Лекция 13: Основы облачной безопасности

Тема: Облачная безопасность и защита инфраструктуры

Технический университет Молдовы

Лектор: Максим Масютин

Введение

Здравствуйте. Добро пожаловать на Лекцию 13, в которой мы рассмотрим одну из наиболее актуальных тем современной кибербезопасности: основы облачной безопасности. В течение следующих двух часов мы изучим, как организации обеспечивают безопасность своих облачных сред, какие уникальные проблемы создают облачные вычисления, а также какие фреймворки и инструменты помогают нам защищать данные и приложения в облаке.

Позвольте начать с вопроса: сколько из вас уже воспользовались облачным сервисом сегодня? Если вы проверяли электронную почту, смотрели видео или работали над документом совместно с другими, вы уже взаимодействовали с облачной инфраструктурой. Облако — это уже не будущее, а настоящее. Согласно Gartner, исследовательской компании в сфере технологий, более 94% предприятий используют облачные сервисы в той или иной форме, а глобальные расходы на облачные технологии превысили 700 миллиардов долларов в год.

Однако вместе с массовым внедрением приходят столь же масштабные проблемы безопасности. Согласно отчёту IBM Cost of a Data Breach за 2024 год, количество утечек данных в облачных средах увеличилось на 35% за последние два года, а неправильно сконфигурированные облачные ресурсы остаются причиной номер один инцидентов облачной безопасности. Данная лекция даст вам знания для понимания этих проблем и противодействия им.

Часть 1: Модели облачных сервисов — IaaS, PaaS, SaaS

Понимание облачного стека

Прежде чем мы сможем обеспечить безопасность облака, необходимо понять, что такое облако на самом деле. Облачные вычисления предоставляют вычислительные услуги через интернет, обеспечивая доступ к ресурсам по запросу без необходимости владеть физической инфраструктурой. Национальный институт стандартов и технологий (NIST) определяет три основные модели обслуживания: Инфраструктура как услуга (IaaS), Платформа как услуга (PaaS) и Программное обеспечение как услуга (SaaS).

Инфраструктура как услуга (IaaS)

Начнём с нижнего уровня стека — Инфраструктуры как услуги. IaaS предоставляет виртуализированные вычислительные ресурсы через интернет. Представьте это как аренду необработанной вычислительной мощности, хранилища и сети. Облачный провайдер управляет физическими центрами обработки данных, серверами и сетевым оборудованием. Вы, как заказчик, управляете всем остальным: операционными системами, промежуточным программным обеспечением, средами выполнения, данными и приложениями.

Примеры IaaS включают Amazon Web Services EC2, Microsoft Azure Virtual Machines и Google Compute Engine. Когда вы запускаете виртуальную машину в AWS, вы используете IaaS. Вы получаете пустой сервер с операционной системой, и вы несёте ответственность за его обновление, обеспечение безопасности и управление всем, что на нём работает.

С точки зрения безопасности, IaaS даёт вам наибольший контроль, но и наибольшую ответственность. Вы должны настраивать межсетевые экраны, управлять доступом, шифровать данные и обеспечивать обновление операционных систем. Типичная ошибка организаций — обращаться с облачными виртуальными машинами как с физическими серверами в центре обработки данных. Они забывают, что эти виртуальные машины доступны из интернета, и часто оставляют конфигурации по умолчанию.

Платформа как услуга (PaaS)

Поднимаясь по стеку, мы переходим к Платформе как услуге. PaaS предоставляет платформу для разработчиков, позволяющую создавать, развёртывать и управлять приложениями без необходимости заботиться о базовой инфраструктуре. Облачный провайдер управляет не только физической

инфраструктурой, но и операционными системами, промежуточным программным обеспечением и средами выполнения.

Примеры включают AWS Elastic Beanstalk, Azure App Service, Google App Engine и Heroku. Когда команда разработчиков развёртывает приложение в Azure App Service, она концентрируется на своём коде. Azure обеспечивает выделение серверов, балансировку нагрузки и обновления платформы.

Безопасность в PaaS в большей степени разделяется с провайдером. Вы по-прежнему обеспечиваете безопасность кода вашего приложения, ваших данных и управления идентификацией, но безопасность платформы обеспечивается провайдером. Однако это не означает, что вы можете игнорировать безопасность. Уязвимости на уровне приложений, небезопасные API и неправильная настройка управления доступом по-прежнему остаются вашей ответственностью.

Программное обеспечение как услуга (SaaS)

На вершине стека находится Программное обеспечение как услуга. SaaS предоставляет готовые приложения через интернет. Пользователи получают доступ к программному обеспечению через веб-браузер без локальной установки. Провайдер управляет всем: инфраструктурой, платформой и приложением.

Вы пользуетесь SaaS каждый день: Microsoft 365, Google Workspace, Salesforce, Dropbox, Slack. Когда вы составляете электронное письмо в Gmail, Google управляет серверами, почтовым приложением и безопасностью платформы.

