15 Руководств по лучшим практикам безопасности DevOps

Описание

Согласно исследованию компании Verizon, почти 58% компаний в прошлом году стали жертвами утечки данных, и из них 41% произошли из-за уязвимостей в программном обеспечении. Из-за таких нарушений организации могут потерять миллионы долларов и даже свою рыночную репутацию. Но в методологии разработки приложений произошла значительная модернизация. Сегодня организации следуют принципам и инструментам DevOps для разработки приложения или программного обеспечения. При подходе DevOps полное приложение не создается за один раз, оно разрабатывается и поставляется итеративно.

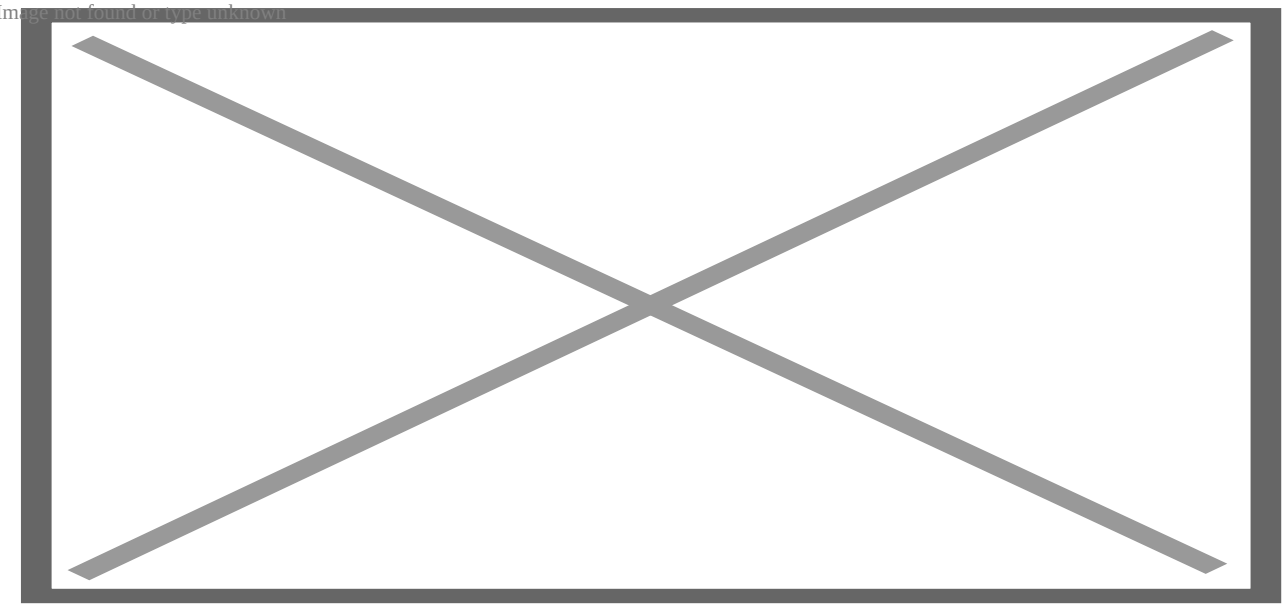
А в некоторых случаях релизы также происходят ежедневно. Но найти проблемы безопасности в ежедневных релизах – задача не из легких. Именно поэтому безопасность является одним из наиболее важных факторов в процессе DevOps. Каждая команда, работающая над созданием приложения, такая как разработчики, тестировщики, операторы и производственники, отвечает за принятие необходимых мер безопасности, чтобы убедиться, что приложение не имеет уязвимостей, ведущих к нарушению безопасности. В этой статье я расскажу о лучших практиках DevOps Security для безопасной разработки и развертывания приложений.

Внедрение модели DevSecOps

DevSecOps – это еще один модный термин в области DevOps. Это

основополагающая практика безопасности при разводе, которую начала применять каждая ИТ-организация. Как следует из названия, это сочетание разработки, безопасности и эксплуатации.

Image not found or type unknown



DevSecOps – это методология использования инструментов безопасности в жизненном цикле DevOps. Таким образом, с самого начала разработки приложения безопасность должна быть его частью. Интеграция процесса DevOps с безопасностью помогает организациям создавать безопасные приложения, в которых нет уязвимостей. Эта методология также помогает устранить разделение между командами разработки и безопасности в организации. Ниже приведены несколько основных практик, которые вы должны внедрить в модель DevSecOps:

- Используйте инструменты безопасности, такие как Snyk, Checkmarx, в конвейере интеграции разработки.
- Все автоматизированные тесты должны оцениваться экспертами по безопасности.
- Команды разработки и безопасности должны сотрудничать для создания моделей угроз.
- Требования безопасности должны иметь высокий приоритет в бэклоге продукта.
- Все политики безопасности инфраструктуры должны быть пересмотрены перед развертыванием.

