Принципы, политики, процедуры и сервисы безопасности в AWS

SOFTPROM
softprom.com • info@softprom.com

amazon web services | Partner Network
ADVANCED CONSULTING PARTNER

2020

Принципы, политики, процедуры и сервисы безопасности в AWS

Основные принципы обеспечения безопасности в облаке AWS

- Внедрение строгой идентификации на основе стратегии наименьших привилегий (см. Стратегия управления доступом в AWS)
- Реализация непрерывного мониторинга, оповещения и аудита в режиме реального времени
- Внедрение безопасности на всех уровнях: сетевом (VPC), load balancing, виртуальной машины, операционной системы, приложения и программного кода
- Применение лучших практик по созданию безопасной инфраструктуры на основе программного кода. Инфраструктура как программный код (AWS CloudFormation)
- Реализация защиты данных при передаче и хранении
- Внедрение инструментов позволяющих исключить человеческий фактор, уменьшить необходимость прямого доступа или ручной обработки данных.
- Моделирование реагирования на инциденты, использование инструментов с автоматизацией, чтобы повысить скорость обнаружения, расследования и восстановления.

Безопасность в облаке состоит из пяти разделов

1. Управление идентификацией и доступом (Identity and access management)
2. Обнаружение (Detection)
3. Защита инфраструктуры (Infrastructure protection)
4. Защиты данных (Data protection)
5. Реагирование на чрезвычайные ситуации (Incident response)

Сервисы AWS по безопасности

Ниже приведены сервисы AWS, применимые в соответствующих разделах безопасности

Управление идентификацией и доступом

Пример использования	Сервис AWS
Безопасное управление доступом к сервисам и ресурсам	AWS Identity & Access Management (IAM)
Облачный сервис единого входа (SSO)	AWS Single Sign-On
Управление идентификацией пользователей в приложениях	Amazon Cognito
Управляемая система Microsoft Active Directory	AWS Directory Service
Простой и безопасный сервис для совместного доступа к ресурсам AWS	AWS Resource Access Manager

Управление и администрирование

Пример использования	Сервис AWS
все аккаунты AWS в одном месте	AWS Organizations

Обнаружение

Пример использования	Сервис AWS
Унифицированный центр безопасности и соответствия требованиям	AWS Security Hub
Управляемый сервис обнаружения угроз	Amazon GuardDuty
Анализ безопасности приложений	Amazon Inspector
Запись и оценка конфигурации ресурсов AWS	AWS Config
Отслеживание действий пользователей и использования API	AWS CloudTrail
Управление безопасностью для устройств Интернета вещей	AWS IoT Device Defender

Защита инфраструктуры

Пример использования	Сервис AWS

Защита от DDoS-атак	AWS Shield
Фильтрация вредоносного сетевого трафика	AWS Web Application Firewall (WAF)
Централизованное управление правилами брандмауэра	AWS Firewall Manager

Защита данных

Пример использования	Сервис AWS
Выявляйте и защищайте конфиденциальные данные в любом масштабе	Amazon Macie
Хранение ключей и управление ими	AWS Key Management Service (KMS)
Аппаратное хранилище ключей для соответствия нормативным требованиям	AWS CloudHSM
Создание, развертывание публичных и частных сертификатов SSL/TLS и управление ими	AWS Certificate Manager
Ротация и извлечение конфиденциальных данных, а также управление ими	AWS Secrets Manager

Реагирование на чрезвычайные ситуации

Пример использования	Сервис AWS
Анализ потенциальных проблем безопасности	Amazon Detective
Быстрое и экономичное автоматизированное аварийное восстановление	CloudEndure Disaster Recovery

Соответствие требованиям

Пример использования	Сервис AWS

Бесплатный портал самообслуживания для доступа по требованию к отчетам AWS о соответствии требованиям	AWS Artifact

Процедуры аудита безопасности по разделам безопасности

This checklist provides customer recommendations that align with the Well-Architected Framework Security Pillar.