С точки зрения безопасности, SaaS, казалось бы, перекладывает основную часть ответственности на провайдера. Однако заказчики по-прежнему управляют критически важными аспектами: доступом пользователей, классификацией данных и параметрами конфигурации. Многие утечки в SaaS происходят потому, что организации не настраивают должным образом свои SaaS-приложения. Они оставляют слишком широкие права на общий доступ, не включают многофакторную аутентификацию или предоставляют пользователям избыточные привилегии.

Часть 2: Модель разделённой ответственности

Определение ответственности

Теперь, когда мы разобрались с моделями обслуживания, необходимо обсудить наиболее важную концепцию облачной безопасности: модель разделённой ответственности. Эта модель определяет, кто за что отвечает в облачной среде.

Вот основополагающий принцип: безопасность САМОГО облака — ответственность провайдера; безопасность В облаке — ваша ответственность. Провайдер обеспечивает безопасность инфраструктуры, центров обработки данных, гипервизоров и сети. Вы обеспечиваете безопасность ваших данных, приложений, конфигураций и управления доступом.

Распределение ответственности по моделям обслуживания

Распределение ответственности варьируется в зависимости от модели обслуживания. В IaaS заказчик несёт наибольшую ответственность. Вы управляете операционными системами, сетевыми средствами управления, приложениями, управлением идентификацией и шифрованием данных. Провайдер обеспечивает физическую безопасность, безопасность гипервизора и сетевую инфраструктуру.

В PaaS провайдер берёт на себя больше ответственности. Он управляет операционными системами и средами выполнения. Вы сосредотачиваетесь на приложениях, данных и управлении доступом.

В SaaS провайдер управляет практически всем. Ваша ответственность сводится к управлению данными, доступу пользователей и настройке приложений.

Распространённые заблуждения

Позвольте привести историю, иллюстрирующую, почему это важно. В 2019 году компания Capital One подверглась масштабной утечке, затронувшей более 100 миллионов клиентов. Утечка произошла из-за неправильной настройки межсетевого экрана уровня приложений в их среде AWS. Capital One полагала, что AWS обеспечит всю безопасность, но ответственность AWS ограничивалась защитой базовой инфраструктуры. Неправильная конфигурация была ответственностью Capital One в рамках модели разделённой ответственности.

Та же ситуация повторилась в 2024 году, когда несколько крупных медицинских организаций пострадали от утечек через неправильно сконфигурированные облачные хранилища. Эти организации полагали, что их облачные провайдеры предотвратят несанкционированный доступ к данным. Они ошибались. Настройка управления доступом — ответственность заказчика.

Практическое применение

При проектировании облачной безопасности всегда задавайте вопрос: кто отвечает за данную меру безопасности? Документируйте ответственность в явном виде. Создайте матрицу ответственности, которая отображает каждую функцию

безопасности на провайдера, заказчика или обоих. Регулярно пересматривайте эту матрицу, особенно при внедрении новых облачных сервисов.

Часть 3: Руководства Cloud Security Alliance

Введение в CSA

Cloud Security Alliance, или CSA, — ведущая организация, определяющая лучшие практики облачной безопасности. Основанная в 2008 году, CSA стала авторитетным источником стандартов и руководств в области облачной безопасности. Их работа влияет на то, как организации по всему миру подходят к облачной безопасности.

Руководство CSA по безопасности

Руководство CSA по безопасности (CSA Security Guidance) — их основополагающий документ, в настоящее время в четвёртой основной версии. Это комплексное руководство охватывает 14 областей облачной безопасности:

1. Облачная архитектура и управление
2. Управление, управление рисками и соответствие
3. Правовые вопросы, договоры и электронное обнаружение
4. Управление соответствием и аудит
5. Управление информацией
6. Плоскость управления и непрерывность бизнеса
7. Безопасность инфраструктуры
8. Виртуализация и контейнеры
9. Реагирование на инциденты
10. Безопасность приложений
11. Безопасность данных и шифрование
12. Управление идентификацией, правами и доступом
13. Безопасность как услуга
14. Смежные технологии

Каждая область содержит подробные рекомендации по обеспечению безопасности облачных сред. Я настоятельно рекомендую внимательно прочитать этот документ — он находится в свободном доступе на веб-сайте CSA.

Матрица облачных контролей

Ещё один важнейший ресурс CSA — Матрица облачных контролей, или CCM. CCM — это фреймворк контролей кибербезопасности, специально разработанный для облачных вычислений. Он отображает контроли на основные фреймворки соответствия, включая ISO 27001, NIST, PCI DSS и GDPR.

CCM содержит более 180 целей контроля, организованных в 17 областей. Каждый контроль указывает, лежит ли ответственность на облачном провайдере, заказчике или обоих. Это делает CCM незаменимым для оценки состояния облачной безопасности и обеспечения соответствия.