Просматривайте код в меньшем размере

Вы должны просматривать код в меньшем размере. Никогда не просматривайте огромный код, и не просматривайте все приложение за один раз, это будет ошибкой. Просматривайте код по частям, чтобы вы могли проанализировать его должным образом.

Внедрите процесс управления изменениями

Вам следует внедрить процесс управления изменениями. Теперь, когда в приложении, которое уже находится на стадии развертывания, происходят изменения, вы не хотите, чтобы разработчики продолжали добавлять в него код, добавлять или удалять функции. Поэтому единственное, что может помочь вам на этом этапе, – это внедрение процесса управления изменениями. Каждое изменение, которое необходимо внести в приложение, должно пройти через процесс управления изменениями. Как только оно будет одобрено, разработчику разрешается вносить изменения.

Продолжайте оценивать приложения в производстве

Часто организации забывают о безопасности, когда приложение работает в производстве. Вы должны постоянно проверять приложение. Вы должны постоянно пересматривать его код и периодически проводить тесты безопасности, чтобы убедиться, что в нем не появились новые лазейки.

Вы можете использовать программное обеспечение для обеспечения непрерывной безопасности, такое как Invicti, Probely и Intruder.

Обучение команды разработчиков вопросам безопасности

В соответствии с рекомендациями по безопасности, вы также должны обучить команду разработчиков лучшим практикам безопасности. Так, например, если в команду пришел новый разработчик и он не знает о SQL-инъекциях, вы должны убедиться, что разработчик знает, что такое SQL-инъекция, что она делает и какой

вред может нанести приложению. Возможно, вы не захотите вдаваться в технические тонкости этого вопроса. Тем не менее, необходимо убедиться, что команда разработчиков в курсе новых норм безопасности и лучших практик на широком уровне.

Разработка и внедрение процессов безопасности

Безопасность сама по себе не может работать без процессов, вам необходимо иметь определенные процессы безопасности в вашей организации, а затем внедрить их. После внедрения может возникнуть ситуация, когда вам придется пересмотреть процессы, потому что некоторые вещи не сработали так, как ожидалось, или процесс оказался слишком сложным. Причина может быть любой, поэтому вам придется изменить эти процессы безопасности. Но что бы ни было сделано, вы должны убедиться, что после внедрения процессы безопасности находятся под контролем и аудитом.

Внедрение и обеспечение управления безопасностью

Внедрение и обеспечение соблюдения политик управления в организации должно быть очень важным, если вы хотите внедрить лучшие практики безопасности DevOps. Вы должны создать такие политики управления, которым должны следовать все команды, работающие над созданием приложения, такие как отдел разработки, отдел безопасности, отдел операций и т.д.. Каждый сотрудник должен четко понимать эти политики, поэтому они должны быть очень прозрачными. Вы должны следить за тем, чтобы сотрудники вашей организации придерживались политик управления.

Стандарты безопасного кодирования

Разработчики в основном концентрируются на создании функциональных возможностей приложения и упускают из виду параметры безопасности, поскольку это не является их приоритетом. Но с ростом киберугроз в наши дни вам необходимо убедиться, что ваша команда разработчиков знает о лучших практиках безопасности при кодировании приложения. Они должны знать об инструментах безопасности, которые помогут им выявить уязвимости в коде во время его

разработки, чтобы разработчики могли немедленно изменить код и устранить уязвимости.

Используйте инструменты автоматизации безопасности DevOps

Вы должны начать использовать инструменты автоматизации безопасности в процессах DevOps, чтобы избежать ручной работы. Привлеките средства автоматизации, чтобы вы могли не только проводить тестирование с помощью средств автоматизации, но и создавать повторяемые тесты для приложения. С помощью автоматизированных инструментов для анализа кода, управления секретами, управления конфигурацией, управления уязвимостями и т.д. вы сможете с легкостью разрабатывать безопасные продукты.

Проведите оценку уязвимостей

Вы должны провести оценку уязвимостей, чтобы выявить уязвимые места приложения и устранить их до развертывания в производственной среде.

Это необходимо делать часто, и если обнаружены лазейки в системе безопасности, команда разработчиков должна поработать над своим кодом, чтобы устранить их. Существует множество инструментов для сканирования и управления уязвимостями, которые можно использовать для выявления слабых мест приложения.

Внедрите управление конфигурацией

Вам также следует внедрить управление конфигурацией. Процесс управления изменениями, о котором я рассказывал ранее, также является частью управления конфигурацией. Итак, вам необходимо убедиться, с какой конфигурацией выимеете дело, какие изменения происходят в приложении, кто их авторизует иутверждает. Все это относится к управлению конфигурацией.

