POSITIVE TECH PRESS CLUB

Безопасная разработка: как жить в цифре, писать много кода и не стать жертвой хакеров

ptsecurity.com

Последствия уязвимостей в исходном коде

Недостатки в исходном коде становятся причиной хакерских атак. Только за последнее время мы стали свидетелями нескольких громких инцидентов. Например, уязвимости в системе обработки багажа аэропорта Хитроу привели к потере 42 000 чемоданов и отмене 500 авиарейсов в течение 10 дней после открытия нового терминала. А недостаточно протестированное ПО для высокочастотной торговли за полчаса спустило на бирже 440 млн $ — почти весь капитал американского брокера Knight Capital Group. Уязвимости в системе удаленного управления и мониторинга IT-инфраструктуры Kaseya VSA позволили злоумышленникам распространить вредоносное ПО (ВПО) в сети порядка 1500 клиентов Kaseya. Новая группировка вымогателей, которая называет себя Epsilon Red, начала использовать уязвимости сервера Microsoft Exchange (ProxyLogon) для шифрования компьютеров в сети. Есть и свежие примеры эксплуатации уязвимостей в ПО для удаленного подключения, что особенно актуально в период сохранения дистанционного режима работы сотрудников во многих компаниях: специалисты ИБ компании FireEye сообщили, что как минимум две хакерские группировки эксплуатируют уязвимости в Pulse Secure VPN для атак на оборонные, правительственные и финансовые организации в США и других странах. В общей сложности эксперты идентифицировали 12 семейств вредоносного ПО, связанных с атаками на Pulse Secure VPN.

Сколько атак можно было бы предотвратить, не будь брешей в исходном коде

Злоумышленники могут эксплуатировать уязвимости на разных этапах атаки; при этом стоит разделять эксплуатацию уязвимостей в веб-приложениях, программных и аппаратных продуктах (хакинг), недостатки защиты или неправильной конфигурации. Исследования Positive Technologies показывают, что в последние два года хакинг набирает обороты: доля атак за второй квартал 2021 года увеличилась до 30%, продемонстрировав рост на 4% относительно предыдущего квартала. При этом с начала 2021 года число атак с использованием веб-уязвимостей снизилось и сейчас держится на уровне прошлого года (10%).

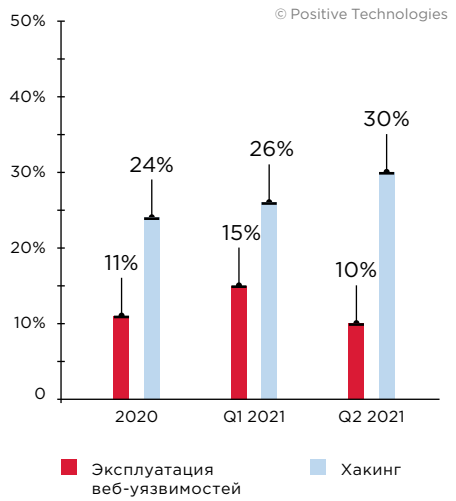

Рисунок 1. Доля кибератак с использованием веб-уязвимостей и хакинга. Статистика Positive Technologies

В первом полугодии 2021 года злоумышленники в качестве объектов атаки чаще всего выбирали компьютеры, серверы и сетевое оборудование (78% случаев), а также веб-ресурсы (16%).

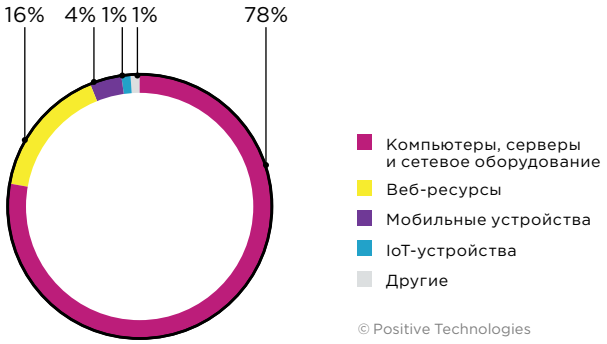

Рисунок 2. Объекты атак (доля атак). Исследование Positive Technologies «Актуальные киберугрозы. Первое полугодие 2021 года»

Хакеры используют уязвимости на разных этапах атаки и для достижения разных целей:

