УДК 004.056

АНАЛИЗ МЕТОДОЛОГИЙ DEVOPS И DEVSECOPS

М. А. Ганжур, Н. В. Дьяченко, А. С. Отакулов
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Модель, получившая имя DevSecOps, подразумевает обеспечение безопасности на всех этапах разработки приложений. Иными словами, контроль безопасности и разработка осуществляются параллельно, причем безопасность стараются внедрить в каждую часть процесса разработки. В случае с моделью DevSecOps речь идет о попытке автоматизировать основные задачи безопасности, внедряя контроль этих процессов на раннем этапе реализации методологии DevOps. Этот подход выгодно отличается от того, что было принято до DevSecOps — контроль безопасности являлся заключительным процессом и осуществлялся в конце разработки. Модель DevSecOps может использоваться, например, при переходе на микросервисы, в процессах непрерывной интеграции (Continuous Integration, CI) и непрерывного развертывания (Continuous Deployment, CD) или просто для тестирования облачной инфраструктуры.

Ключевые слова: информационная безопасность, компьютерная безопасность, DevSecOps, DevOps, микросервисы.

ANALYSIS OF DEVOPS AND DEVSECOPS

M. A. Ganzhur, N. V. Dyachenko, A. S. Otakulov
Don State Technical University (Rostov-on-Don, Russian Federation)

The model, called DevSecOps, is about ensuring security at all stages of application development. In other words, security control and development are carried out in parallel, and security is tried to be implemented in every part of the development process. In the case of DevSecOps, it is about trying to automate basic security tasks by implementing control of these processes at an early stage of DevOps. This approach compares favorably with what was adopted before DevSecOps — security control was the final process and was carried out at the end of development. DevSecOps can be used, for example, when migrating to microservices, in Continuous Integration (CI) and Continuous Deployment (CD) processes, or simply for testing cloud infrastructure.

***Keywords*:** information security, computer security, DevSecOps, DevOps, microservices.

Введение. Основное различие между методологиями DevOps и DevSecOps заключается в том, что DevOps представляет собой конвергенцию разработки операций и доставки приложений, а DevSecOps объединяет все это с безопасностью. DevOps фокусируется на технологиях и методах, которые могут помочь разработчикам и операционным группам функционировать вместе для достижения общих целей, в то время как DevSecOps сосредоточена на методах, которые могут добавить функции безопасности в существующий конвейер DevOps.

Основная часть. В группе DevOps разработчики часто используют архитектуру микросервисов, создавая программное обеспечение как набор независимых сервисов, каждый из которых выполняет отдельную функцию. Каждая микрослужба может работать автономно в контейнере или виртуальной машине. Выявлять и устранять производственные проблемы легче в одном микросервисе или контейнере, чем в большой сложной системе.

Инфраструктура как код (IaC) — это метод использования кода для управления и автоматизации вычислительных ресурсов, таких как хосты, виртуальные машины и контейнеры. Разработчики используют IaC для автоматического выполнения ИТ-операций, устраняя

необходимость в ИТ-поддержке и надзоре за задачами, связанными с инфраструктурой. Операционный персонал также может использовать IaC для развертывания сред по запросу и предоставления разработчикам функций самообслуживания.

Политика как код (PaC) — это способ использования кода для управления политиками, такими как решение организации об использовании определенных типов технологий, стандартов безопасности или ИТ-практик. Политики представлены в формате кода, что позволяет автоматически применять политики в организации на всех этапах разработки.

Элементы DevSecOps переносят задачу на более раннюю стадию цикла разработки. Сдвиг безопасности «влево» гарантирует соблюдение стандартов безопасности с момента первой разработки кодовой базы. Задачи разработки считаются выполненными не только тогда, когда выполняются функциональные требования, но и когда кодовая база проверяется на отсутствие недостатков и уязвимостей безопасности [1].

Непрерывный цикл обратной связи поддерживается автоматизированным процессом, который может отслеживать уязвимость системы, ее безопасность и предоставлять предупреждения в реальном времени разработчикам и соответствующим экспертам. Как только проблема безопасности появляется в конвейере разработки, все команды в сотрудничестве немедленно ее исправляют [2]. Постоянная обратная связь регулярно побуждает всех членов команды улучшать свои методы разработки и обслуживания.

Автоматизация является ключевым фактором в соблюдении стандартов и практик DevSecOps на всех этапах жизненного цикла разработки. Автоматизация позволяет командам DevSecOps быстро брать на себя дополнительные обязанности по обеспечению безопасности, включая автоматический анализ кода, мониторинг соответствия и расследование угроз.