CAIQ и реестр STAR

CSA также предоставляет Опросник Консенсусной Оценки (Consensus Assessments Initiative Questionnaire, CAIQ). Этот опросник помогает организациям оценивать безопасность облачных провайдеров. Он содержит более 250 вопросов, основанных на контролях CCM.

Реестр STAR — это публичная база данных, где облачные провайдеры публикуют свои оценки безопасности на основе CAIQ. Прежде чем выбрать облачного провайдера, вам следует ознакомиться с его сертификацией STAR. Основные провайдеры, такие как AWS, Azure и Google Cloud, поддерживают сертификации STAR.

Недавние инициативы CSA

В 2024 и 2025 годах CSA расширила свою деятельность, чтобы охватить новые вызовы. Организация выпустила руководства по безопасности ИИ в облачных средах, архитектурам нулевого доверия для облака и подготовке к квантовоустойчивой криптографии. Отчёт CSA по главным угрозам, обновляемый ежегодно, теперь включает риски, связанные с ИИ, наряду с традиционными проблемами облачной безопасности, такими как небезопасные интерфейсы, неправильная конфигурация и недостаточная видимость.

Часть 4: Управление идентификацией и доступом в облачных средах

Основа облачной безопасности

Управление идентификацией и доступом, или IAM, является основой облачной безопасности. Если злоумышленники могут скомпрометировать учётные данные,

они могут получить доступ к вашим данным и системам, не эксплуатируя технические уязвимости. IAM отвечает на два фундаментальных вопроса: кто вы и что вам разрешено делать?

Архитектура облачного IAM

В облачных средах IAM работает на нескольких уровнях. Во-первых, существует система IAM облачного провайдера: AWS IAM, Azure Active Directory, Google Cloud IAM. Эти системы контролируют доступ к облачным ресурсам — кто может создавать виртуальные машины, читать хранилища или изменять сетевые конфигурации.

Во-вторых, существует IAM уровня приложений. Ваши приложения, работающие в облаке, имеют собственные каталоги пользователей и средства управления доступом. Они могут быть интегрированы с облачным IAM или работать независимо.

В-третьих, существует федерация идентификации. Организации подключают свои локальные каталоги, такие как Active Directory, к облачным системам IAM. Это обеспечивает единую точку входа и централизованное управление идентификацией в гибридных средах.

Ключевые принципы IAM

Минимальные привилегии — важнейший принцип IAM. Пользователи и сервисы должны иметь только минимальные разрешения, необходимые для выполнения их функций. В облачных средах это означает отказ от использования учётных записей root, создание конкретных ролей для конкретных задач и регулярный пересмотр и отзыв ненужных разрешений.

Доступ точно в срок набирает популярность. Вместо предоставления постоянных привилегий доступ предоставляется временно, когда он необходим. AWS предлагает IAM Access Analyzer, Azure — Privileged Identity Management, а Google Cloud — IAM Recommender. Эти инструменты помогают выявлять и сокращать избыточные разрешения.

Сервисные учётные записи требуют особого внимания. Многие облачные утечки происходят через скомпрометированные сервисные учётные записи с избыточными разрешениями. Эти учётные записи часто имеют постоянные привилегии, а их учётные данные хранятся в репозиториях кода или файлах конфигурации. Используйте управляемые идентификации везде, где это возможно — AWS IAM Roles, Azure Managed Identities, Google Cloud Service Accounts с Workload Identity.

Многофакторная аутентификация

Многофакторная аутентификация является обязательным требованием в облачных средах. Каждый пользователь должен проходить аутентификацию с использованием как минимум двух факторов. Облачные провайдеры теперь предлагают варианты беспарольной аутентификации с использованием аппаратных ключей безопасности FIDO2 (Fast IDentity Online 2, стандарт быстрой онлайн-идентификации) или биометрической аутентификации.

Для привилегированных учётных записей — администраторов, инженеров DevOps, сотрудников службы безопасности — рекомендуется рассмотреть аппаратные ключи безопасности. Программные приложения MFA уязвимы для фишинговых атак; аппаратные ключи — нет.

Управление идентификацией

Помимо технических контролей, облачный IAM требует управления. Внедряйте регулярные проверки доступа. Автоматизируйте создание и деактивацию учётных записей пользователей с помощью управления жизненным циклом идентификации. Отслеживайте аномальные модели аутентификации с использованием облачных инструментов или сторонних систем управления информацией и событиями безопасности (SIEM).

Отчёт Verizon об исследовании утечек данных за 2025 год показал, что скомпрометированные учётные данные были задействованы в 44% утечек. Надёжное управление идентификацией и доступом — это не опция, а необходимость.

