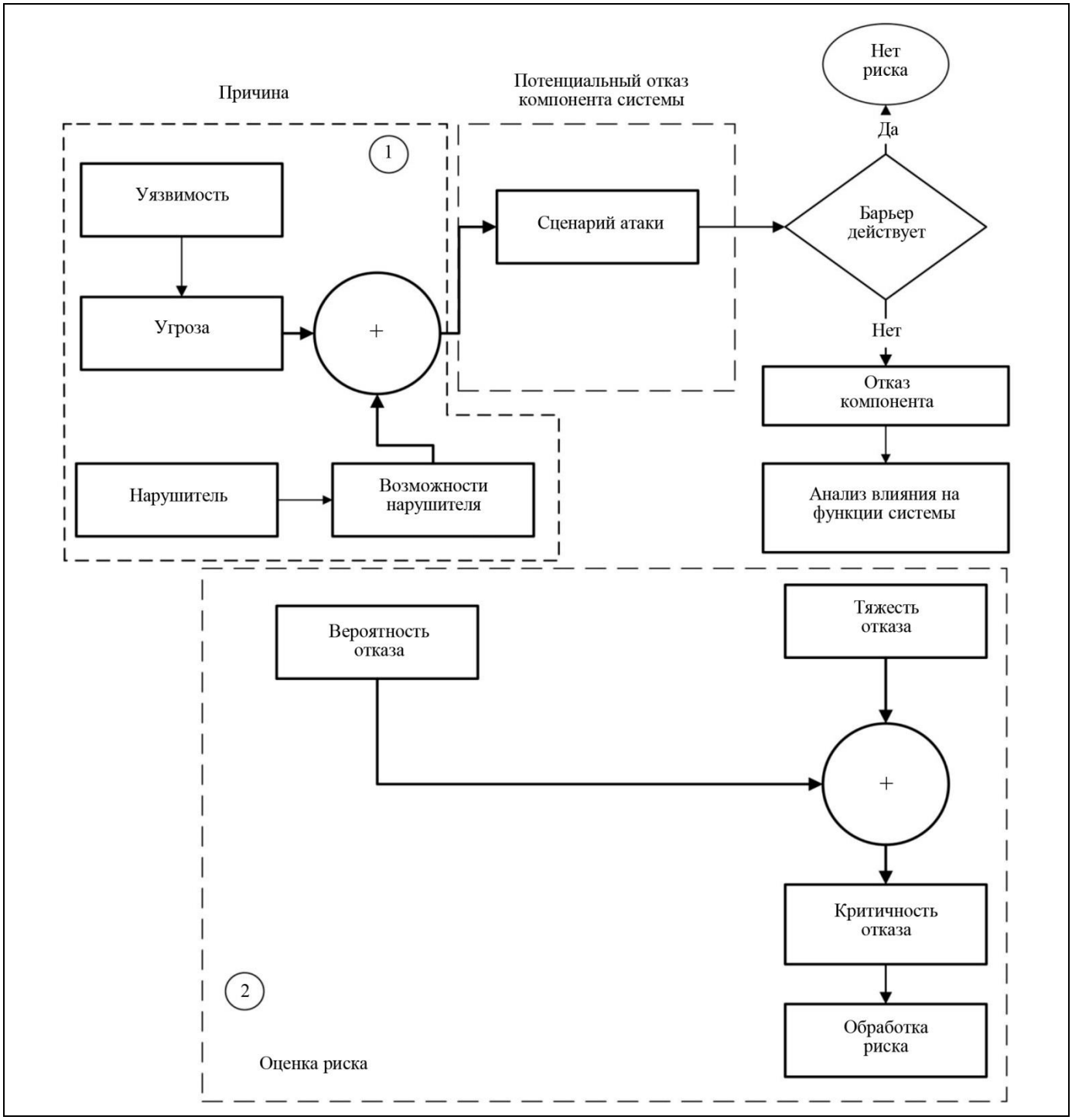

Рис. 2. Комбинированная последовательность шагов при оценке критичности отказа с учетом барьеров защит для задач надежности и кибербезопасности