Identity & Access Management

1. Secure your AWS account. Use AWS Organizations to manage your accounts, use the root user by exception with multi-factor authentication (MFA) enabled, and configure account contacts.
2. Rely on centralized identity provider. Centralize identities using either AWS Single Sign-On or a third-party provider to avoid routinely creating IAM users or using long-term access keys—this approach makes it easier to manage multiple AWS accounts and federated applications.
3. Use multiple AWS accounts to separate workloads and workload stages such as production and non-production. Multiple AWS accounts allow you to separate data and resources, and enable the use of Service Control Policies to implement guardrails. AWS Control Tower can help you easily set up and govern a multi-account AWS environment.
4. Store and use secrets securely. Where you cannot use temporary credentials, like tokens from AWS Security Token Service, store your secrets like database passwords using AWS Secrets Manager which handles encryption, rotation, and access control..

Detection

1. Enable foundational services: AWS CloudTrail, Amazon GuardDuty, and AWS Security Hub. For all your AWS accounts configure CloudTrail to log API activity, use GuardDuty for continuous monitoring, and use AWS Security Hub for a comprehensive view of your security posture..
2. Configure service and application level logging. In addition to your application logs, enable logging at the service level, such as Amazon VPC Flow Logs and Amazon S3, CloudTrail, and Elastic Load Balancer access logging, to gain visibility into events. Configure logs to flow to a central account, and protect them from manipulation or deletion.
3. Configure monitoring and alerts, and investigate events. Enable AWS Config to track the history of resources, and Config Managed Rules to automatically alert or remediate on undesired changes. For all your sources of logs and events, from AWS CloudTrail, to Amazon GuardDuty and your application logs, configure alerts for high priority events and investigate.

Infrastructure Protection

1. Patch your operating system, applications, and code. Use AWS Systems Manager Patch Manager to automate the patching process of all systems and code for which you are responsible, including your OS, applications, and code dependencies.
2. Implement distributed denial-of-service (DDoS) protection for your internet facing resources. Use Amazon Cloudfront, AWS WAF and AWS Shield to provide layer 7 and layer 3/layer 4 DDoS protection.
3. Control access using VPC Security Groups and subnet layers. Use security groups for controlling inbound and outbound traffic, and automatically apply rules for both security groups and WAFs using AWS Firewall Manager. Group different resources into different subnets to create routing layers, for example database resources do not need a route to the internet.

Data Protection

1. Protect data at rest. Use AWS Key Management Service (KMS) to protect data at rest across a wide range of AWS services and your applications. Enable default encryption for Amazon EBS volumes, and Amazon S3 buckets.
2. Encrypt data in transit. Enable encryption for all network traffic, including Transport Layer Security (TLS) for web based network infrastructure you control using AWS Certificate Manager to manage and provision certificates.
3. Use mechanisms to keep people away from data. Keep all users away from directly accessing sensitive data and systems. For example, provide an Amazon QuickSight dashboard to business users instead of direct access to a database, and perform actions at a distance using AWS Systems Manager automation documents and Run Command.

Incident Response

1. Ensure you have an incident response (IR) plan. Begin your IR plan by building runbooks to respond to unexpected events in your workload. For details, see the AWS Security Incident Response Guide.
2. Make sure that someone is notified to take action on critical findings. Begin with GuardDuty findings. Turn on GuardDuty and ensure that someone with the ability to take action receives the notifications. Automatically creating trouble tickets is the best way to ensure that GuardDuty findings are integrated with your operational processes.
3. Practice responding to events. Simulate and practice incident response by running regular game days, incorporating the lessons learned into your incident management plans, and continuously improving them.

Процедуры аудита безопасности по сервисам AWS

IAM

- Avoid using AWS root account user access keys as it gives full access to all resources.
- MFA authentication is enabled for the root account to provide two-factor authentication.
- Assign individual IAM users with necessary permissions to enable login.
- Ensure User Accounts also have MFA authentication.
- IAM Access Keys must be rotated at periodic intervals.
- Ensure a strong password policy for users.
- Assign permissions to users based on User Groups, instead of individual IAM users.
- Provide access to a resource through IAM Roles
- Grant least access while creating IAM Policies, needed to perform the necessary actions
- Attach IAM Policies to Groups or Roles on creation
- If required, conditions can be defined for Policies under which access is granted to a resource
- Get rid of unnecessary IAM credentials, those with are inactive or unused
- Use IAM Roles to grant access to applications on EC2 Instances

S3

- Ensure S3 buckets are not publicly accessible (public read or write permissions) — users can enable Amazon S3 to block public access.
- Make use of object-level or bucket-level permissions in addition to IAM Policies to grant access to resources.
- Enable MFA Delete to prevent accidental deletion of buckets.
- Consider encryption of stored data, which can be done in two ways — server-side and client-side encryption.
- Enable encryption of inbound and outbound data traffic, through SSL endpoints.
- Configure S3 lifecycle management through rule-based actions and use versioning to store and retrieve multiple versions of an object in a bucket, to deal with accidental deletions.
- Ensure S3 access logging is enabled.
- Constantly audit and monitor S3 buckets using CloudWatch metrics.