Часть 5: Шифрование данных при передаче и хранении

Почему шифрование важно

Поговорим о защите самих данных. Даже если злоумышленники преодолеют периметр вашей защиты, шифрование гарантирует, что они не смогут прочитать конфиденциальную информацию. В облачных средах шифрование защищает от множества векторов атак: злоумышленных инсайдеров у облачного провайдера, злоумышленников, скомпрометировавших соседних арендаторов, и утечки данных из-за неправильно сконфигурированных хранилищ.

Шифрование хранимых данных

Шифрование хранимых данных защищает сохранённые данные. Все крупные облачные провайдеры предлагают шифрование для сервисов хранения. AWS шифрует бакеты S3 и тома EBS с использованием AES-256. Azure Storage использует 256-битное шифрование AES. Google Cloud шифрует все данные заказчиков при хранении по умолчанию.

Однако шифрование по умолчанию может быть недостаточным. Шифрование по умолчанию часто использует ключи, управляемые провайдером, что означает, что облачный провайдер контролирует ключи шифрования. Для конфиденциальных данных рассмотрите ключи, управляемые заказчиком, или решения с привлечением собственных ключей.

При использовании ключей, управляемых заказчиком, вы контролируете ключи через облачные сервисы управления ключами, такие как AWS KMS, Azure Key Vault или Google Cloud KMS. Вы определяете политики ротации ключей, управление доступом и ведение журнала аудита. Если вы удалите ключ, данные станут навсегда недоступными.

Принцип Bring Your Own Key (BYOK), или привлечение собственного ключа, идёт дальше. Вы генерируете ключи за пределами облака и импортируете их. Некоторые организации используют аппаратные модули безопасности в своих центрах обработки данных для генерации и защиты ключей перед загрузкой в облачный сервис управления ключами.

Шифрование при передаче

Шифрование при передаче защищает данные, перемещающиеся по сетям. Протокол TLS (Transport Layer Security) является стандартным протоколом. Все коммуникации между пользователями и облачными сервисами должны использовать TLS 1.3 — последнюю версию. Более старые версии, в частности TLS 1.0 и 1.1, имеют известные уязвимости и должны быть отключены.

Внутри облачных сред также необходимо шифровать внутренний трафик. Виртуальные частные облака должны использовать шифрование между сервисами. AWS предлагает шифрование VPC, Azure — шифрование виртуальной сети, а Google Cloud — VPC Service Controls.

Технологии сервисной сетки (service mesh), такие как Istio и Linkerd, обеспечивают взаимный TLS между микросервисами. Это гарантирует, что все межсервисные коммуникации зашифрованы и аутентифицированы.

Лучшие практики управления ключами

Управление ключами зачастую сложнее, чем само шифрование. Следуйте этим рекомендациям:

Разделяйте ключи по уровню конфиденциальности. Не используйте одни и те же ключи для сред разработки и промышленной эксплуатации.

Внедряйте ротацию ключей. Автоматическая ротация ключей снижает последствия их компрометации. Облачные сервисы управления ключами поддерживают автоматическую ротацию — включите её.

Проводите аудит использования ключей. Отслеживайте, кто обращается к ключам и когда. Настройте оповещения при аномальных моделях доступа.

Планируйте восстановление ключей. Если вы потеряете доступ к ключам шифрования, вы потеряете доступ к данным. Внедрите безопасные процедуры резервного копирования и восстановления ключей.

Рассмотрите конвертное шифрование для больших объёмов данных. Зашифруйте данные ключом шифрования данных, затем зашифруйте этот ключ мастер-ключом. Это ограничивает зону воздействия и обеспечивает эффективную ротацию ключей.

Часть 6: Мультиоблачная безопасность и безопасность контейнеров

Мультиоблачная реальность

Большинство организаций сегодня используют нескольких облачных провайдеров. Согласно отчёту Flexera, компании по управлению облачными ресурсами, за 2025 год, 87% предприятий имеют мультиоблачную стратегию. Организации используют AWS для вычислений, Azure для интеграции идентификации с Microsoft 365 и Google Cloud для рабочих нагрузок машинного обучения.

Мультиоблачные среды значительно увеличивают сложность. Каждый провайдер имеет различные инструменты безопасности, различные модели IAM и различные сетевые архитектуры. Команды безопасности должны владеть всеми платформами.

Стратегии мультиоблачной безопасности

Внедряйте облачно-независимый уровень безопасности. Такие инструменты, как HashiCorp Vault (HashiCorp, разработчик инструментов автоматизации облачной инфраструктуры), обеспечивают управление секретами во всех облаках. Платформы SIEM агрегируют журналы от нескольких облачных провайдеров. Инструменты управления безопасностью облачных конфигураций (CSPM) сканируют конфигурации в разных облаках.

Стандартизируйте политики, используя подход "политика как код". Open Policy Agent, или OPA, позволяет определять политики безопасности, применимые ко всем облакам. Определите политики один раз — применяйте везде.

Создайте единый уровень идентификации. Используйте поставщика идентификации, который осуществляет федерацию во все облака — Okta, Azure AD или Ping Identity. Избегайте создания отдельных учётных записей в каждом облаке.