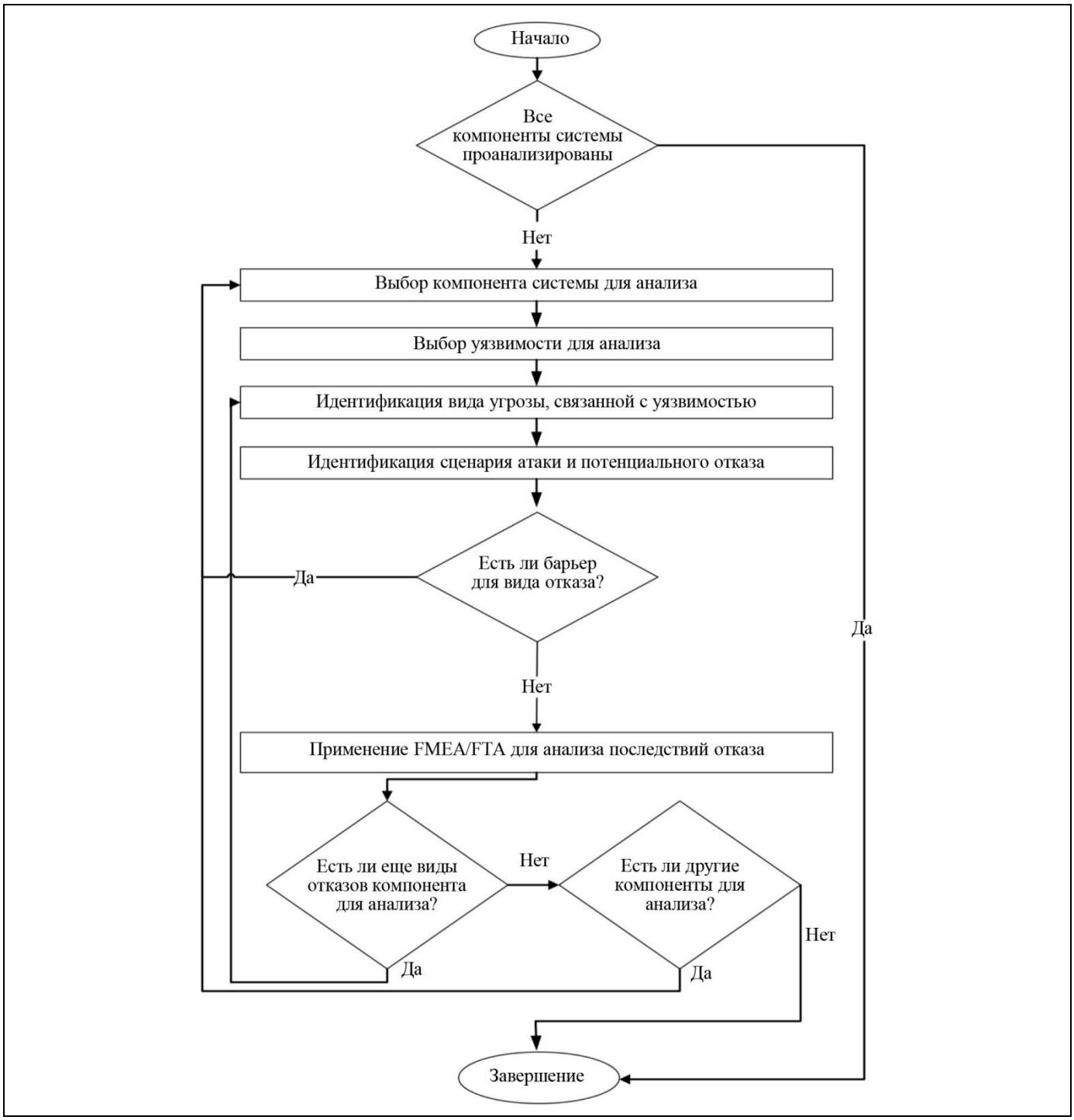

Рис. 3. Комбинированная блок-схема анализа методом интегрированной защиты и оценки уязвимостей VICS

Рассмотрим последовательность действий при анализе уязвимостей и необходимости применения патчей для современной АСУ ТП [45] в рамках VICS. Исходными данными для анализа являются:

– список анализируемых уязвимостей;

– принятая система классификации угроз, действующих на систему (например, модель КЦД или классификатор ФСТЭК);

– принятая модель нарушителя;

– принятые типовые сценарии атак, позволяющие определить для пары «класс угрозы – компетенция нарушителя» потенциальный вид отказа компонента;

– список функций системы, в котором для каждой из функций определены негативные последствия ее нарушения;

– таблица соответствия «отказ функции – барьер», описывающая биграф, в котором для каждого вида отказа определяется барьер в соответствии с реализуемыми мерами защиты;

– структурная схема системы (с ее помощью можно связать уязвимость с одним или несколькими компонентами системы);

– дерево отказов (англ. *Fault Tree Analysis*, FTA), позволяющее проследить влияние отказа скомпрометированных компонентов на функции системы и сгруппировать их по типам.

Анализ сводится к выполнению следующих действий:

1. Для всех уязвимостей определенным способом сопоставить уязвимость с классом угрозы, типовым сценарием атаки, видом отказа компонента с учетом квалификации и мотивации нарушителя.

2. Проанализировать наличие барьера защиты. Если барьер присутствует, то можно прекратить анализ для данной уязвимости.

