Introduction to AWS Security by Design

A Solution to Automate Security, Compliance, and Auditing in AWS

November 2015

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Security by Design (SbD) is a security assurance approach that enables customers to formalize AWS account design, automate security controls, and streamline auditing.

This whitepaper discusses the concepts of Security by Design, provides a four-phase approach for security and compliance at scale across multiple industries, points to the resources available to AWS customers to implement security into the AWS environment, and describes how to validate controls are operating.

Introduction

Security by Design (SbD) is a security assurance approach that enables customers to formalize AWS account design, automate security controls, and streamline auditing. It is a systematic approach to ensure security; instead of relying on auditing security retroactively, SbD provides you with the ability to build security control in throughout the AWS IT management process.

SbD encompasses a four-phase approach for security and compliance at scale across multiple industries, standards and security criteria. AWS SbD is about designing security and compliance capabilities for all phases of security by designing everything within the AWS customer environment: the permissions, the logging, the use of approved machine images, the trust relationships, the changes made, enforcing encryption, and more. SbD enables customers to automate the front-end structure of an AWS account to make security and compliance reliably coded into the account.

Security in the AWS Environment

The AWS infrastructure has been designed to provide the highest availability while putting strong safeguards in place regarding customer privacy and segregation. When deploying systems in the AWS Cloud, AWS and its customers share the security responsibilities. AWS manages the underlying infrastructure, while your responsibility is to secure the IT resources deployed in AWS.

AWS allows you to formalize the application of security controls in the customer platform, simplifying system use for administrators and allowing for a simpler and more secure audit of your AWS environment.

There are two aspects of AWS security:

> Security of the AWS environment. The AWS account itself has configurations and features you can use to build in security. Identities, logging functions, encryption functions, and rules around how the systems are used and networked are all part of the AWS environment you manage.
>
> Security of hosts and applications. The operating systems, databases stored on disks, and the applications customers manage need security protections as well. This is up to the AWS customer to manage. Security process, tools

and techniques which customers use today within their on-premise environments also exist within AWS.

The Security by Design approach here applies primarily to the AWS environment. The centralized access, visibility and transparency of operating with the AWS cloud provides for increased capability for designing end-to-end security for all services, data, and applications in AWS.

Security by Design: Overview

SbD allows customers to automate the fundamental structure to reliably code security and compliance of the AWS environment, making it easier to render non-compliance for IT controls a thing of the past. By creating a secure and repeatable approach to the cloud infrastructure approach to security; customers can capture, secure, and control specific infrastructure control elements. These elements enable deployment of security compliant processes for IT elements, such as pre-defining and constraining the design of AWS Identify and Access Management (IAM), AWS Key Management Services (KMS) and AWS CloudTrail.

SbD follows the same general concept as Quality by Design, or QbD. Quality by Design is a concept first outlined by quality expert Joseph M. Juran in *Juran on Quality by Design*. Designing for quality and innovation is one of the three universal processes of the Juran Trilogy, in which Juran describes what is required to achieve breakthroughs in new products, services, and processes. The general shift in manufacturing companies moving to a QbD approach is to ensure quality is *built* into the manufacturing process, moving away from using post-production quality checks as the primary way in which quality is controlled.

As with QbD concepts, Security by Design can also be planned, executed and maintained through system design as a reliable way to ensure real-time, scalable and reliable security throughout the lifespan of a technology deployment in AWS. Relying on the audit function to fix present issues around security is not reliable or scalable.

Security by Design Approach

SbD outlines the inheritances, the automation of baseline controls, the operationalization and audit of implemented security controls for AWS infrastructure, operating systems, services and applications running in AWS. This standardized, automated, and repeatable architectures can be deployed for common use cases, security standards and audit requirements across multiple industries and workloads.