Основы безопасности контейнеров

Контейнеры произвели революцию в развёртывании приложений. Контейнеры Docker упаковывают приложения вместе с их зависимостями, обеспечивая единообразное поведение в различных средах. Однако контейнеры привносят новые проблемы безопасности.

Безопасность образов контейнеров — ваша первая забота. Образы могут содержать уязвимости, унаследованные от базовых образов. Всегда сканируйте образы перед развёртыванием. Такие инструменты, как Trivy, Clair и Anchore, анализируют образы контейнеров на наличие известных уязвимостей.

Используйте минимальные базовые образы. Alpine Linux и distroless-образы имеют меньшую поверхность атаки, чем полные образы операционных систем. Чем меньше пакетов в вашем образе, тем меньше потенциальных уязвимостей.

Никогда не запускайте контейнеры от имени root. Настраивайте контейнеры на работу от имени непривилегированных пользователей. Используйте файловые системы "только для чтения" там, где это возможно. Ограничивайте возможности контейнеров с помощью контекстов безопасности.

Реестры контейнеров также нуждаются в защите. Используйте приватные реестры с управлением доступом. Включите доверие к содержимому для обеспечения целостности образов. Внедрите подписание образов с помощью таких инструментов, как Cosign или Notary.

Безопасность Kubernetes

Kubernetes оркестрирует контейнеры в масштабе, но его сложность создаёт проблемы безопасности. Поверхность атаки Kubernetes включает API-сервер, хранилище данных etcd, агенты kubelet и среду выполнения контейнеров.

Обеспечьте безопасность API-сервера Kubernetes. Включите аутентификацию — никогда не предоставляйте доступ к неаутентифицированному API. Используйте управление доступом на основе ролей (RBAC) для ограничения возможностей пользователей и сервисов. Включите ведение журнала аудита для отслеживания вызовов API.

Сетевые политики контролируют межподовые коммуникации. По умолчанию все поды могут взаимодействовать со всеми другими подами. Внедрите сетевые политики для ограничения коммуникаций только необходимыми.

Стандарты безопасности подов заменили устаревшие политики безопасности подов. Эти стандарты определяют три уровня: привилегированный, базовый и ограниченный. Используйте ограниченный уровень для чувствительных рабочих нагрузок.

Управление секретами в Kubernetes требует внимания. Секреты Kubernetes кодируются в base64, но по умолчанию не шифруются. Включите шифрование хранимых данных для etcd. Рассмотрите внешнее управление секретами с помощью Vault или интеграции с облачным сервисом управления ключами.

Ландшафт безопасности Kubernetes 2025 года включает ряд важных инструментов. Falco обеспечивает мониторинг безопасности во время выполнения. KubeArmor применяет политики безопасности на системном уровне. OPA Gatekeeper обеспечивает политики контроля допуска.

Часть 7: Безопасность бессерверных вычислений и инфраструктуры как кода

Безопасность бессерверных вычислений

Бессерверные вычисления, также называемые "Функции как услуга" (Functions as a Service), абстрагируют всё управление инфраструктурой. AWS Lambda, Azure Functions и Google Cloud Functions позволяют выполнять код без выделения серверов. Провайдер обеспечивает масштабирование, обновление и доступность.

С точки зрения безопасности, бессерверные вычисления существенно смещают ответственность. Вы больше не управляете операционными системами или средами выполнения — это делает провайдер. Ваше внимание сосредотачивается на коде приложения, конфигурации функций и обработке данных.

Однако бессерверные вычисления привносят уникальные проблемы. Функции управляются событиями, вызываемыми HTTP-запросами, сообщениями из очередей или изменениями в базе данных. Каждая точка вызова является потенциальным вектором атаки. Проверяйте и очищайте все входные данные, независимо от их источника.

Разрешения функций требуют тщательного управления. Каждой функции необходима роль выполнения с конкретными разрешениями. Строго применяйте принцип минимальных привилегий — функция, которая читает из S3, не должна иметь разрешений на запись в DynamoDB.

Безопасность при холодном запуске — это новая проблема. Когда функция не запускалась в последнее время, она должна инициализироваться заново. Злоумышленники могут пытаться приурочить атаки к холодным запускам, когда выполняется логика инициализации.

Зависимости остаются вашей ответственностью. Бессерверные функции используют библиотеки так же, как и традиционные приложения. Сканируйте зависимости на наличие уязвимостей. Используйте такие инструменты, как Snyk, npm audit или OWASP Dependency-Check.

Безопасность инфраструктуры как кода

Инфраструктура как код, или IaC, определяет облачные ресурсы в конфигурационных файлах, а не посредством ручных действий в консоли. Terraform, AWS CloudFormation, шаблоны Azure Resource Manager и Pulumi — популярные инструменты IaC.

