ся, что для защиты корпоративной информации будут использоваться специальные программы.

Список литературы

1. Львович Я.Е., Преображенский А.П., Преображенский Ю.П., Клименко Ю.А. Анализ программно-технических решений блокировки запрещенной информации в информационно-телекоммуникационных сетях // Информационные технологии. 2022. Т. 28. № 8. С. 429-437.

2. Бормотов В.Е. Проблемы защиты информации в компьютерной сети // Молодой ученый. 2016. № 11 (115). С. 148-150. URL: https://moluch.ru/archive/115/31145/ (дата обращения: 15.01.2025).

3. Аветисян Т.В., Львович Я.Е., Преображенский А.П. Алгоритмы моделирования для оценки рисков в киберфизических системах // Телекоммуникации. 2023. № 2. С. 9-15.

4. Поначугин А.В. Проблемы и реализация комплекса мер безопасности компьютерных сетей // Вестник НГИЭИ. 2016. № 2 (57). URL: https://cyberleninka.ru/article/n/problemy-i-realizatsiya-kompleksa-mer-bezopasnosti-kompyuternyh-setey (дата обращения: 15.01.2025).

5. Келдыш Наталья Всеволодовна Системная защита информации компьютерных сетей: учебное пособие. М.: Мир науки, 2022. URL: https://izdmn.com/PDF/43MNNPU22.pdf (дата обращения: 15.01.2025).

БЕЗОПАСНАЯ РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ОБЪЕКТА КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ НА ОСНОВЕ ИНТЕГРАЦИИ МЕТОДОЛОГИИ DEVSECOPS В ПРОЦЕСС ПРОЕКТИРОВАНИЯ

Сверчков Р.В., Андрияхов Я.В.

Российский экономический университет им. Г. В. Плеханова, Москва, e-mail: rvsverchkov@gmail.com

Научный руководитель: Микрюков А.А.

Введение

В современных реалиях веб-приложения становятся все более сложными по технической реализации, что приводит к повышенной вероятности возникновения уязвимостей [5]. Вместе с этим возрастает и зависимость критической информационной инфраструктуры (КИИ) от применяемых решений и подходов к проектированию, что делает задачи по обеспечению киберустойчивости первостепенными [2].

Распространенные методы защиты (антивирусы, межсетевые экраны и т.д.) могут противостоять многим атакам, однако практика показывает, что при разработке программных продуктов вопросы информационной безопасности чаще всего рассматриваются лишь на заключительных этапах жизненного цикла. Это приводит к неэффективному выявлению и устранению уязвимостей и, как следствие, к большим затратам на их исправление [4]. С учетом этого использование интегрированных подходов на основе DevSecOps становится востребованным, особенно в сфере КИИ [3].

В данной статье предлагается усовершенствованный поэтапный подход к интеграции методологии DevSecOps, дополняющий стандартную схему встроенными механизмами количественной оценки уязвимостей и анализом затрат на их устранение. Также приводится сравнительная оценка предлагаемого решения с традиционными методами безопасной разработки, где меры защиты внедряются преимущественно на поздних стадиях SDLC (Systems development life cycle).

Целью данной работы является применение комплексного решения DevSecOps, адаптированного под задачи объектов КИИ, снижающего общее число уязвимостей и повышающего оперативность реагирования на угрозы.

Пример поэтапной интеграции DevSecOps при разработке веб-приложения для объекта критической информационной инфраструктуры

Создание современных веб-приложений требует особенного внимания к безопасности компонентов на каждом этапе жизненного цикла разработки [1]. Непосредственная интеграция мер защиты с учетом особенностей разработки веб-приложений позволяет значительно снизить риски, а также обеспечить надежную работу веб-приложений в условиях постоянно меняющихся киберугроз. Ниже приведен пример поэтапной интеграции DevSecOps в рамках разработки веб-приложения для системы управления водоснабжением города. Рассматриваемое приложение является объектом критической информационной инфраструктуры [2].