We recommend building in security and compliance into your AWS account by following a basic four-phase approach:

- **Phase 1 – Understand your requirements.** Outline your policies, and then document the controls you inherit from AWS, document the controls you own and operate in your AWS environment, and decide on what security rules you want to enforce in your AWS IT environment.
- **Phase 2 – Build a "secure environment" that fits your requirements and implementation.** Define the configuration you require in the form of AWS configuration values, such as encryption requirements (forcing server side encryption for S3 objects), permissions to resources (which roles apply to certain environments), which compute images are authorized (based on hardened images of servers you have authorized), and what kind of logging needs to be enabled (such as enforcing the use of CloudTrail on all resources for which it is available). Since AWS provides a mature set of configuration options (with new services being regularly released), we provide some templates for you to leverage for your own environment. These security templates (in the form of AWS CloudFormation Templates) provide a more comprehensive rule set that can be systematically enforced. We have developed templates that provide security rules that conform to multiple security frameworks and leading practices. These pre-packaged industry template solutions are provided to customers as a suite of templates or as stand alone templates based on specific security domains (e.g. access control, security services, network security, etc.)

  More help to create this "secure environment" is available from AWS experienced architects, AWS Professional Services, and partner IT transformation leaders. These teams can work alongside your staff and audit teams to focus on high quality secure customer environments in support of third-party audits.
- **Phase 3 – Enforce the use of the templates.** Enable Service Catalog, and enforce the use of your template in the catalog. This is the step, which enforces the use of your "secure environment" in new

environments that are being created, and prevents anyone from creating an environment that doesn't adhere to your "secure environment" standard rules or constraints. This effectively operationalizes the remaining customer account security configurations of controls in preparation for audit readiness.

- Phase 4 – Perform validation activities. Deploying AWS through Service Catalog and the "secure environment" templates creates an audit-ready environment. The rules you defined in your template can be used as an audit guide. AWS Config allows you to capture the current state of any environment, which can then be compared with your "secure environment" standard rules. This provides audit evidence gathering capabilities through secure "read access" permissions, along with unique scripts, which enable audit automation for evidence collection. Customers will be able to convert traditional manual administrative controls to technically enforced controls with the assurance that, if designed and scoped properly, the controls are operating 100% at any point in time - versus traditional audit sampling methods or point-in-time reviews.

This technical audit can be augmented by pre-audit guidance; support and training for customer auditors to ensure audit personnel understand the unique audit automation capabilities, which the AWS cloud provides.

Impact of Security by Design

SbD Architecture is meant to achieve the following:

- Creating forcing functions that cannot be overridden by the users without modification rights.
- Establishing reliable operation of controls.
- Enabling continuous and real-time auditing.
- The technical scripting your governance policy.

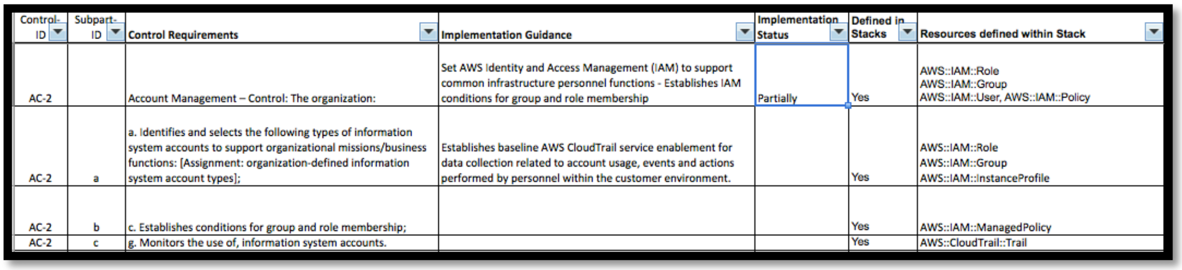
The result is an automated environment enabling the customer's security assurance, governance, security, and compliance capabilities. Customers can now get reliable implementation of what was previously written in policies, standards and regulations. Customers can create enforceable security and compliance, which in turn creates a functional reliable governance model for AWS customer environments.