IaC повышает безопасность, обеспечивая согласованные, воспроизводимые развёртывания. Однако она также создаёт новые риски. Неправильные конфигурации в шаблонах IaC становятся неправильными конфигурациями в промышленной инфраструктуре.

Сканируйте шаблоны IaC перед развёртыванием. Такие инструменты, как Checkov, tfsec и KICS, анализируют шаблоны на предмет проблем безопасности. Они обнаруживают чрезмерно разрешительные группы безопасности, незашифрованные хранилища и отсутствующие конфигурации журналирования.

Относитесь к IaC как к коду с полным циклом безопасности разработки программного обеспечения. Храните шаблоны в системе контроля версий. Требуйте проверки кода при изменениях. Внедряйте автоматизированное тестирование в конвейерах CI/CD.

Секреты в IaC — распространённая уязвимость. Никогда не записывайте учётные данные, ключи API или пароли непосредственно в шаблоны. Используйте ссылки на секреты из внешних менеджеров секретов. Включите сканирование секретов в ваших репозиториях с помощью таких инструментов, как GitLeaks или TruffleHog.

Безопасность файлов состояния имеет значение для таких инструментов, как Terraform. Файлы состояния содержат конфиденциальную информацию о вашей инфраструктуре. Храните файлы состояния безопасно — используйте зашифрованные удалённые хранилища, такие как S3 с шифрованием или Terraform Cloud.

Обнаружение отклонений гарантирует, что развёрнутая инфраструктура соответствует вашим определениям IaC. Облачные ресурсы могут быть изменены за пределами IaC через консольные изменения или скрипты. Обнаруживайте и устраняйте отклонения для поддержания уровня безопасности.

Часть 8: CNAPP, безопасность API и новые тенденции

Платформы защиты облачных приложений

CNAPP — Платформа защиты облачных приложений (Cloud-Native Application Protection Platform) — это новая категория, объединяющая множество возможностей облачной безопасности. Gartner ввёла этот термин в 2021 году, и к 2025 году CNAPP стала необходимой для организаций со значительным присутствием в облаке.

CNAPP объединяет несколько ранее отдельных инструментов: управление безопасностью облачных конфигураций (CSPM) для анализа конфигурации, платформу защиты облачных рабочих нагрузок (CWPP, Cloud Workload Protection Platform) для безопасности во время выполнения, управление правами в облачной инфраструктуре (CIEM, Cloud Infrastructure Entitlement Management) для анализа разрешений и сканирование инфраструктуры как кода для безопасности на этапе разработки.

Преимущество CNAPP — единая видимость. Вместо управления пятью различными инструментами безопасности с пятью различными информационными панелями, CNAPP обеспечивает единое представление состояния облачной безопасности. Это позволяет коррелировать риски в средах разработки и выполнения.

Ведущие поставщики CNAPP включают Palo Alto Prisma Cloud, Wiz, разработчика платформы облачной безопасности, Orca Security, разработчика решений безагентной облачной безопасности, и Lacework, разработчика платформы защиты облачных данных. AWS, Azure и Google также предлагают встроенные возможности CNAPP через свои сервисы безопасности.

При оценке решений CNAPP учитывайте охват по вашим облачным провайдерам, интеграцию с вашими конвейерами разработки и точность приоритизации рисков.

Безопасность API

API — это соединительная ткань облачных приложений. Микросервисы взаимодействуют через API. Мобильные приложения вызывают серверные API. Сторонние интеграции используют API. Согласно отчёту Salt Security, разработчика решений защиты API, за 2024 год, более 60% организаций столкнулись с инцидентом безопасности API за последний год.

Безопасность API начинается с аутентификации и авторизации. Используйте надёжную аутентификацию для всех вызовов API. OAuth 2.0 и OpenID Connect являются стандартными протоколами. Внедряйте надлежащие проверки

авторизации — убеждайтесь, что аутентифицированные пользователи имеют разрешение на доступ к запрашиваемым ресурсам.

Ограничение частоты запросов предотвращает злоупотребления. Без ограничения частоты злоумышленники могут проводить атаки методом перебора, атаки типа "отказ в обслуживании" или эксфильтрацию данных посредством быстрых вызовов API.

Валидация входных данных применяется к API так же, как и к веб-приложениям. Проверяйте все входные параметры. Используйте валидацию схемы с помощью спецификаций OpenAPI.

Обнаружение API — сложная задача. Организации часто не знают обо всех своих API — теневые API, созданные командами разработчиков без проверки безопасности. Инструменты безопасности API могут обнаруживать API с помощью анализа трафика.

Шлюзы API обеспечивают централизованные меры безопасности. AWS API Gateway, Azure API Management и Kong применяют аутентификацию, ограничение частоты запросов и анализ трафика на уровне шлюза.