1. Планирование и анализ угроз;

Данный этап являются ключевыми в обеспечении информационной безопасности на ранних стадиях разработки. На этом этапе формируется перечень нормативных требований с учётом отечественных и международных стандартов (ФСТЭК, ГОСТ Р ИСО/МЭК 27001-2019, ISO/IEC 27001), проводится моделирование угроз по методологии STRIDE и определяются метрики, позволяющие объективно оценить уровень защищённости системы. Такой комплексный подход даёт возможность чётко обозначить приоритеты и сроки устранения уязвимостей, что существенно повышает надёжность и устойчивость будущего продукта к киберугрозам, а именно:

• Требования безопасности. Формируется перечень нормативных требований на основе стандартов ФСТЭК, ГОСТ Р ИСО/МЭК 27001-2019 [1], а также международных рекомендаций (ISO/IEC 27001);

• Моделирование угроз. Проводится сессия с применением методологии STRIDE [6], формируются сценарии возможных атак;

• Определение метрик. На этом этапе задаются целевые показатели: допустимое количе-

ство уязвимостей уровня High и Critical, а также предельное время на их устранение.

В традиционной схеме разработки безопасность зачастую оговаривается поверхностно, что не позволяет формализовать четкие метрики и точки контроля.

2. Разработка и безопасное кодирование;

• Проектирование архитектуры. Учитываются результаты моделирования угроз, разрабатываются решения для снижения рисков (например, разделение подсистем, минимизация прав доступа, обязательный шифрованный канал).

• Реализация требований стандартов безопасного программирования. Внедряются гайды (чек-листы) безопасного кодирования, в том числе рекомендации OWASP (Open Worldwide Application Security Project) [5].

• Статический анализ кода (SAST). Инструменты вроде SonarQube интегрируются в IDE, чтобы в реальном времени выявлять проблемные участки кода.

Сравнительный эффект. При классическом подходе проверка кода на безопасность часто переносится на этап тестирования, в результате чего затраты на исправление выше в 2–3 раза по сравнению с ранним обнаружением.

3. Сборка и тестирование;

• CI/CD (Continuous Integration/Continuous Delivery). Внедрение системы интеграции Jenkins (Программная система с открытым исходным кодом на Java, предназначенная для обеспечения процесса непрерывной интеграции программного обеспечения). или GitLab CI/CD для автоматического запуска тестов при каждом обновлении продукта.

• Применение инструментов динамического анализа DAST и IAST. Используются инструменты динамического (OWASP ZAP) и интерактивного анализа (Contrast Security), позволяющие обнаруживать уязвимости в реальных условиях выполнения.

• Пентесты. Автоматизированные и ручные тесты на проникновение с привлечением внешних экспертов.

• Контрольный чекпоинт. После выполнения тестов и пентеста сравнивают результаты с установленными метриками (количество уязвимостей и время на их устранение). В случае несоответствия пайплайн блокируется, и команда возвращается к доработке кода.

Сравнительный эффект. В классической схеме пентесты нередко проводят непосредственно перед релизом, что может привести к значительному сдвигу сроков выпуска продукта из-за срочного устранения найденных проблем.

4. Релиз и управление версиями;

• Управление конфигурациями. Ansible, Puppet [7] или другие инструменты автоматически применяют проверенные конфигурации серверов, сетевого окружения и баз данных.

• Версионирование. Используется Git, pull requests с обязательным код-ревью, что предотвращает «случайные» изменения, повышающие уязвимость приложения.

• Система показателей релиза. В момент выхода каждой версии анализируются оставшиеся «открытые» проблемы по безопасности и время, затраченное на их исправление.

5. Развертывание и безопасность инфраструктуры;

• Контейнеризация. Docker для изоляции сред [8], интеграция с Kubernetes с политиками безопасности (Pod Security Policies, Network Policies).

• Внутриконтурная безопасность. Организация VPN, межсетевых экранов, сегментация сети для критичных компонентов.

• Мониторинг. Prometheus, Grafana для постоянного контроля метрик производительности и показателей безопасности.

6. Эксплуатация и мониторинг;

• SIEM-система. Например, RuSIEM или аналоги, позволяющие выявлять аномальную активность и коррелировать события безопасности.

• Сканирование уязвимостей. Периодический автоматизированный аудит (Nessus, Qualys) с сопоставлением с исходными метриками.

• Управление инцидентами. Разработанная процедура реагирования с привязкой к ответственным лицам, временам восстановления и каналам уведомления.

7. Аудит и обратная связь.

• Регулярные аудиты. Внутренние и внешние проверки соответствия требованиям ФСТЭК и собственным политикам безопасности.

• Актуализация метрик. По результатам аудитов и отзывов адаптируют значения целевых показателей, оптимизируют процесс.

Комплексный подход к информационной безопасности, реализованный на каждом этапе жизненного цикла – от планирования и определения требований до аудита и непрерывного мониторинга, обеспечивает своевременное выявление и устранение уязвимостей. В итоге продукт выходит на рынок более защищённым, соответствующим требованиям ФСТЭК и международным стандартам, при этом снижаются затраты на исправление ошибок, а также повышается доверие пользователей и заинтересованных сторон.