3. Если барьер отсутствует или недостаточен для данного типа атаки, то выбрать защитную меру или применить патч, которые нейтрализуют выявленные угрозы безопасности или сводят риск их использования к приемлемому, руководствуясь принятой методикой управления патчами (см., например, руководство [9]). Если принято решение реализовать новую меру защиты, то необходимо внести данный сценарий атаки и барьер, ассоциированный с ним, в таблицу, описывающую биграф.

Таким образом, критериями применения обновления безопасности в качестве меры снижения риска, связанного с уязвимостью, являются:

– отсутствие барьера против данного типа атаки,

– недостаточность барьера против данного типа атаки,

– отсутствие иных более эффективных, чем установка обновления, мер защиты.

Предлагается следующая последовательность действий с уязвимостями информационных систем.

1. При разработке системы защиты проводится анализ угроз и создается каталог барьеров защиты и каталог типовых сценариев атак.

2. На основе двух каталогов формируется биграф, описывающий парирование атак барьерами защиты.

3. Каждая новая уязвимость классифицируется в соответствии с классами угроз, нарушителя и сценариями атаки (см. § 4). Если уязвимость попадает в класс, для которого уже установлен барьер в виде набора мер защиты, то установка патча не требуется.

4. В случае если уязвимость приводит к новому сценарию атаки, то необходимо либо установить патч, либо принять другие меры понижения риска. Они, в частности, могут включать введение новых барьеров защиты, сводящих к минимуму последствия данного сценария атаки.

5. Периодически проводится анализ корректности мер защиты, формирующих барьеры. В случае если защитные меры по результатам анализа будут сочтены недостаточными, должны быть разработаны и внедрены другие или дополнительные (включая установку патчей).

Отметим, что стратегия VICS не покрывает уязвимости в программных барьерах защиты и корректность самой среды функционирования. Однако, так как рассматриваемые системы обычно строятся с учетом высокого доверия к корректности реализованных мер защиты и качеству эксплуатации системы, то число уязвимостей в барьерах, как правило, много меньше количества всех уязвимостей системы, и, следовательно, предлагаемый метод покрывает подавляющее большинство уязвимостей. Для управления уязвимостями в барьерах защиты можно использовать существующие нормативные методики (см., например, [3, 8, 9]).

Далее на примере построения системы защиты для системы верхнего уровня АСУ ТП будет продемонстрирована применимость метода VICS на практике.

4. ПРИМЕНЕНИЕ МЕТОДА НА ПРАКТИКЕ

Принципиальную применимость метода VICS продемонстрируем на примере системы верхнего уровня (СВУ) АСУ ТП. Вкратце опишем ее основные функции и свойства (с более подробным рассмотрением систем такого класса можно ознакомиться, например, в работе [45]).

Система верхнего уровня АСУ ТП предназначена:

– для реализации информационных, управляющих и вспомогательных функций;

– для передачи команд оператора по управлению технологическими процессами и технологическим оборудованием;

– для контроля состояния АСУ ТП;

– для интеграции информации от систем АСУ ТП.

Рассматриваемая СВУ АСУ ТП состоит из серверов, которые занимаются сбором и архивированием информации от смежных систем и команд оператора, и рабочих станций, где отображается

информация о состоянии АСУ ТП и вводятся команды управления. Все элементы СВУ резервированы и связаны резервированной сетью. Доступа в Интернет АСУ ТП не имеет. Допускается передача однонаправленного потока данных из АСУ ТП наружу (например, через диод данных).

В качестве модели угроз примем модель угроз КЦД, с которой свяжем основные типы угрозы нарушения конфиденциальности, целостности и доступности. Такой подход, несомненно, является упрощением, но он следует практике, отраженной во многих широко используемых классификаторах безопасности (например, CWE, CVSS). В качестве модели нарушителя примем модель нарушителя с низким или средним уровнем привилегий (права администратора у такого нарушителя отсутствуют) и средними возможностями по реализации угроз безопасности информации (уровень компетентности нарушителя по ФСТЭК Н3). Предположим, что с помощью локальной атаки такой нарушитель хочет нарушить целостность программного обеспечения или данных (сюда входит неавторизованный запуск команд) и выполнить таким способом команду управления, которая приведет к физическому ущербу для объекта управления.

Предположим, что при проектировании АСУ ТП был заложен набор мер защиты, который формирует барьер, препятствующий доступу непривилегированного пользователя со средним уровнем компетенции к системному ПО. Для АСУ ТП меры защиты могут быть выбраны из перечня [46]. Также будем считать, что на момент начала атаки система не является скомпрометированной. Тогда фрагмент биграфа для данного сценария атаки и барьера представлен на рис. 4.

Перейдем теперь непосредственно к анализу уязвимостей. Из базы данных уязвимостей необходимо выгрузить описания уязвимостей, относящихся к используемому в СВУ АСУ ТП программному обеспечению с учетом версий компонентов. Для облегчения работы можно воспользоваться каким-либо сканером уязвимостей.