Отслеживайте трафик API на предмет аномалий. Определите базовое нормальное поведение и настройте оповещения при отклонениях. Обращайте внимание на избыточный объём возвращаемых данных, необычные модели доступа или попытки обращения к недокументированным конечным точкам.

Новые тенденции облачной безопасности на 2025-2026 годы

Позвольте завершить этот раздел рассмотрением новых тенденций, за которыми вам следует наблюдать.

Архитектура нулевого доверия больше не является теоретической — она стала ожидаемым стандартом. Облачная безопасность всё чаще исходит из отсутствия неявного доверия на основе сетевого расположения. Каждый запрос доступа проверяется, независимо от источника.

Безопасность ИИ действует в обоих направлениях. Инструменты ИИ помогают защитникам анализировать журналы и обнаруживать угрозы, но злоумышленники используют ИИ для создания изощрённых атак. Облачные сервисы ИИ требуют собственных мер безопасности в отношении защиты моделей, безопасности обучающих данных и защиты конечных точек вывода.

Начинается подготовка к квантовым вычислениям. Хотя квантовые компьютеры пока не способны взломать существующее шифрование, организации начинают инвентаризацию криптографического использования и планирование миграции на квантовоустойчивые алгоритмы. NIST стандартизировал постквантовые

криптографические алгоритмы в 2024 году, и облачные провайдеры начинают предлагать квантовоустойчивые решения.

Конфиденциальные вычисления защищают данные во время обработки с использованием аппаратных доверенных сред выполнения. AMD SEV, Intel SGX и ARM Confidential Compute Architecture позволяют выполнять рабочие нагрузки, при которых даже облачный провайдер не может получить доступ к обрабатываемым данным.

FinOps и безопасность сближаются. Команды безопасности работают с финансовыми командами для понимания затрат на облако, поскольку меры безопасности влияют на расходы, а оптимизация затрат не должна ставить под угрозу безопасность.

Часть 9: План практического лабораторного занятия

Практическая работа по облачной безопасности

Для закрепления сегодняшних концепций я хочу изложить план практического лабораторного занятия, которое вы можете выполнить самостоятельно, используя бесплатные облачные учётные записи.

Упражнение 1: Создайте безопасную конфигурацию бакета S3. Включите шифрование по умолчанию с ключом KMS, управляемым заказчиком. Настройте политику бакета для запрета незашифрованных загрузок. Включите журналирование доступа. Заблокируйте публичный доступ.

Упражнение 2: Внедрите принцип минимальных привилегий IAM. Создайте пользователя IAM только с теми разрешениями, которые необходимы для чтения объектов из конкретного бакета S3. Используйте IAM Access Analyzer для проверки отсутствия избыточных разрешений.

Упражнение 3: Просканируйте образ контейнера. Загрузите публичный образ Docker. Просканируйте его с помощью Trivy. Определите и задокументируйте все уязвимости критической и высокой степени серьёзности.

Упражнение 4: Просканируйте шаблон Terraform. Возьмите образец конфигурации Terraform, создающей экземпляры EC2 и группы безопасности. Просканируйте с помощью tfsec или Checkov. Определите неправильные конфигурации и устраните их.

Упражнение 5: Настройте безопасность API Gateway. Создайте простой API с помощью AWS API Gateway. Добавьте аутентификацию по ключу API. Настройте ограничение частоты запросов. Включите журналирование CloudWatch.

Эти упражнения обеспечивают практический опыт работы с концепциями, рассмотренными сегодня.

Заключение

Позвольте подвести итоги сегодняшней лекции по основам облачной безопасности.

Мы начали с изучения трёх моделей облачных сервисов: IaaS, PaaS и SaaS. Каждая модель имеет различные характеристики и различные последствия для безопасности. Знание модели помогает вам понять вашу ответственность.

Модель разделённой ответственности — наиболее важная концепция облачной безопасности. Провайдер обеспечивает безопасность инфраструктуры; вы обеспечиваете безопасность ваших данных, приложений и конфигураций. Непонимание этого разделения является причиной большинства облачных утечек.

Альянс по облачной безопасности (CSA) предоставляет важнейшие руководства через свой документ Security Guidance, Матрицу облачных контролей и реестр STAR. Эти ресурсы помогают оценивать и улучшать состояние вашей облачной безопасности.

Управление идентификацией и доступом является основой облачной безопасности. Внедряйте принцип минимальных привилегий, требуйте многофакторную аутентификацию и контролируйте доступ посредством регулярных проверок.

Шифрование защищает данные при хранении и передаче. Используйте ключи, управляемые заказчиком, для конфиденциальных данных. Внедряйте TLS 1.3 для всех сетевых коммуникаций.

Мультиоблачные среды требуют облачно-независимых инструментов безопасности и стандартизированных политик. Безопасность контейнеров охватывает сканирование образов, защиту во время выполнения и укрепление Kubernetes.

Безопасность бессерверных вычислений фокусируется на разрешениях функций и валидации входных данных. Безопасность инфраструктуры как кода означает сканирование шаблонов перед развёртыванием и защиту файлов состояния.

