функции часто ограничивают число полуоткрытых каналов в любой момент времени), ограничение объема трафика (traffic rate limiting).

Проанализировав самые распространенные вида атаки, можно сделать следующие выводы: не стоит использовать уязвимые протоколы, для защиты корпоративных сетей следует использовать технологии корпоративного класса, MFP, необходимо контролировать радиосреду на предмет наличия DoS атак , необходима правильная настройка оборудования и клиентских ПК, автоматизированные средства распространения обновления для ПО и антивирусов, принудительное использование VPN при работе вне корпоративной беспроводной сети.

Литература

1. Боршевников, А. Е. Сетевые атаки. Виды. Способы борьбы / Современные тенденции технических наук: материалы I Международной научной конференции. – 2011. – С. 8–13.

2. Парасрам Шива, Хериянто Теди, Замм Алекс и др. Kali Linux. Тестирование на проникновение и безопасность. / Kali Linux, 2020.

3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. / Питер, 2016. 5-е изд.

К. В. Кислова
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **В. В. Грищенко**, ст. преподаватель

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ СБОРКИ, ТЕСТИРОВАНИЯ И РАЗВЕРТЫВАНИЯ ПРИЛОЖЕНИЯ НА ОСНОВЕ СОВРЕМЕННЫХ ОБЛАЧНЫХ РЕШЕНИЙ

В последние несколько десятилетий стремительно возросла роль компьютерных наук и технологий. С развитием данной отрасли каждое предприятие старается не отставать от современных решений и внедрять их в свои продукты и разработки. Такой темп развития в бизнесе задает высокую конкуренцию на рынке, что заставляет компании предпринимать действия для улучшения качества и скорости реализации их программного обеспечения потребителям.

Решением данных проблем являются технологии автоматизации и облачные решения.

Благодаря автоматизации процессов сборки, тестирования и развертывания компания имеет возможность поставлять свое программное обеспечение в течение малого промежутка времени, независимо от размера самого проекта, а также его кодовой базы.

Составляющими такого подхода поддержки программного обеспечения напрямую относятся к методологии DevOps. Данная методология основана на тесном взаимодействии специалистов по разработке и специалистов по техническому обслуживанию, а также взаимной интеграции их рабочих процессов.

В качестве кодовой базы программного обеспечения используется система контроля версий, которая позволяет специалистам по разработке добавлять, обновлять, удалять и объединять фрагменты исходного кода продукта. Такой удобной системой является платформа Git. Для обеспечения жизненного цикла методологии DevOps подходящей будет система управления репозиториями кода для Git, с возможностью отслеживания ошибок и CI/CD конвейером, платформа GitLab.

Облачные решения являются крупной составляющей информационных технологий. Подавляющее большинство компаний пользуются услугами облачных провайдером для оптимизации своей информационной инфраструктуры. Такие гиганты как Amazon AWS, Microsoft Azure, Google Cloud и другие предоставляют широкий спектр ресурсов и услуг для поддержки программного обеспечения.

Для реализации инфраструктуры на облачной платформе есть два подхода: настройка вручную и концепция «инфраструктура как код». Первый подход не является лучшим на уровне крупных проектов и задач, по причине того, что является ненадежным и может затрачивать много времени на конфигурацию, поддержку и поиск ошибок в дальнейшем.

Подход «инфраструктура как код» считается более оптимальным, потому что имеет ряд достоинств: конфигурация ресурсов инфраструктуры является сразу же и документацией к проекту, возможно неоднократное применение конфигурации, приводящее к одному результату, а также устраняется человеческий фактор. В качестве инструмента «инфраструктура как код» подходящим является Terraform.

В итоге мы получаем приложение, которое размещается на базе облачной платформы, с доступом к обслуживанию его ресурсов, а также способностью внесения обновлений его составляющих в короткие сроки.