Пусть для определенности СВУ АСУ ТП реализовано на базе ОС Linux. Рассмотрим для примера уязвимость в базовом компоненте glibc с номером CVE-2020-1752, которая по классификатору CVSS 3.0 имеет высокую степень опасности. Эта уязвимость позволяет локальному нарушителю, передав программе специальным образом сформированный путь к файлу, выполнить произвольный код и нарушить таким образом целостность программного обеспечения. Все эти свойства уязвимости отражены в векторе CVSS 3.0.

В принятой модели безопасности такой уязвимости соответствует сценарий атаки из таблицы, позволяющий выполнить произвольный код без повышения привилегий; против такой атаки действует барьер, препятствующий эксплуатации уязвимости нарушителем в рассматриваемом сценарии атаки.

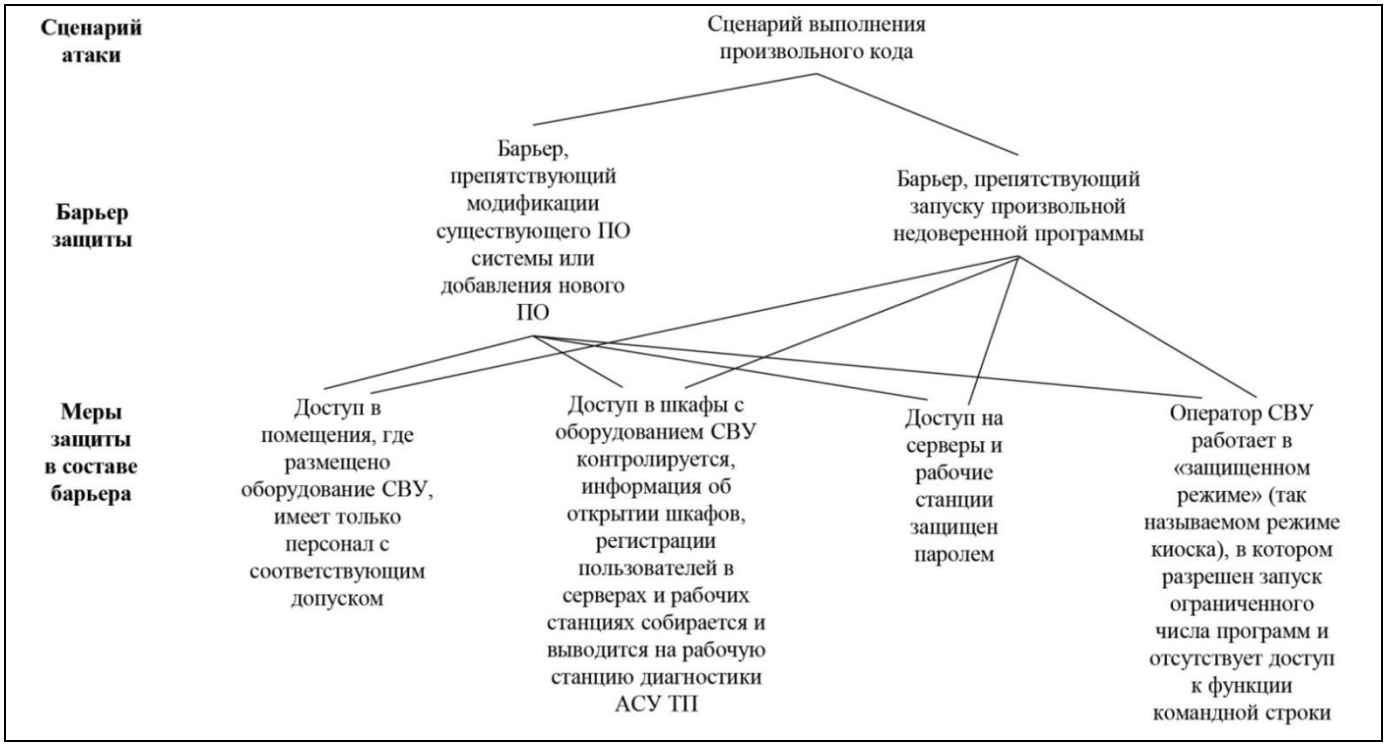

Рис. 4. Биграф, описывающий отношение «сценарий атаки – барьер защиты»

Таким образом, с учетом высокого доверия к способности барьера противостоять неконтролируемому доступу к системному ПО, можно считать, что риск уязвимости минимален и для уязвимости CVE-2020-1752 патчи устанавливать не нужно.

Очевидно, что в отношении уязвимостей, которые могут приводить к сходному сценарию атаки, метод VICS даст такие же результаты.