SbD Approach Details

Phase 1 – Understand Your Requirements

Start by performing a security control rationalization effort. You can create a security Controls Implementation Matrix (CIM) that will identify inherency from existing AWS certifications, accreditations and reports, as well as identify the shared, customer architecture optimized controls, which *should* be implemented in any AWS environment, regardless of security requirements. The result of this phase will provide a customer specific map (e.g. AWS Control Framework) which will provide customers with a security recipe for building security and compliance at scale across AWS services.

CIM works to map features and resources to specific security controls requirements. Security, compliance, and audit personnel can leverage these documents as a reference to make certifying and accrediting of systems in AWS more efficient. The matrix outlines control implementation reference architecture and evidence examples, which meet the security control "risk mitigation" for the AWS customer environment.



Control-ID	Subpart-ID	Control Requirements	Implementation Guidance	Implementation Status	Defined in Stacks	Resources defined within Stack
AC-2		Account Management – Control: The organization:	Set AWS Identity and Access Management (IAM) to support common infrastructure personnel functions - Establishes IAM conditions for group and role membership	Partially	Yes	AWS::IAM::Role AWS::IAM::Group AWS::IAM::User, AWS::IAM::Policy
AC-2	a	a. Identifies and selects the following types of information system accounts to support organizational missions/business functions: [Assignment: organization-defined information system account types];	Establishes baseline AWS CloudTrail service enablement for data collection related to account usage, events and actions performed by personnel within the customer environment.		Yes	AWS::IAM::Role AWS::IAM::Group AWS::IAM::InstanceProfile
AC-2	b	c. Establishes conditions for group and role membership;			Yes	AWS::IAM::ManagedPolicy
AC-2	c	g. Monitors the use of, information system accounts.			Yes	AWS::CloudTrail::Trail

Figure 1: NIST SP 800-53 rev. 4 control security control matrix

- ### Security Services Provided (Inherency)
  Customers can reference and inherit security control elements from AWS based on their industry and the AWS associated certification, attestation and/or report (e.g. PCI, FedRAMP, ISO, etc.). The inheritance of controls can vary based on certifications and reports provided by AWS.

- ### Cross Service Security (Shared)
  Cross service security controls are those, which both AWS and the customer implement within the host operating system, and the guest operating systems. These controls include technical, operational and administrative (e.g. IAM, Security Groups, Configuration Management, etc.) controls, which in some case can be partially inherited (e.g. Fault

Tolerance). Example: AWS builds its data centers in multiple geographic regions, as well as across multiple Availability Zones within each region, offering maximum resiliency against system outages. Customers should leverage this capability by architecting across separate Availability Zones in order to meet their own fault tolerance requirements.

- Service Specific Security (Customer)
  Customer controls may be based on the system and services they deploy in AWS. These customer controls may also be able to leverage several cross service controls such as IAM, Security Groups and defined configuration management processes.
- Optimized IAM, Network and Operating Systems (OS) Controls
  These controls are security control implementations or security enhancements an organization should deploy based on leading security practices, industry requirements, and/or security standards. These controls typically cross multiple standards and service and can be scripted as part of a defined "secure environment" through the use of AWS CloudFormation templates and Service Catalog.

Phase 2 – Build a "Secure Environment"

This enables you to connect the dots on the wide range of security and audit services and features we offer and provide security, compliance and auditing personnel a straightforward way to configure an environment for security and compliance based on "least privileges" across the AWS customer environment. This helps align the services in a way that will make your environment secure and auditable real time verses within point in time or period in time.

- Access Management
  Create groups and roles - like developers, testers, or administrators - and provide them with their own unique credentials for accessing AWS cloud resources through the use of groups and roles.
- Network Segmentation
  Set up subnets in the cloud to separate environments (that should remain isolated from one another). For example, to separate your development environment from your production environment, and then configure network ACLs to control how traffic is routed between them. Customers can also set up separate management environments to ensure security integrity through the use of a Bastion host for limiting direct access to

production resources.