CNAPP объединяет возможности облачной безопасности в единые платформы. Безопасность API решает проблему растущей поверхности атаки современных приложений.

Для подготовки прочитайте CSA Security Guidance версии 4. Изучите документацию вашего облачного провайдера по разделённой ответственности. Выполните практические лабораторные упражнения, которые я изложил.

На следующей лекции мы рассмотрим устойчивость инфраструктуры — как проектировать системы, способные выдерживать сбои и атаки.

Вопросы для обсуждения

1. Каковы наиболее распространённые заблуждения относительно модели разделённой ответственности и как они приводят к инцидентам безопасности?
2. Как организациям следует подходить к безопасности иначе при работе в мультиоблачных средах?
3. Какие уникальные проблемы безопасности привносят бессерверные вычисления по сравнению с традиционной инфраструктурой?
4. Стартап хранит все данные клиентов у одного облачного провайдера. Какие риски это создаёт и как можно уменьшить зависимость?
5. Если ваш облачный провайдер допустил утечку данных, кто несёт ответственность за уведомление пострадавших клиентов: вы или провайдер?

Благодарю за внимание. Есть ли вопросы?

Контрольные вопросы

1. Объясните модель разделённой ответственности и то, как обязанности по безопасности различаются между IaaS (Infrastructure as a Service, инфраструктура как услуга), PaaS (Platform as a Service, платформа как услуга) и SaaS (Software as a Service, программное обеспечение как услуга).
2. Что такое Альянс по облачной безопасности CSA (Cloud Security Alliance, Альянс по облачной безопасности) и какие ключевые ресурсы он предоставляет для облачной безопасности?
3. Каковы ключевые аспекты безопасности для мультиоблачных сред?
4. Назовите три основные модели облачных сервисов (IaaS, PaaS, SaaS) и приведите по одному примеру каждой.
5. Что такое CASB (Cloud Access Security Broker, брокер безопасности облачного доступа) и какую проблему он решает?

Ключевые термины

- **Безопасность API (API Security)**: Практики и инструменты для защиты программных интерфейсов приложений от несанкционированного доступа и атак
- **CASB**: Брокер безопасности облачного доступа (Cloud Access Security Broker), обеспечивающий видимость и контроль для облачных приложений

- **CCM**: Матрица облачных контролей (Cloud Controls Matrix), фреймворк CSA, отображающий контроли облачной безопасности
- **CNAPP**: Платформа защиты облачных приложений (Cloud-Native Application Protection Platform), объединяющая инструменты облачной безопасности
- **Безопасность контейнеров (Container Security)**: Меры безопасности для защиты контейнеризованных приложений на протяжении всего их жизненного цикла
- **CSA**: Альянс по облачной безопасности (Cloud Security Alliance), ведущая организация в области руководств по облачной безопасности
- **CSPM**: Управление безопасностью облачных конфигураций (Cloud Security Posture Management), мониторинг облачной конфигурации на предмет проблем безопасности
- **CWPP**: Платформа защиты облачных рабочих нагрузок (Cloud Workload Protection Platform), обеспечивающая безопасность облачных рабочих нагрузок
- **Конвертное шифрование (Envelope Encryption)**: Техника, использующая ключ шифрования данных, обёрнутый мастер-ключом, обеспечивающая эффективное крупномасштабное шифрование
- **FedRAMP**: Федеральная программа управления рисками и авторизацией (Federal Risk and Authorization Management Program), сертификация облачной безопасности правительства США
- **IaaS**: Инфраструктура как услуга (Infrastructure as a Service), предоставление виртуализированных вычислительных ресурсов
- **IAM**: Управление идентификацией и доступом (Identity and Access Management), контроль того, кто может получить доступ к чему в облачных средах
- **IaC**: Инфраструктура как код (Infrastructure as Code), управление инфраструктурой посредством машиночитаемых конфигурационных файлов
- **KMS**: Сервис управления ключами (Key Management Service), сервис облачного провайдера для управления ключами шифрования
- **Kubernetes**: Платформа оркестрации контейнеров с открытым исходным кодом с функциями безопасности, включая RBAC и сетевые политики
- **PaaS**: Платформа как услуга (Platform as a Service), предоставление платформы для разработки и развёртывания приложений
- **SaaS**: Программное обеспечение как услуга (Software as a Service), предоставление приложений через интернет
- **Безопасность бессерверных вычислений (Serverless Security)**: Практики безопасности для сред "Функции как услуга", ориентированные на разрешения и валидацию входных данных

- **Реестр STAR (STAR Registry)**: Реестр Security Trust Assurance and Risk, реестр CSA с оценками безопасности облачных провайдеров
- **Модель разделённой ответственности (Shared Responsibility Model)**: Разделение обязанностей по безопасности между облачным провайдером и заказчиком