Сравнительный анализ результатов (DevSecOps vs традиционный подход)

В таблице приведен анализ и сопоставление показателей при использовании предлагаемого подхода DevSecOps (с дополнительными метриками и автоматизированными чекпоинтами) и традиционного подхода, в котором безопасность учитывается преимущественно в финальной стадии разработки.

Сравнительные показатели разработки веб-приложения

Показатель	При применении методологии DevSecOps	При традиционном подходе
Среднее время на выявление уязвимостей	1–2 дня (благодаря непрерывным проверкам)	7–10 дней (проведение финальных тестов)
Стоимость исправления 1 уязвимости	Условно 1 единица времени (обнаружение на ранних этапах)	В 2-3 раза дольше (иногда требует переделки архитектуры)
Количество критических уязвимостей к релизу	Снижено за счет раннего статического анализа	Часто выявляются перед запуском, влекут задержки
Риск нарушения сроков релиза	Низкий, т.к. уязвимости устраняются постепенно	Высокий, т.к. серьезные проблемы выявляются на финише
Уровень соблюдения требований ФСТЭК	Высокий (постоянный контроль)	Средний (принимаются меры только к концу проекта)

Представленные результаты иллюстрируют, что предложенный интегрированный подход позволяет существенно уменьшить затраты на исправление уязвимостей, а также снижает риск срывов сроков релиза и невыполнения требований регуляторов. Экономический эффект достигается за счет объединения процессов разработки, тестирования и обеспечения безопасности в единый конвейер.

Преимущества интеграции методологии DevSecOps в контексте критической информационной инфраструктуры

Интеграция подходов и практик DevSecOps в процессы разработки веб-приложений для объектов КИИ способна принести ряд существенных преимуществ:

• Повышение уровня безопасности. Раннее выявление и исправление уязвимостей благодаря постоянному анализу.

• Соблюдение нормативных требований. Учет регламентов ФСТЭК, ISO/IEC 27001 и других стандартов с самого начала проекта.

• Ускорение процессов разработки и развертывания. Автоматизированные тесты и проверенные конфигурации инфраструктуры снижают рутинные операции.

• Улучшение качества программного продукта. Непрерывное выявление ошибок на всех этапах и формирование «безопасной» архитектуры.

• Экономическая эффективность. Снижение затрат на доработку кода, улучшение прогнозирования бюджета.

• Улучшение взаимодействия между командами. Единый конвейер CI/CD стимулирует обмен знаниями между разработчиками, тестировщиками и специалистами по безопасности.

• Гибкость и адаптивность к изменениям. Возможность вносить доработки в приложение без значительного риска сломать уже реализованные механизмы защиты.

• Повышение бесперебойности работы. Минимизируется риск критичных простоев и инцидентов.

Заключение

В данной статье представлен поэтапный подход к интеграции DevSecOps, учитывающий специфику объектов КИИ. В отличие от классических методов разработки, где безопасность остается «точечной» мерой на завершающих стадиях, здесь безопасность становится непрерывной частью процесса и тем самым повышая общий уровень защищенности конечного продукта.

В ходе исследования было показано, что:

• Раннее выявление уязвимостей уменьшает совокупные затраты на их устранение (до 2–3 раз).

• Непрерывный мониторинг метрик безопасности повышает прозрачность процессов и снижает риск невыполнения требований регуляторов (ФСТЭК).

• Встраивание контрольных точек (checkpoints) и метрик безопасности позволяет оперативно реагировать на изменения и поддерживать проект на требуемом уровне защищенности.

Таким образом, внедрение DevSecOps, расширенного механизмами количественной оценки рисков и затрат, оправдывает себя для объектов КИИ, снижая вероятность сбоев и киберинцидентов, а также улучшая экономические показатели проектов. Предложенный метод может быть адаптирован и для других типов критических систем, где особо важна стабильность и защищенность.

Список литературы

1. ГОСТ Р ИСО/МЭК 27001-2019. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования.

2. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации».

3. Приказ ФСТЭК России от 11 февраля 2013 г. № 17 «Об утверждении требований по защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».

4. Кулиш А.А., Фролова Е.А. Методология DevSecOps в обеспечении безопасности информационных систем // Информационная безопасность. 2018. № 3. С. 45–49.

