О некоторых подходах в DevSecOps к обеспечению безопасности ПО на основе процесса тестирования

Андрей Гапон[1], Владимир Федорченко[2]

1. Кафедра безопасности информационных технологий, Харьковский национальный университет радиоэлектроники, УКРАИНА, г. Харьков, пр. Науки, 14,
E-mail: gapon.andrei@gmail.com

2. Кафедра безопасности информационных технологий, Харьковский национальный университет радиоэлектроники, УКРАИНА, г. Харьков, пр. Науки, 14,
E-mail: fedorchenko.fedor@gmail.com

Annotation: The subject of research in the article is security approaches such as test driven development and penetration testing which are used in DevSecOps and provides proactive security for program applications.

Ключевые слова: DevSecOps, Test Driven Development, Penetration testing, программное обеспечение, реактивный подход, проактивный подход, белый ящик, черный ящик, серый ящик.

I. Введение

Уязвимости программного кода обычно появляются случайно на этапе разработки и внедрения программного обеспечения. Распространенные уязвимости включают ошибки проектирования, ошибки конфигурации, ошибки программного обеспечения и т. д.

DevSecOps меняет безопасность с реактивной на проактивную, при помощи различных подходов (методов), например, таких как Test Driven Development (разработка через тестирование) и Penetration testing (тестирование на проникновение) [1].

При реактивном подходе действие не начинается, пока не произойдет инцидент. После оповещения инициируется план реагирования компании на инцидент; вначале цель состоит в том, чтобы сдержать угрозу и восстановить службу.

При проактивном подходе организации пытаются обнаружить потенциальные угрозы до того, как произойдет инцидент. Данный подход также включает в себя регулярные запланированные учения для обнаружения угроз, которые могут скрываться в системе, но еще не обнаружены или, возможно, еще не активированы [2].

Контроль безопасности и тесты должны быть внедрены на ранних этапах и повсюду в жизненном цикле разработки, и они должны проводиться в автоматическом режиме.

Анализ проникновения зависит от двух методов, а именно от оценки уязвимости и тестирования на проникновение [3].

II. Penetration testing

Тестирование на проникновение – это санкционированная попытка отдельного лица или группы использовать существующие уязвимости в технической инфраструктуре организации и всех ее компонентах, чтобы определить, возможен ли несанкционированный доступ или злонамеренная деятельность [4].

Тестирование на проникновение классифицируются на основе уровня знаний и доступа. Спектр простирается от тестирования «черного ящика», когда тестер получает минимальные знания о целевой системе, до тестирования «белого ящика», когда тестеру предоставляется высокий уровень знаний и доступа. Этот спектр знаний делает различные методологии тестирования идеальными для различных ситуаций.

Тестирование «чёрного ящика» тестеру вообще не предоставляется никакой информации. Этот сценарий лучше всего подходит для определения того, как будет действовать злоумышленник, не обладающий знаниями о структуре приложения.

Тестирование «белого ящика» включает в себя предоставление полной информации о сети и системе тестеру, включая сетевые карты и учетные данные. Это помогает сэкономить время и снизить общую стоимость задания. Тест на проникновение в «белый ящик» полезен для моделирования целевой атаки на конкретную систему с использованием максимально возможного числа векторов атаки.

Тестирование «серого ящика» являет собой предоставление только ограниченной информации. Обычно это принимает форму учетных данных для входа. Тестирование «серого ящика» полезно и помогает понять уровень доступа, который может получить привилегированный пользователь, и потенциальный ущерб, который он может причинить. Тесты «серого ящика» устанавливают баланс между глубиной и эффективностью и могут использоваться для имитации внутренней угрозы или атаки, которая нарушила периметр сети [5].

На рис.1 приведены действия, которые необходимо выполнить для выполнения теста на проникновение

Рис.1 – Этапы выполнения теста на проникновение

По итогам тестирования на проникновение формируется отчет, который описывает обнаруженные уязвимости, а также степень их критичности и рекомендации по их ликвидации. На основании отчета принимаются меры по устранению выявленных уязвимостей.