Внедряйте модель наименьших привилегий

В лучших практиках безопасности DevOps одним из важнейших правил является использование модели наименьших привилегий. Никогда не давайте никому больше привилегий, чем требуется. Например, если разработчику не требуется доступ ROOT или Admin, вы можете назначить ему доступ обычного пользователя, чтобы он мог работать над необходимыми модулями приложения.

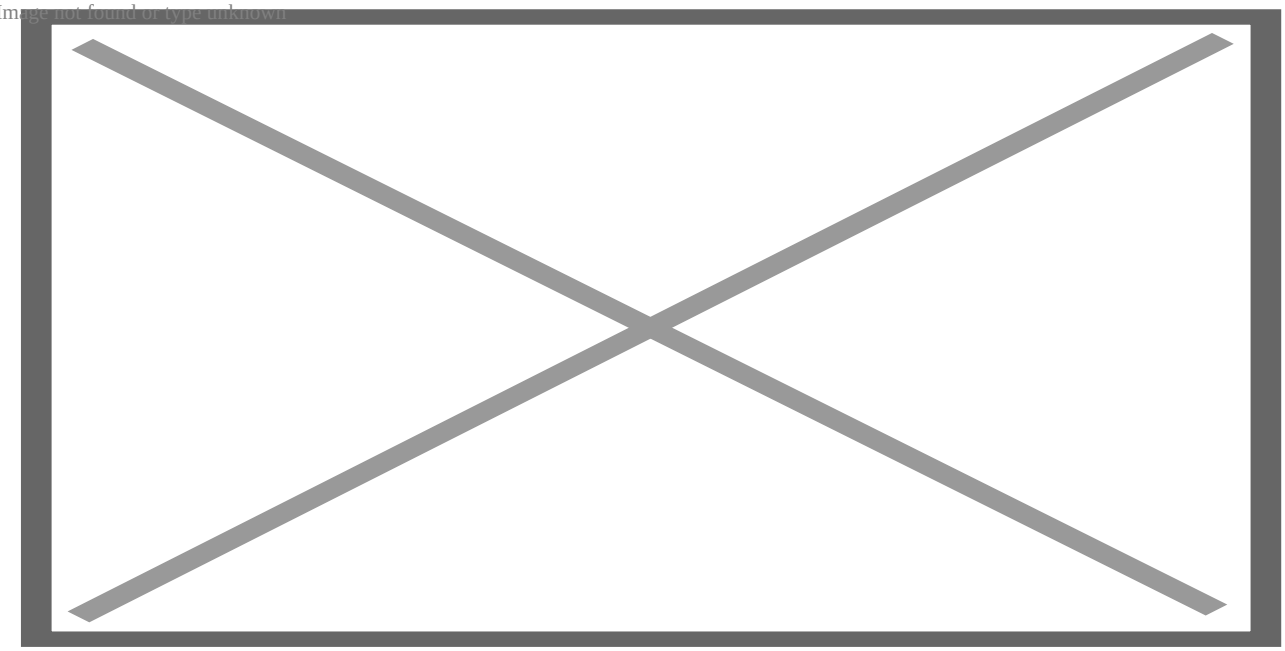
Сегментируйте сеть DevOps

В организации следует применять сегментацию сети. Активы организации, такие как приложения, серверы, хранилища и т.д., не должны работать в одной сети, что приведет к проблеме единой точки отказа. Если хакер сможет проникнуть в сеть вашей организации, он сможет взять под контроль все активы организации. Поэтому для каждой логической единицы у вас должна быть отдельная сеть. Например, среда разработки и производственная среда должны работать в разных сетях, изолированных друг от друга. Вы также можете использовать сетевые решения Zero-Trust.

Используйте менеджер паролей

Не храните учетные данные в excel. Вместо этого используйте централизованный менеджер паролей.

Image not found or type unknown



Ни при каких обстоятельствах индивидуальные пароли не должны передаваться пользователям. Лучше всего хранить учетные данные в безопасном и централизованном месте, где только необходимая команда, имеющая к нему доступ, сможет совершать вызовы API и использовать эти учетные данные.

Внедрение аудита и проверки

Вам также следует внедрить аудит и проверку на постоянной основе. Необходимо регулярно проводить аудит кода приложения, среды процессов безопасности и данных, которые оно собирает.

Заключение

Это несколько важнейших лучших практик безопасности DevOps, которым должна следовать организация для создания безопасных приложений и программного обеспечения. Внедрение практик безопасности в процесс DevOps поможет организации сэкономить миллионы. Итак, начните внедрять методы безопасности, упомянутые в этой статье, для безопасного и быстрого выпуска приложений.

Дата Создания
04.07.2023