ЗАКЛЮЧЕНИЕ

Проблемы установки обновлений безопасности стоят довольно остро, что связано с повышением внимания к кибербезопасности во всех областях применения компьютеров. Несмотря на то, что на сегодня имеются выработанные годами рекомендации по использованию патчей [3, 8, 9], применение этих рекомендаций вызывает трудности.

Первая трудность, которая касается информационных систем любого назначения, – это эффект масштаба. На данный момент каждый год открываются тысячи уязвимостей, что делает анализ каждой уязвимости «вручную» неэффективным. Трудности анализа каждой уязвимости по отдельности могут привести к стратегии «патчить все». Однако эта стратегия чревата проблемами с валидацией совместимости компонентов программного обеспечения с оборудованием и друг с другом и нарушением непрерывности защищаемых бизнес-процессов.

Если говорить об АСУ ТП, то их дополнительно отличают долгий жизненный цикл (десятки лет), малая изменчивость аппаратного обеспечения в ходе эксплуатации, достаточно длительный этап разработки и зачастую необходимость сертификации ПО. Отсюда следует, что даже если в принятом к эксплуатации ПО АСУ ТП будут устранены все известные на момент приемки уязвимости, то к моменту завершения пусконаладки на объекте ПО наверняка будет содержать вновь открытые уязвимости, которые невозможно оперативно устранить. Неизменность аппаратного обеспечения приведет к тому, что через несколько лет эксплуатации многие компоненты ПО будет невозможно обновить, не нарушив программно-аппаратную совместимость системы, так как новые версии ПО от сторонних производителей могут перестать поддерживать устаревшее оборудование. Отраслевые правила могут потребовать повторной сертификации после установки обновлений, что дополнительно приведет к затратам времени и средств. Необходимость ресертификации после каждого изменения ПО является существенной и осознается не только разработчиками, но и регулирующими органами. В частности, на международном уровне рассматривается возможность классификации изменений (патчей) по степени их влияния на функциональность ПО и, в зависимости от этого, изменения требований к ресертификации.

Очевидно, что нужен какой-то метод, позволяющий для конкретной системы анализировать большое количество уязвимостей и выдавать рекомендации о необходимости установки патчей, при этом не требующий полного и детального анализа каждой уязвимости.

Предложенный в настоящей работе метод основан на методе анализа отказов и последствий FMEA, в котором рассматривается не уязвимость сама по себе, а влияние уязвимости как части сценария атаки, с учетом принятой модели угроз и нарушителя, на функции всей системы и с учетом существующих барьеров (наборов мер защиты, эффективных в отношении определенного сценария атаки).

В соответствии с принципами безопасного проектирования в систему встраиваются определенные наборы мер защиты, которые формируют гарантированные барьеры для определенных классов уязвимостей, и на этапе эксплуатации именно эти барьеры задействуются.

Отношения «сценарий атаки – барьер» составляют биграф, а анализ рисков при появлении новых уязвимостей сводится к его анализу.

Если в результате обнаружения новой уязвимости появится сценарий атаки, для которого барьер защиты не установлен или неэффективен, то это означает, что новый источник риска требует принятия дополнительных мер в виде установки обновления безопасности или применения иных компенсирующих мер. Это и есть критерий принятия решений об установке обновлений безопасности.

Так как барьеры рассматриваются в контексте защиты от классов уязвимостей, а не от отдельной уязвимости, то данный подход позволяет бороться не только с открытыми, известными уязвимостями, но и предлагает защиту от еще не открытых уязвимостей, которые всегда присутствуют в сложных программных продуктах.

В заключение отметим, что в сложной системе ручной анализ уязвимостей даже при небольшом количестве сценариев атаки, видов нарушителей и угроз является крайне затруднительным и сводится к задаче, относящейся к массовой проблеме. Авторы полагают, что автоматизация анализа уязвимостей с применением формальных проблемно-ориентированных языков описания уязвимости и

сценариев атаки дает возможность применить разработанный метод к реальным системам управления, и ведут работу в этом направлении.