- Для получения первоначального доступа в инфраструктуру — для этого они эксплуатируют уязвимости в сервисах, доступных из интернета. Это могут быть недостатки в коде веб-сайтов, баз данных, сетевых служб и протоколов (например, SMB или SSH), протоколов администрирования и управления сетевыми устройствами (например, SNMP и Smart Install). Так, по данным наших внешних пентестов за 2020 год, уязвимости в исходном коде веб-приложений позволяют проникнуть во внутреннюю сеть 50% компаний.
- Для повышения привилегий в системе — для этого хакеры эксплуатируют уязвимости программного обеспечения. Например, уязвимости в Windows Server.
- Для обхода функций безопасности — злоумышленники используют уязвимости в самих продуктах, предназначенных для защиты. Например, ошибки в межсетевых экранах Cisco ASA и Fortinet FortiWeb, обнаруженные нашими исследователями безопасности.
- При перемещениях внутри сети и применении техник lateral movement (перемещение внутри периметра). Например, использование уязвимости Zerologon в Windows Server.

В каких отраслях наличие уязвимостей представляет наибольшую опасность

На данный момент аккумулированная статистика не позволяет определить самые слабые отрасли в части безопасности кода. Тем не менее можно выделить сферы, находящиеся в зоне риска, которым стоит уделять повышенное внимание защите исходного кода и устранять уязвимости еще на этапах его создания.

- Наибольшую опасность представляют уязвимости в программном и аппаратном обеспечении промышленных систем. Злоумышленники, используя недостатки этих систем, могут нарушить технологические процессы предприятия или вызвать техногенные катастрофы, которые повлекут за собой человеческие жертвы. За последние годы мы видим рост числа уязвимостей в сетевом промышленном оборудовании: в коммутаторах, конвертерах интерфейсов, шлюзах. В 2020 году, согласно отчету компании Claroty, специализирующейся на промышленной кибербезопасности, в АСУ ТП было выявлено на 24,7% больше уязвимостей, чем годом ранее.
- В связи с набирающим обороты трендом supply chain attack (атака на цепь поставок) поставщикам ИБ и ИТ-решений крайне важно обеспечивать не только собственную безопасность, но также и безопасность своих продуктов — атаки, реализованные посредством внедрения ВПО в программное обеспечение, могут затронуть компании любой отрасли.

DevSecOps и безопасность компаний. Степень проникновения подхода в российский бизнес

Алексей Жуков, эксперт отдела систем защиты приложений Positive Technologies

Тимур Гильмуллин, руководитель направления по построению безопасной разработки центра исследований и разработки Positive Technologies

Какое место DevSecOps занимает в безопасности компаний в целом?

Алексей Жуков: «Место DevSecOps в безопасности компаний зависит от профиля бизнеса и особенностей приложений.

У любой организации есть периметр и системы внутри него. Когда злоумышленники проникают в инфраструктуру и перемещаются от одного узла к другому, они встречают различные приложения. Эти приложения могут быть написаны как компаниями самостоятельно, так и сторонними организациями — подрядчиками или вендорами. Последние сами устраняют недостатки защищенности своих продуктов (оставим пока вопрос своевременного развертывания таких обновлений), а вот с безопасностью кода самописных приложений или приложений, написанных подрядчиками, дела обстоят хуже.

Место DevSecOps в бизнесе зависит и от особенностей самих систем. В любой компании есть приложения, в которых эксплуатация уязвимостей является недопустимым с точки зрения бизнеса событием. Если такие приложения становятся доступны хакеру (независимо от того, доступны они непосредственно из внешней сети или же для их взлома потребуется получить контроль над промежуточными элементами инфраструктуры), то в конечном итоге их взлом приведет к опасным последствиям для жертвы. В этом случае безопасность кода является очевидным

приоритетом для бизнеса. Кроме того, даже если взломанное приложение не представляет для злоумышленника прямого интереса, оно вполне может стать платформой для проведения последующих атак на приложения других компаний, что также может повлечь негативные последствия для бизнеса.

Такая классификация может стать хорошим подспорьем в расстановке приоритетов в постепенном выстраивании DevSecOps: вне зависимости от архитектуры приложения в нем всегда можно выделить некую часть, стабильность и надежность которой максимально опасны. Такое понимание — хороший старт для реализации безопасной разработки и адаптации существующих процессов таким образом, чтобы закрывать уязвимости, начиная с самых ранних этапов создания кода.