Проблемы и решения при внедрении DevSecOps:

1. Персонал должен обладать устойчивостью к изменению условий труда при переходе на DevOps. Для людей является естественным сопротивление смене привычного состояния или процесса. Исполнителям, которые не развивались в командах DevOps, сложнее приспособиться к сотрудничеству между отделами. Команды, которые работали независимо друг от друга, теперь должны функционировать вместе с другими и корректировать свой рабочий процесс. Это требует планирования и терпения. В других случаях команды могут принять изменение, но административные функционеры или руководители выступают против него. Чтобы сделать переход как можно более плавным, заручитесь поддержкой заинтересованных сторон. Четко изложите обоснование перехода с точки зрения повышения производительности, доверия потребителей и финансовых выгод. При планировании перехода на DevSecOps необходимо объединить руководство и членов групп разработки, безопасности и эксплуатации. Это позволит учитывать потребности и приоритеты каждого при планировании своей стратегии, даст возможность каждому попрактиковаться в общении и переговорах, что очень важно для будущего.

2. Для удовлетворения потребностей DevSecOps необходимы инструменты и процессы, несоответствующие ранее применяемым. Разрабатывается все больше и больше инструментов, которые могут не подходить для всех команд. Некоторые из уже используемых инструментов и процессов могут оставаться полезными после перехода. Остальные необходимо переоборудовать или заменить. Интеграция автоматизированных инструментов тестирования безопасности часто является первым шагом, например, инструменты статического (SAST) и динамического (DAST) тестирования безопасности приложений могут использоваться на протяжении всего процесса разработки. Решения об использовании инструментов и процессов команды должны принимать совместно. Если инструмент затруднителен в использовании или напрямую снижает

производительность, он помешает организационным изменениям. Выбранные инструменты и процессы должны быть прозрачно интегрированы и оптимизированы для всех вовлеченных сторон, они должны максимально сфокусировать и автоматизировать рабочий процесс, чтобы упростить совместную работу и повысить производительность.

3. Разработчики не являются специалистами по безопасности. DevSecOps требует, чтобы группы эксплуатации и разработки разделяли обязанности по обеспечению безопасности. Кроме того, команда должна включить процессы безопасности в свой рабочий процесс. Связанная с этим проблема — сложность процесса и требований безопасности. С самого начала только сотрудники службы безопасности будут обладать соответствующими знаниями и навыками, что является проблемой для разработчиков. Им нужно будет больше узнать о методах безопасного кодирования и включить тестирование безопасности в свой повседневный рабочий процесс. Эта интеграция значительно снижает производительность, особенно на ранних этапах [1].

Выводы. Чтобы свести к минимуму вышеуказанные недостатки, необходимо проводить обучение безопасности всех участников проекта DevSecOps. Обучение должно информировать участников, не связанных с безопасностью, о передовых методах безопасности и их важности, а также обучать группы безопасности инструментам и методам, используемым организацией DevOps.

Осторожное использование инструментов также поможет в этом отношении, например, интеграция информации о безопасности и предупреждений в среду разработки (IDE) может помочь исполнителям овладеть навыками безопасного программирования и делать соответствующий выбор при кодировании.

Библиографический список

1. Флорен, М. В. Организация управления доступом / М. В. Флорен // Защита информации. Конфидент. — 1995. — № 5. — С. 87–93.

2. Тарасов, Ю. Контрольно-пропускной режим на предприятии / Ю. Тарасов // Защита информации. Конфидент. — 2002. — № 1. — С. 55–61.

Об авторах:

Ганжур Марина Александровна, старший преподаватель Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), mganzhur@yandex.ru

Дьяченко Никита Владимирович, студент Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), nikita7890@yandex.ru

Отакулов Артур Собирович, студент Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), diletants23z@gmail.com

Authors:

Ganzhur, Marina A., Senior Lecturer, Department of Computing Systems and Information Security, Don State Technical University (1, Gagarin sq., Rostov-on-Don, RF, 344003), mganzhur@yandex.ru

Dyachenko, Nikita V., Student, Don State Technical University (1, Gagarin sq., Rostov-on-Don, RF, 344003), nikita7890@yandex.ru

Atakulov, Artur S., Student, Don State Technical University (1, Gagarin sq., Rostov-on-Don, RF, 344003), diletants23z@gmail.com