ЛИТЕРАТУРА

1. *IEC* TR 62443-2-3. Technical report. Security for industrial automation and control systems – Part 2-3: Patch management in the IACS environment. – Geneva: International Electrotechnical Commission, 2015. – 61 p.
2. *Методика* тестирования обновлений безопасности программных, программно-аппаратных средств: утв. ФСТЭК России 28 октября 2022 г. [*Metodika* testirovaniya obnovlenii bezopasnosti programmnykh, programmno-apparatnykh sredstv: appr. by FSTEC of Russia on October 28, 2022. (In Russian)]
3. *Souppaya, M., Scarfone, K.* Guide to enterprise patch management planning: preventive maintenance for technology. Special Publication (NIST SP) – 800-40r4. – Gaithersburg, MD: National Institute of Standards and Technology, 2022. – DOI: https://doi.org/10.6028/NIST.SP.800-40r4
4. *National* Information Assurance (IA) Glossary. CNSS Instruction No. 4009. – Fort Meade: Committee on National Security Systems Instruction, 2015. – 103 p.
5. *CVE*. – URL: https://cve.mitre.org (дата обращения: 02.10.2024). [Accessed October 2, 2024].
6. *Банк* данных угроз безопасности информации. – URL: https://bdu.fstec.ru/vul (дата обращения: 02.10.2024). [*Bank* dannykh ugroz bezopasnosti informatsii. – URL: https://bdu.fstec.ru/vul (accessed October 2, 2024). (In Russian)]
7. *CERT-FR* avis. – URL: https://www.cert.ssi.gouv.fr/avis/ (дата обращения: 02.02.2024). [Accessed February 2, 2024].
8. *ISO/IEC* TS 9569:2023. Technical specification. Information security, cybersecurity and privacy protection – Evaluation criteria for IT security – Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045. – Geneva: International Standard Organization/IEC, International Electrotechnical Commission, 2023. – 36 p.
9. Руководство по организации процесса управления уязвимостями в органе (организации): утв. ФСТЭК России 17 мая 2023 г. [*Rukovodstvo* po organizatsii protsessa upravleniya uyazvimostyami v organe (organizatsii): appr. by FSTEC of Russia on May 17, 2023. (In Russian)]
10. Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств: утв. ФСТЭК России 28 октября 2022 г. [*FSTEHK* Rossii. Metodika otsenki urovney kritichnosti uyazvimostey programmnykh, programmno-apparatnykh sredstv: appr. by FSTEC of Russia on October 28, 2022. (In Russian)]
11. *Scarfone, K., Souppaya, M., Dodson, D.* Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. Special Publication (NIST SP) - 800-218. – Gaithersburg, MD: National Institute of Standards and Technology, 2022. – DOI: https://doi.org/10.6028/NIST.SP.800-218
12. Методика оценки угроз безопасности информации: утв. ФСТЭК России 5 февраля 2021 г. [*Metodika* otsenki ugroz bezopasnosti informatsii: appr. by FSTEC of Russia on February 5, 2021. (In Russian)]
13. *ГОСТ* Р 56205-2014 IEC/TS 62443-1-1:2009. Сети коммуникационные промышленные. Защищенность (кибербезопасность) сети и системы. Часть 1-1. Терминология, концептуальные положения и модели: стандарт. – М.: Стандартинформ, 2020. – 72 р. [*IEC/TS* 62443-1-1:2009 Industrial communication networks. – Network and system security. – Part 1-1: Terminology, concepts and models (IDT). – Geneva: International Electrotechnical Commission, 2009. – 81 p.]
14. *ГОСТ* Р 56939-2016. Защита информации. Разработка безопасного программного обеспечения. Общие требования. – М.: Стандартинформ, 2016. – 24 с. [*GOST* R 56939-2016. Zashchita informatsii. Razrabotka bezopasnogo programmnogo obespecheniya. Obshchie trebovaniya. – Moscow: Standartinform, 2016. – 24 s. (In Russian)]
15. *ГОСТ* 19.101-77. Единая система программной документации. Виды программ и программных документов. – М.: Стандартинформ, 2010. [*GOST 19.101-77.* Edinaya sistema programmnoi dokumentatsii. Vidy programm i programmnykh dokumentov. – Moscow: Standartinform, 2010. (In Russian)]
16. *Промыслов В.Г., Жарко Е.Ф.* Подходы к оценке риска кибербезопасности АСУТП АЭС // Автоматизация в промышленности. – 2022. – № 11. – С. 28–33. [*Promyslov, V.G., Zharko, E.F.* Approaches to risk assessment in cybersecurity of A-plant process control systems // Automation in Industry. – 2022. – No. 11. – P. 28–33. (In Russian)]
17. *Dissanayake, N., Jayatilaka, A., Zahedi, M., AliBabar, M.* Software security patch management - A systematic literature review of challenges, approaches, tools and practices // Information and Software Technology. – 2022. – Vol. 144. – Art. no. 106 771.
18. *ГОСТ* Р 51901.12-2007 (МЭК 60812:2006) Менеджмент риска. Анализ видов и последствий отказов.: стандарт. – М.: Стандартинформ, 2008. – 35 р. [*IEC* 60812:2006 Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA). – Geneva: International Electrotechnical Commission, 2006. – 93 p.]
19. *About* CWE. – URL: https://cwe.mitre.org/about/index.html (дата обращения: 22.04.2024). [Accessed April 22, 2024].
20. *ГОСТ* Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности. – М.: Стандартинформ, 2011. – 51 с. [*GOST* R ISO/MEK 27005-2010. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Menedzhment riska informatsionnoi bezopasnosti. – M.: Standartinform, 2011. – 51 s. (In Russian)]
21. *ГОСТ* Р ИСО/МЭК 15408-3-2008. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности. – М.: Стандартинформ, 2008. – 118 с. [*GOST* R ISO/MEK 15408-3-2008. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologii. Chast' 3. Komponenty doveriya k bezopasnosti. – M.: Standartinform, 2008. – 118 s. (In Russian)]
22. *CVE* Metrics. – URL: https://www.cve.org/About/Metrics (дата обращения: 22.04.2024). [Accessed April 22, 2024].
23. *Haddad, A., Aaraj, N., Nakov P.*, et al. Automated Mapping of CVE Vulnerability Records to MITRE CWE Weaknesses. – arXiv:2304.11130, 2023. – DOI: 10.48550/arXiv.2304.11130
24. *Lin, Y.-Z., Mamun, M., Chowdhury, V.A.*, et al. HW-V2W-Map: Hardware Vulnerability to Weakness Mapping Framework for Root Cause Analysis with GPT-assisted Mitigation Suggestion. – arXiv:2312.13530, 2023. – DOI: https://doi.org/10.48550/arXiv.2312.13530
25. *Mell, P., Scarfone, K., Romanosky, S.* A Complete Guide to the Common Vulnerability Scoring System Version 2.0. – 2007. – 23 p. – URL: https://www.first.org/cvss/v2/guide (дата обращения: 01.04.2024). [Accessed April 1, 2024].
26. *Common* Vulnerability Scoring System v3.0: Specification Document. – URL: https://www.first.org/cvss/v3.0/specificati