Плюсы этого подхода очевидны: чем раньше будет выявлен дефект, тем дешевле обойдется его исправление. Причем это утверждение справедливо независимо от того, какую модель разработки применяет команда. Разница в стоимости исправления ошибок между этапами написания кода и эксплуатации ПО такова: выявление и устранение уязвимостей, начиная с самых ранних этапов разработки, становится еще и способом сокращения затрат на разработку».

Тимур Гильмуллин: «Комплексный подход при разработке приложений актуален как никогда. Разработчики регулярно узнают о багах в своих приложениях из самых разных источников: от пользователей, специалистов по ИБ и из отчетов по результатам внешних пентестов. Правда, как показывает практика, при разработке новых фич продукта программисты часто не задумываются над безопасностью кода. Инженеры по эксплуатации также редко контролируют уязвимые компоненты тех приложений, которые им приходится развертывать на внешних серверах. Наглядный пример — возросшее за последний год количество атак на цепочку поставок.

Многих проблем с безопасностью кода можно избежать еще на стадии разработки, если этому уделять внимание. Например, использовать сканеры уязвимостей кода, интегрированные в конвейер разработки продуктов, или так называемые blackbox-сканеры в режиме внешнего пентеста для проверки веб-приложений еще на стадии тестирования внутри компании.

Однако при внедрении любых средств защиты важно найти баланс между безопасностью и уже сложившимся процессом разработки. Логично организовать security test'ы на базе конвейера поверх уже налаженного DevOps-инженерами, не нарушая, а дополняя его процессы».

DevSecOps — прерогатива разработчиков или служб по ИБ?

Алексей Жуков: «Если говорить о продуктовых компаниях, для которых DevSecOps — это ключевой бизнес, то мы видим, что сегодня безопасная разработка среди них больше интересует CTO и технических специалистов, а не специалистов по ИБ. CTO и технические специалисты понимают, чем опасны уязвимости в коде приложений. Именно они используют инструменты DevSecOps, следят за процентом исправленных уязвимостей, у них есть понимание, что внедрять процессы безопасной разработки в производственный цикл необходимо и это реалии нашего времени. Когда компании разрабатывают приложения, которыми будут пользоваться клиенты, вопрос о написании защищенного кода не стоит. Все понимают, что это необходимо, и это очень хороший знак.

Если же рассматривать enterprise-сегмент и внутренние приложения, сервисы и порталы, которые разрабатывают крупные компании самостоятельно, то тут картина менее радужная. Представители этого сегмента пока ошибочно полагают, что проблема связана со специалистами по ИБ, которые в свою очередь в коде не разбираются, и, как следствие, решают ее формально: получают сертификаты, соблюдают требования регулятора. Такое отношение к разработке внутреннего ПО значительно упрощает задачу злоумышленникам: они смогут не только легко проникнуть в инфраструктуру компании, но и закрепиться там для проведения атак».

Реально ли хотеть от безопасности 100%-й защиты, особенно если эта защита нужна в приложении? Как в этом поможет DevSecOps?

Самое главное — выстроить архитектуру приложений компании таким образом, чтобы в случае взлома одного уязвимого приложения злоумышленники не смогли добраться до целевой системы и через нее реализовать недопустимое для компании событие

Алексей Жуков: «Любую компанию можно взломать, и это надо принять как данность. Например, в рамках внешних пентестов в 2020 году нашим исследователям удалось получить доступ к локальным сетям 93% организаций. Рано или поздно уязвимости можно пропустить, а возможно, даже и не заметить, как злоумышленники начнут их эксплуатировать. Поэтому самое главное — выстроить архитектуру приложений компании таким образом, чтобы в случае взлома одного уязвимого приложения злоумышленники не смогли добраться до целевой системы и через нее реализовать недопустимое для компании событие. Конечно же, важно удлинять цепочку потенциальной атаки: например, после сервиса с уязвимостями в коде устанавливать дополнительные системы верификации, микросервисы и прочее. Если выходить за рамки стандартной парадигмы и перестраивать инфраструктуру, можно приблизиться к 100%-й защите. Также не стоит исключать из этого процесса специалистов центров реагирования на киберугрозы (security operations center, SOC), которые постоянно мониторят инфраструктуру и обеспечивают сетевую безопасность. Таким образом, удлинение цепочки, умноженное на сниженную вероятность взлома, а также участие специалистов по предотвращению киберинцидентов позволит избежать недопустимого события. Каждый из перечисленных компонентов по отдельности не дает 100%-й защиты, но в совокупности все это обеспечит максимальную защищенность, так как на каждом из последующих звеньев снижается вероятность взлома компании».