5. OWASP Foundation. OWASP Top Ten Project, 2021. [Электронный ресурс]. URL: https://owasp.org/www-project-top-ten/ (дата обращения: 04.02.2025).

6. Microsoft Corporation. The STRIDE Threat Model, 2005. [Электронный ресурс]. URL: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20) (дата обращения: 04.02.2025).

7. Ansible Documentation. Red Hat, 2021. [Электронный ресурс]. URL: https://docs.ansible.com/ (дата обращения: 04.02.2025).

8. Docker Documentation. Docker Inc., 2021. [Электронный ресурс]. URL: https://docs.docker.com/ (дата обращения: 04.02.2025).

О ПРОЕКТИРОВАНИИ ТЕЛЕКОММУНИКАЦИОННЫХ КОМПОНЕНТОВ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Соколов А.С., Уланова Ю.А.

АНОО ВО «Воронежский институт высоких технологий», Воронеж, e-mail: bbosly@yandex.ru

Информационная инфраструктура общества обеспечивает потребности общества в информации, необходимой для всех видов деятельности людей и организаций. Решение проблемы компьютеризации общества основано на необходимости создания информационно-вычислительных сетей и сетей передачи данных различного назначения.

Анализ современного состояния телекоммуникаций в России показывает, что основным направлением в создании телекоммуникационных сетей (ТС) является маршрутизация, основанная на использовании коммутируемых телефонных сетей общего пользования с одновременным внедрением современных цифровых сетей с высокоскоростными каналами, обеспечивающими реализацию самых передовых протоколов передачи информации основан на использовании аппаратного обеспечения с высокой степенью интеграции.

Достижения в области интегрированных электронных устройств позволили разработать новые информационные технологии [1,2]. Один из наиболее перспективных методов высокоскоростной передачи информации, мультиплексирования и коммутации, ориентированный на передачу разнородного трафика, асинхронный режим передачи – международная стандартизация ATM (Asynchronous Transfer Mode) организации рекомендуют использовать технологию ATM в качестве основы для построения широкополосной цифровой интегрированной сервисной сети (SCSIOS)[3].Это определяет актуальность задачи создания методов и инструментов, направленных на поддержку проектирования сети с использованием технологии ATM.

Сложность и дороговизна современных перспективных телекоммуникационных сетей не позволяют проводить работы, основанные на формировании архитектуры, выборе основных проектных параметров и оценке характеристик, основываясь исключительно на инженерной интуиции. При разработке перспектив или модернизации существующей системы необходимо провести предварительные исследования, чтобы продемонстрировать рассматриваемые проекты и предложения с точки зрения технических и экономических показателей.

На основе анализа, проведенного в этой работе, можно сделать следующие замечания относительно зарубежных программных средств, которые следует рассмотреть [4,5]:

- поскольку большинство продуктов основаны на методах имитационного моделирования и требуют уточнения ряда специфических деталей для моделируемой модели, область применения ограничена.

- математические методы и вычислительные алгоритмы, используемые при моделировании рассматриваемого пакета, скрыты в программе, и без проверки реального объекта невозможно оценить его надежность и валидность.

- стоимость такого рода продукта очень высока.

На основе обзора исследований по математическому моделированию сетей ATM предложена модель, описывающая передачу информации по сети, представленной графом, где расположены коммутаторы ATM в узлах, а ребра являются каналами передачи. Существует явный недостаток теоретических исследований, посвященных сложному многоуровневому моделированию сетей банкоматов распределенной структуры. Мы можем сделать вывод, что ячейки ATM (в технологии ATM вместо термина «пакет» используется термин «ячейка», а ячейки ATM имеют фиксированную длину 53 байта). Большая часть работы по этой теме посвящена моделированию только фрагментов сети.

Поэтому актуальной является задача создания автоматизированных средств поддержки принятия решений при проектировании распределенных телекоммуникационных сетей, использующих технологию ATM, основанную на новых математических моделях и методах.

Моделирование является основным современным методом исследования сложных информационных систем на всех этапах разработки, верификации и модернизации. Важнейшие задачи анализа и синтеза транспортных средств решаются методами моделирования. Моделирование позволяет разработчикам систем экспериментировать с существующими или предлагаемыми системами, когда это нецелесообразно или невозможно сделать с реальными объектами. На основе результатов моделирования может быть построена исчерпывающая зависимость между параметрами транспортного средства и функциями, характеризующими его характеристики. Мы исследуем качественные закономерности, которые определяют область