on-document (дата обращения: 22.09.2024). [Accessed September 22, 2024].
27. *Калькулятор* CVSS V3. – URL: https://bdu.fstec.ru/calc3 (дата обращения: 22.09.2024). [*Calculiator* besjpasonosti. – URL: https://bdu.fstec.ru/calc3 (accessed September 22, 2024). (In Russian)].
28. *Kekül, H., Ergen, B., and Arslan, H*. Comparison and Analysis of Software Vulnerability Databases // Int. J. Eng. Manuf.. – 2022. – Vol. 12, no. 4. – P. 1–14.
29. *How* We Assess Acceptance Levels. – URL: https://nvd.nist.gov/vuln/cvmap/How-We-Assess-Acceptance-Levels (дата обращения: 26.04.2024). [Accessed April 26, 2024].
30. *CAPEC* List. – URL: https://capec.mitre.org/data/index.html (дата обращения: 31.05.2024). [Accessed April 22, 2024].
31. *MITRE* ATT&CK. – URL: https://attack.mitre.org/ (дата обращения: 22.04.2024). [Accessed April 22, 2024].
32. *Microsoft* Threat Modeling Tool. – URL: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats (дата обращения: 31.05.2024). [Accessed May 31, 2024].
33. *Pietre-Cambacedes, L., Chaudet, C.* Disentangling the relations between safety and security // Proceedings of the 9th WSEAS International Conference on Applied Informatics and Communications. – Stevens Point, Wisconsin, 2009. – P. 156–161.
34. *Boehs, E.* Everything I Know About the XZ Backdoor. – URL: https://boehs.org/node/everything-i-know-about-the-xz-backdoor (дата обращения: 10.04.2024). [Accessed April 10, 2024].
35. *CVE-2022-23812*. – URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-23812 (дата обращения: 10.04.2024). [Accessed April 10, 2024].
36. *Капранов О., Гуреева Ю.* Сотрудникам Минпросвещения запретили пользоваться техникой Apple. – Текст: электронный // Российская газета. – 19.07.2023. – URL: https://rg.ru/2023/07/19/sotrudnikam-minprosveshcheniia-zapretili-polzovatsia-tehnikoj-apple.html (дата обращения: 22.10.2024). [*Kapranov O., Gureeva Yu.* Sotrudnikam Minprosveshcheniya zapretili pol'zovat'sya tekhnikoi Apple. – Tekst: elektron-nyi // Rossiiskaya gazeta. – July 7, 2023. – URL: https://rg.ru/2023/07/19/sotrudnikam-minprosveshcheniia-zapretili-polzovatsia-tehnikoj-apple.html (accessed October 22, 2024). (In Russian)]
37. *Commerce* Department Prohibits Russian Kaspersky Software for U.S. Customers // Bureau of Industry and Security. – June 20, 2024. – URL: https://www.bis.gov/press-release/commerce-department-prohibits-russian-kaspersky-software-us-customers (дата обращения: 20.12.2024). [Accessed December 20, 2024].
38. *Semenkov, K., Promyslov, V., Poletykin, A.,* et al. Validation of Complex Control Systems with Heterogeneous Digital Models in Industry 4.0 Framework // Machines. – 2021. – Vol. 9. – No. 3. – Art. no. 62.
39. *ГОСТ* ISO/IEC TS 19249-2021. Информационные технологии. Методы и средства обеспечения безопасности. Каталог принципов построения архитектуры и проектирования безопасных продуктов, систем и приложений. – М.: Стандартинформ, 2021. – 32 с. [*GOST* ISO/IEC TS 19249-2021. Informatsionnye tekhnologii. Metody i sredstva obespecheniya bezopasnosti. Katalog printsipov postroeniya arkhitektury i proektirovaniya bezopasnykh produktov, sistem i prilozhenii. – M.: Standartinform, 2021. – 32 s. (In Russian)]
40. *MIL-P* 1629. USA Military Standard, Procedure for Performing a Failure Mode, Effects and Criticality Analysis. – Washington, DC: Department of Defense, 1980. – 54 p.
41. *Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E*. Security Application of Failure Mode and Effect Analysis (FMEA) // In: Lecture Notes in Computer Science. – 2014. – Vol. 8666. – P. 310–325. – DOI: https://doi.org/10.1007/978-3-319-10506-2_21
42. *Talwar, P.* Software Failure Mode and Effects Analysis // Advances in Intelligent Systems and Computing. – 2020. – Vol. 1131. – P. 86–91.
43. *Busquim e Silva, R.A., Piqueira, J.R.C., Cruz, J.J., Marques R.P*. Cybersecurity Assessment Framework for Digital Interface Between Safety and Security at Nuclear Power Plants // International Journal of Critical Infrastructure Protection. – 2021. – Vol. 34. – Art. no. 100453. – DOI: https://doi.org/10.1016/j.ijcip.2021.100453.
44. *Бугайский К.А., Калашников А.О., Бирин Д.С.* и др. Применение логико-вероятностного подхода в информационной безопасности (Часть 1) // Cybersecurity Issues. – 2023. – No. 4 (56). – P. 23–32. [*Kalashnikov, A.O., Bugajskij, K.A., Birin D.S.,* et al. Application of the logical-probabilistic method in information security (part 1) // Cybersecurity Issues. 2023. – No. 4 (56). – P. 23–32. (In Russian)]
45. *Менгазетдинов Н.Э., Полетыкин А.Г., Промыслов В.Г.* и др. Комплекс работ по созданию первой управляющей системы верхнего блочного уровня АСУ ТП для АЭС «Бушер» на основе отечественных информационных технологий. – М.: ИПУ РАН. – 2013. – 95 с. [*Mengazetdinov, N.E., Poletykin, A.G., Promyslov, V.G.,* et al. Kompleks rabot po sozdaniyu pervoi upravlyayushchei sistemy verkhnego blochnogo urovnya ASU TP dlya AES “Busher” na osnove otechestvennykh informatsionnykh tekhnologii. – Moscow: IPU RAN. – 2013. – 95 p. (In Russian)]
46. Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации: утв. приказом ФСТЭК России от 25 декабря 2017 г. № 239. [Trebovania po obespecheniyu bezopasnosti znachimyh ob’ektov kriticheskoi informatsionnoi infrastruktury Rossijskoj Federatsii: appr. by FSTEC of Russia on December 25, 2017. (In Russian)]