Как реализовать безопасную разработку с точки зрения бизнеса? Какие составляющие для этого нужны?

Тимур Гильмуллин: «Чтобы реализовать DevSecOps, необходимо сначала принять, что идеальных технических средств нет. Важно максимально использовать все имеющиеся возможности: привлекать пентестеров, проводить киберучения, организовывать на постоянной основе внутреннее обучение сотрудников безопасной разработке, проводить для них воркшопы и вебинары.

Так, сканеры уязвимостей умеют хорошо обнаруживать технические проблемы в коде, а пентестеры — находить уязвимости другого рода, например функциональные, в бизнес-логике или несоответствии API. Киберучения помогают сканерам искать уязвимости под конкретные сценарии. Любой продукт позволяет написать правила, которые можно использовать как паттерны для этих сценариев. Обучения, воркшопы и вебинары, в свою очередь, помогают получить и закрепить практические навыки безопасной разработки.

Важно также понимать, что методология DevSecOps подразумевает безопасную разработку на всех технологических этапах и шагах производственного CI/CD-конвейера[1]. При этом security-процесс и контроль безопасности должны происходить параллельно и непрерывно, как и разработка, развертывание, тестирование и доставка. А инструменты, обеспечивающие безопасность, рекомендуется внедрять на каждом этапе производственного конвейера.

В частности, все изменения в коде продукта должны проверяться на наличие потенциальных уязвимостей и ошибок, а значит, сканеры кода должны быть интегрированы в продуктовый конвейер непосредственно на этапе разработки и сборки компонентов продукта. Разработчики должны понимать отчеты сканера, оценивать опасность найденных уязвимостей и уметь их устранять.

Понятно, что добиться такой интеграции без тесного сотрудничества специалистов по ИБ (Sec), разработчиков (Dev) и инженеров по инфраструктуре и эксплуатации (Ops) практически невозможно. Даже сама суть термина DevSecOps (или SecDevOps) подразумевает сотрудничество специалистов разных направлений».

Как найти баланс между безопасностью и влиянием на time to market? За кем финальное решение — скорый выход продукта на рынок или ИБ?

Ключевое условие DevSecOps — все играют по одним правилам

Алексей Жуков: «Ключевое условие DevSecOps — все играют по одним правилам. Результат будет в том случае, когда все понимают, что безопасная разработка необходима, когда не нужно решать организационные вопросы и лишний раз доказывать, что DevSecOps необходим. Благодаря этому мы легко выстраиваем процессы безопасной разработки в компаниях и быстро реализуем проекты.

Например, распространенные методики по выстраиванию процессов безопасной разработки включают в себя большое число мероприятий организационного характера. Очевидно, что, не заинтересовав всех участников, не заручившись поддержкой руководства, мы рискуем в лучшем случае получить «итальянскую забастовку». Для нас одним из примеров почти идеального проекта по выстраиванию безопасной разработки стал тот, в котором все эти организационные вопросы были решены, а роли всех участников были четко распределены. И несмотря на то, что в начале проекта далеко не все разработчики понимали, чем опасна та или иная уязвимость, именно такая командная работа стала залогом успеха проекта в целом».

Тимур Гильмуллин: «В компаниях, которые занимаются разработкой, задача по анализу кода часто стоит перед службой ИБ. Но передать это только специалистам по ИБ нереально, так как в современной разработке используются различные языки программирования и технологии и кода всегда написано много. Например, у нашей компании, разрабатывающей больше десятка решений в области ИБ, количество сборок различных компонентов при каждом изменении кода — почти 3 миллиона в год! То есть автоматизация процессов безопасной разработки совершенно необходима. Но финальное решение о внедрении DevSecOps всегда принимает бизнес.

1 Непрерывная интеграция (Continuous Integration, CI) и непрерывная поставка (Continuous Delivery, CD).

Концепцию непрерывного и безопасного процесса разработки можно развивать только при условии заинтересованности и инициативы самих сотрудников. Если руководство закупит какие-то инструменты и "спустит их сверху вниз" — ничего не выйдет. Нужно, чтобы разработчики были заинтересованы в таких инструментах, чтобы эти решения их устраивали и были удобны в использовании. Например, сканер кода, внедренный в конвейер разработки, не должен мешать программистам разрабатывать фичи, а наоборот, должен работать параллельно с основными процессами сборки и при этом блокировать выпуск релиза, если будет найден уязвимый код. Требуется разумный инженерный подход, чтобы не увеличивать время time to market и объемы работ на дополнительную безопасность».