EC2, VPC, and EBS

- Ensure data and disk volumes in EBS are encrypted with AES-256, the industry-standard algorithm.
- Restrict access to instances from limited IP ranges using a Security Group.
- Limit the range of open ports on EC2 security groups, to prevent exposure to vulnerabilities.

- Ensure ELBs have a valid security group attached to it.
- Monitor and optimize default security groups, as they allow unrestricted access for inbound and outbound traffic.
- Ensure restricted inbound access to SSH, FTP, SMTP, MySQL, PostgreSQL, MongoDB, MSSQL, CIFS, etc. to required entities only.
- Use IAM roles to grant access to EC2, instead of access keys for temporary requirements.
- If you're using IAM user access keys for long term permissions, ensure that you don't embed the keys directly into code, generate different keys for different applications, rotate your access keys, use MFA authentication and decommission unused key pairs.
- Enable and activate your VPC flow logs to record inbound and outbound traffic in your VPC for better monitoring and early diagnosis.
- Delete unused Virtual Private Gateways and VPC Internet Gateways.
- Make sure that no VPC endpoints are exposed, by checking the principal value in the policy.
- Ensure no ACLs allow unrestricted inbound or outbound access.

CloudTrail

- Ensure CloudTrail is activated across all regions, and for global services like IAM, STS, etc.
- It is recommended to log to a centralized S3 bucket.
- Make sure both CloudTrail itself and CloudTrail logging are enabled for all regions.
- Ensure CloudTrail log file integrity validation is enabled.
- Ensure CloudTrail log files are encrypted.

RDS

- Ensure RDS security groups do not allow unrestricted access.
- Ensure encryption of the RDS instances and snapshots, using AES-256 level encryption.
- Protect data in transit to RDS through SSL endpoints.
- Monitor control to RDS using AWS KMS and Customer Managed Keys.
- Configure AWS Secrets Manager to automatically rotate the secrets for Amazon RDS.
- Ensure RDS database instances and snapshots are not publicly accessible.
- Enable the auto minor upgrade feature for RDS.

Redshift

- Enable the require_ssl parameter in all Redshift clusters to minimize risk for encryption of data in transit for Redshift, and to connect your SQL client with your cluster.
- Enable Redshift Cluster encryption.
- Ensure Redshift user activity logging is enabled.
- Ensure Redshift encryption with KMS Customer Managed Keys.

- It is recommended that Redshift clusters are launched within a VPC for better control.
- Ensure that the Redshift clusters are not publicly accessible.

Встроенная безопасность

Встроенная безопасность (SbD) – это подход к обеспечению безопасности, который формализует проектирование аккаунта AWS, автоматизирует системы контроля безопасности и упрощает процессы аудита. Подход SbD, в отличие от аудита безопасности за прошлый период, обеспечивает контроль безопасности, изначально встроенный в процесс управления IT-ресурсами платформы AWS. Использование шаблонов встроенной безопасности (SbD) в AWS CloudFormation позволяет добиться разносторонней и эффективной защиты и обеспечить соответствие требованиям в облаке.

SbD – это подход к обеспечению безопасности и соответствия требованиям в любом масштабе с учетом специфики различных отраслей, стандартов и критериев безопасности. Подход SbD в AWS можно использовать при проектировании функциональных возможностей безопасности и соответствия требованиям на всех этапах, что позволяет клиенту проектировать все компоненты собственной среды AWS: разрешения, ведение журналов, доверительные отношения, принудительное шифрование, использование только одобренных образов машин и многое другое. Подход SbD позволяет автоматизировать интерфейсную часть аккаунта AWS, надежно программируя безопасность и соответствие стандартам в используемых аккаунтах AWS и оставляя в прошлом несоответствие систем ИТ-управления стандартным требованиям.