III. Test Driven Development

Разработка через тестирование (TDD) – это подход, используемый для разработки кода на основе автоматизированных тест кейсов. Тестовая разработка включает в себя:

- Добавление теста, который фиксирует концепцию программиста о желаемом функционировании небольшого фрагмента кода.
- Запуск теста, который должен завершиться неудачей, так как код не существует.
- Написание кода и выполнение теста в тесном цикле, пока тест не пройдет.
- Рефакторинг кода после прохождения теста.
- Повторение этого процесса для следующего небольшого фрагмента кода, выполнение предыдущих тестов, а также добавленных тестов [6].

На рис.2 изображен цикл разработки программного обеспечения на основе подхода разработки через тестирование.

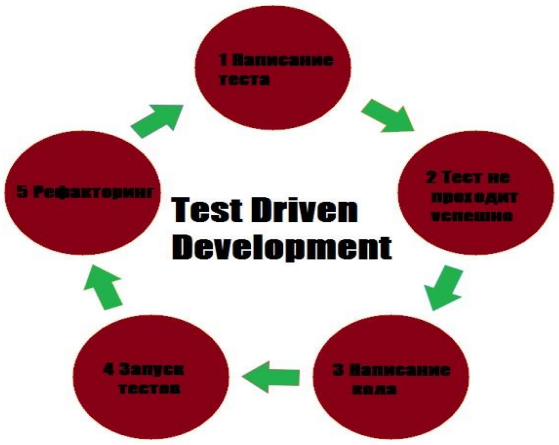

Рис.2 – Цикл разработки TDD

Разработка, основанная на тестировании, снижает вероятность появления дефектов, помогая разработчику сосредоточиться на четко определенных ожидаемых результатах. Испытания служат формой выполненной проектной спецификации для будущих работ по техническому обслуживанию. Тесты автоматизированы и используются при непрерывной интеграции.

Разработка через тестирование используется в гибких методологиях. Такой подход к разработке позволяет исправлять дефекты кодирования сразу после их появления.

Таким образом, при использовании подхода TDD, полностью предотвращается написание излишнего кода, поскольку, написание кода происходит исключительно для успешного прохождения тестов.

Выводы

Программное обеспечение и автоматизация продолжают изменять наш мир. Автоматизация в рамках жизненного цикла разработки программного обеспечения помогает нам быстрее и качественнее доставлять наш код. Добавление тестирования безопасности в эту автоматизацию также помогает создавать более безопасные приложения. DevSecOps – все еще новый подход и он развивается быстро.

В DevSecOps тестирование на проникновение должно выполняться на постоянной основе, чтобы идти в ногу с разработкой. Ручное выполнение тестов на проникновение может быть утомительной задачей, поскольку это может замедлить процесс разработки. И если это произойдет, следование принципам DevSecOps не даст никаких преимуществ.

Следовательно, существует острая необходимость в реализации автоматических тестов безопасности ПО для своевременного выявления недостатков, уязвимостей, утечки данных и лазеек.

Литература

[1] Guru99 [Електронний ресурс]. – Режим доступу : https://www.guru99.com/learn-penetration-testing.html

[2] DevSecOps Whitepaper [Електронний ресурс]. – Режим доступу : https://www.devseccon.com/wp-content/uploads/2017/07/DevSecOps-whitepaper.pdf

[3] DevOps [Електронний ресурс]. – Режим доступу : https://devops.com/shifting-data-protection-paradigm-proactive-vs-reactive/

[4] Breachlock [Електронний ресурс]. – Режим доступу : https://www.breachlock.com/penetration-testing-and-devops/

[5] Redscan [Електронний ресурс]. – Режим доступу : https://www.redscan.com/news/types-of-pen-testing-white-box-black-box-and-everything-in-between/

[6] *ISTQB* (*International Software Testing Qualifications Board*) [Електронний ресурс]. – Режим доступу : https://www.istqb.org/news/news/2014/189-in-chapter-3-1-1-in-agile-tester-extension-test-driven-development.html