- Resource Constraints & Monitoring
  Establish hardened guest OS and services related to use of Amazon Elastic Compute Cloud (Amazon EC2) instances along with the latest security patches; perform backups of your data; and install antivirus, and intrusion detection tools. Deploy monitoring, logging and notification alarms.

- Data Encryption
  Encrypt your data or objects when they're stored in the cloud, either by encrypting automatically on the cloud side, or on the client side before you upload it.

Phase 3 – Enforce the Use of Templates

After creating a "secure environment," you need to enforce its use in AWS. You do this by enforcing Service Catalog. Once you enforce the Service Catalog, everyone with access to the account must create their environment using the CloudFormation templates you created. Every time anyone uses the environment, all those "secure environment" standard rules and/or constraints will be applied. This effectively operationalizes the remaining customer account security configurations of controls and prepares you for audit readiness.

Phase 4 – Perform Validation Activities

The goal of this phase is to ensure AWS customers can support an independent audit based on public, generally-accepted auditing standards. Auditing standards provide a measure of audit quality and the objectives to be achieved when auditing a system built within an AWS customer environment.

AWS provides tooling to detect whether there are actual instances of non-compliance. AWS Config gives you the point-in-time current settings of your architecture. You can also leverage AWS Config Rules, a service that allows you to use your secure environment as the authoritative criteria, to execute a sweeping check of controls across the environment. You'll be able to detect who isn't encrypting, who is opening up ports to the Internet, and who has databases outside a production VPC. Any measurable characteristic of any AWS resource in the AWS environment can be checked.

The ability to do a sweeping audit is especially valuable if you are working on an AWS account for which you did not first establish and enforce a secure environment. This allows you to check the entire account, no matter how it was created, and audit it against your secure environment standard. With AWS Config Rules, you can also continually monitor it, and the console will show you at any time which IT resources are and aren't in compliance. In addition, you will know if a user was out of compliance, even if for a brief period of time. This makes point-in-time and period-in-time audits extremely effective.

Since auditing procedures differ across industry verticals, AWS customers should review the audit guidance provided based on their industry vertical. If possible, engage audit organizations that are "cloud-aware" and understand the unique audit automation capabilities that AWS provides. Work with your auditor to determine if they have experience with auditing AWS resources; if they do not, AWS provides several training options to address how to audit AWS services through an instructor-led, eight-hour class including hands-on labs. For more information please contact: awsaudittraining@amazon.com.

Additionally, AWS provides several audit evidence gathering capabilities through secure read access along with unique API (Application Programming Interface) scripts which enable audit automation for evidence collection. This provides auditors the ability to perform 100% audit testing (versus testing with a sampling methodology).

SbD: How to Get Started

Here are some starter resources for you to get you and your teams ramped up:

- Take the self-paced training on "Auditing your AWS Architecture". This will allow for hands on exposure to the features and interfaces of AWS, in particular the configuration options that are available to auditors and security control owners.
- Request more information about how SbD can help email: aws-securitybydesign@amazon.com
- Be familiar with additional relevant resources available to you:
  - Amazon Web Services: Overview of Security Processes
  - Introduction to Auditing the Use of AWS Whitepaper
  - Federal Financial Institutions Examination Council (FFIEC) - Audit Guide

- SEC - Cybersecurity Initiative Audit Guide

Further Reading

- AWS Compliance Center: http://aws.amazon.com/compliance
- AWS Security by Design: http://aws.amazon.com/compliance/security-by-design
- AWS Security Center: http://aws.amazon.com/security
- FedRAMP FAQ: http://aws.amazon.com/compliance/fedramp
- Risk and Compliance Whitepaper: https://d0.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf
- Security Best Practices Whitepaper: https://d0.awsstatic.com/whitepapers/aws-security-best-practices.pdf
- AWS Products Overview: http://aws.amazon.com/products/
- AWS Sales and Business Development: https://aws.amazon.com/compliance/contact/
- Government and Education on AWS https://aws.amazon.com/government-education/
- AWS Professional Services https://aws.amazon.com/professional-services