Подход, основанный на встроенной безопасности (SbD)

Подход SbD формирует обязанности по контролю, принципы автоматизации основ безопасности, настройки безопасности и требования по аудиту клиентом систем управления в своей инфраструктуре, операционных системах, сервисах и приложениях, работающих на AWS. Предлагаемые программой директивные стандартизованные воспроизводимые проекты со встроенной автоматизацией можно развертывать для распространенных примеров использования с учетом стандартов безопасности и требований к аудиту в различных отраслях и для различных рабочих нагрузок.

Компания AWS рекомендует использовать в своем аккаунте AWS встроенную безопасность и соответствие требованиям, следуя представленному ниже подходу из четырех этапов.

Этап 1.

Сформируйте требования. Определите политики и опишите системы контроля, наследуемые от AWS. Затем опишите системы контроля, используемые вами в среде AWS, и определите список правил безопасности, которые требуется внедрить в ИТ-среду AWS.

Этап 2.

Создайте безопасную среду, соответствующую вашим требованиям и особенностям внедрения. Определите требуемую конфигурацию в виде значений конфигурации AWS, таких как требования к шифрованию (например, принудительное шифрование для объектов S3 на стороне сервера), разрешения для ресурсов (какие роли применимы к определенным средам), список разрешенных к использованию вычислительных образов (на основании фиксированных образов серверов, разрешенных к использованию), а также типы обязательных журналов (например, принудительное применение сервиса CloudTrail для совместимых ресурсов). Поскольку AWS предоставляет разносторонний набор вариантов конфигурации и регулярно добавляет в него новые сервисы, вы сможете найти шаблоны для приведения своей среды в соответствие заданным требованиям безопасности. Данные шаблоны безопасности (в виде шаблонов AWS CloudFormation) предоставляют всеобъемлющий набор правил, который можно внедрять систематически. Компания AWS разработала шаблоны, которые обеспечивают правила безопасности, соответствующие различным инфраструктурам безопасности.

Дополнительную помощь при создании безопасной среды могут оказать опытные архитекторы компании AWS, специалисты AWS Professional Services и решения от наших партнеров. Эти группы специалистов могут работать вместе с вашим персоналом и группами аудита для помощи в реализации высококачественных безопасных сред для прохождения стороннего аудита.

Этап 3.

Внедрите созданные шаблоны. AWS Service Catalog дает пользователям возможность принудительно применять в каталоге собственные шаблоны. Этот шаг позволяет гарантировать использование безопасной среды при создании любых новых сред. Он также предотвращает создание сред, которые не соответствуют установленным правилам безопасности среды. Требование обязательного использования определенного шаблона в каталоге гарантирует клиентам, что настроенные механизмы безопасности будут готовы к стороннему аудиту.

Этап 4.

Выполните проверочные действия. Развертывание на AWS с помощью Service Catalog и шаблонов безопасной среды помогает создать среду, готовую к аудиту. Правила, определенные в шаблоне, можно использовать как руководство по аудиту. AWS Config позволяет сканировать текущее состояние любой среды, чтобы сравнить его с установленными правилами безопасной среды. Использование защищенных разрешений на доступ к чтению вместе с уникальными скриптами позволяет автоматизировать сбор сведений для проведения аудита. Клиенты могут перейти от традиционных ручных средств административного контроля к реализуемым принудительно на техническом уровне средствам контроля, сохраняя уверенность, что при правильной разработке и применении эти средства контроля функциональны на 100 % в любое время (в отличие от традиционных методов выборочного аудита и проверок состояния на определенный момент).

Значение встроенной безопасности

Встроенная безопасность позволяет реализовать следующие возможности:

- создание принудительно применяемых функций, недоступных для переопределения пользователями, которые не имеют соответствующего разрешения;
- организация надежной эксплуатации систем контроля;
- непрерывный аудит в режиме реального времени;
- применение политик управления в форме программных скриптов.

Результат обеспечивает автоматизированную среду, поддерживающую функции безопасности, управления и соответствия конкретным требованиям. Вы можете использовать надежную реализацию возможностей, ранее описанных только в политиках, стандартах и нормативах. Кроме того, можно обеспечивать безопасность и соответствие стандартам принудительно, что в итоге приводит к созданию функциональной и надежной модели управления для сред AWS.