С какими сложностями сталкиваются компании при построении безопасной разработки?

Тимур Гильмуллин: «В целом сложности с внедрением идей DevSecOps можно разделить на три направления:

1. Организационные трудности, когда менеджмент не желает внедрять DevSecOps, — руководители (ошибочно!) считают, что устранять не влияющие на функциональность приложений недостатки кода нет смысла, что такие уязвимости серьезной опасности не представляют.
2. Не хватает квалификации программистов. Команда разработки недооценивает опасность уязвимостей и не понимает, почему служба ИБ просит их устранить.
3. Технологические сложности, связанные с внедрением любых инструментов безопасной разработки в продуктовый CI/CD-конвейер.

В последнем случае у разработчиков ПО, как правило, уже есть выстроенный процесс разработки и CI/CD-конвейер с десятками этапов и сотнями шагов. В общем виде он начинается с коммита строчки кода в git-репозиторий; затем выполняется отправка кода на сборку; собранные компоненты или приложение развертываются на тестовом стенде; выполняется ручное или автоматизированное тестирование; затем сборка продвигается в релизное хранилище и оттуда попадает на внешние серверы доставки продукта и обновлений. У CI-инженеров и инженеров по инфраструктуре DevOps сразу возникают конкретные технические вопросы по любому инструменту, который нужно добавить в этот отлаженный механизм: есть ли методика его внедрения, на каком этапе его необходимо использовать, какие у него ограничения и совместимость с текущей CI-системой, какова архитектура предлагаемого решения, насколько инструмент сложен в поддержке. И это не считая того, что компании потребуется обучить разработчиков использованию каждого инструмента».

Фундаментом успешного выстраивания DevSecOps является формирование сообщества заинтересованных разработчиков

Алексей Жуков: «Построение безопасной разработки не сводится к одному только приобретению и встраиванию соответствующего инструментария в инфраструктуру. Да, конечно, эта тема очень важна, да, в ходе внедрения потребуется решить множество технических задач, но фундаментом успешного выстраивания процесса все-таки является формирование сообщества заинтересованных в DevSecOps разработчиков. Этот путь компании уже проходили, когда учились работать в парадигме DevOps, организуя взаимодействие между различными подразделениями, ответственными за разработку и эксплуатацию ПО. Выстраивание безопасной разработки становится логичным продолжением этого пути, когда у разработчиков формируется понимание того, что одна-единственная уязвимость в приложении может решить судьбу всего бизнеса; это приводит к качественному скачку, и в результате ИБ перестают воспринимать как подразделение, которое вставляет палки в колеса».

На какие метрики эффективности нужно ориентироваться при построении безопасной разработки?

Тимур Гильмуллин: «Метрики эффективности можно разделить на два вида:

- технические — они появляются после внедрения анализа кода. Например, можно измерять количество уязвимостей или процент уменьшения их числа в проекте от релиза к релизу, промежуток времени на внесение патчей, прохождение или непрохождение условий так называемых quality gates[2] и security gates[3], а также количество проектов, подключенных к анализу. Главное, чтобы метрики были измеримыми, максимально достоверными и понятными большинству специалистов;
- организационные — появление в команде разработки специалиста, отвечающего за проверку кода на безопасность, или назначение ответственного руководителя, комплексно реализовывающего проекты по DevSecOps».

Алексей Жуков: «Можно использовать любые метрики эффективности, такие как число найденных уязвимостей, средний срок исправления, количество релизов с исправлениями багов или финальное число инцидентов. Но при определении этих метрик важно не замкнуться внутри себя: только реальные кейсы лучше всего покажут, насколько правильно и эффективно построен процесс безопасной разработки. Так как возможности ждать, когда компанию попытаются взломать злоумышленники, нет, необходимо либо участвовать в киберучениях, либо проводить тесты на проникновение в IT-инфраструктуру, сервисы и приложения.

DevSecOps — это часть общей стратегии безопасности, и если на киберучениях команды атакующих не могут найти уязвимости в исходном коде, значит, безопасная разработка в компании выстроена верно и метрики эффективности используются правильные».

2 Измерение качества кода.

3 Измерение безопасности кода.

Positive Tech Press Club_Whitepaper_Secure-Development_A4.RUS.0004.02.SEP.01.2021