Статья представлена к публикации членом редколлегии Р.В. Мещеряковым.

Поступила в редакцию 06.06.2024,
после доработки 18.03.2025.
Принята к публикации 28.04.2025.

Семенков Кирилл Валерьевич – канд. физ.-мат. наук,
✉ semenkov@ipu.ru
ORCID iD: https://orcid.org/0000-0003-0865-9072

Промыслов Виталий Георгиевич – канд. физ.-мат. наук,
✉ vp@ipu.ru
ORCID iD: https://orcid.org/0000-0003-1919-8718

Институт проблем управления им. В. А. Трапезникова РАН, г. Москва

A PROCEDURE FOR ASSESSING SECURITY UPDATES IN INDUSTRIAL SYSTEMS

K. V. Semenkov* and V. G. Promyslov**

*,**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ semenkov@ipu.ru, **✉ vp@ipu.ru

Abstract. This paper is devoted to the problem of applying cybersecurity updates (patches) for the software of instrumentation and control systems (ICS) with a long lifecycle. The problem is considered for the system operation stage. The main focus is on the large number of vulnerabilities found in software, the complexity of analyzing the impact of a vulnerability on system security, and the requirements for testing the compatibility of updates and software certification after changes have been made. Based on the Failure Mode and Effects Analysis (FMEA), a procedure is proposed to simplify the analysis of the impact of a vulnerability on cybersecurity. This procedure considers a smaller set of attack scenarios rather than each vulnerability separately. The analysis of attack scenarios also covers the effect of security measures. The procedure includes simple criteria for applying security updates based on the analysis results. An example of vulnerability analysis using this procedure is provided.

Keywords: vulnerability, patch, risk assessment, instrumentation and control system (ICS), cybersecurity, criterion.