Best Practices to Implement Security Controls for SWIFT Connectivity

SWIFT Customer Security Controls Framework (v2022) on AWS

SWIFT Customer Security Controls Framework (v2022) on AWS: Best Practices to Implement Security Controls for SWIFT Connectivity

Table of Contents

Abstract and introduction 1
Abstract 1
Are you Well-Architected? 1
Introduction 2
AWS Shared Responsibility model 3
Requirement 1 - SWIFT environment protection 5
SWIFT environment protection 5
Overall design for environment segmentation 5
Operating system privileged account control 8
Virtualization platform protection 9
Restriction of internet access 10
Requirement 2 - Reduce attack surface and vulnerabilities 12
Internal data flow security 12
Security updates 12
System hardening 14
Back-office data flow security 14
External transmission data protection 15
Operator session confidentiality and integrity 15
Vulnerability scanning 16
Application hardening 17
Requirement 3 - Physically secure the environment 18
Physical security 18
Requirement 4 - Prevent compromise of credentials 19
Password policy 19
Multi-factor authentication 19
Requirement 5 - Manage identities and segregate privileges 21
Logical access control 21
Token management 22
Physical and logical password storage 22
Requirement 6 - Detect anomalous activity 24
Malware protection 24
Software integrity 24
Database integrity 24
Logging and monitoring 25

Intrusion detection 26
Requirement 7 - Plan for incident response and information sharing 27
Cyber incident response planning 27
Security training and awareness 27
Penetration testing 27
Conclusion and contributors 28
Conclusion 28
Contributors 28
Further reading 29
Document Revisions 30
AWS Glossary 31

SWIFT Customer Security Controls Framework (v2022) on AWS

Publication date: **July 21, 2021** (*Document Revisions*)

Abstract

The SWIFT Customer Security Programme (CSP) was introduced to support SWIFT customers and drive industry-wide collaboration in the fight against cyber fraud. The CSP establishes a common set of security controls known as the Customer Security Controls Framework (CSCF) which is designed to help SWIFT users secure their local environments and to foster a more secure financial ecosystem.

The SWIFT Customer Security Controls Framework (CSCF) consists of both mandatory and advisory security controls for SWIFT users. Mandatory security controls establish a security baseline for the entire community, and must be implemented by all users on their local SWIFT infrastructure. With the shift to cloud computing, Appendix G of the latest CSCF provides guidance for users using digital connectivity.

The objective of this guide is to provide SWIFT customers with sufficient information and best practices to implement the CSCF security controls when implementing their SWIFT Client Connectivity Stack on AWS.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

With the current business landscape created by the COVID-19 pandemic, the Society for Worldwide Interbank Financial Telecommunication (SWIFT) issued the v2021 guidance for its users to implement its updated Customer Security Programme (CSP) and Customer Security Controls Framework (CSCF).

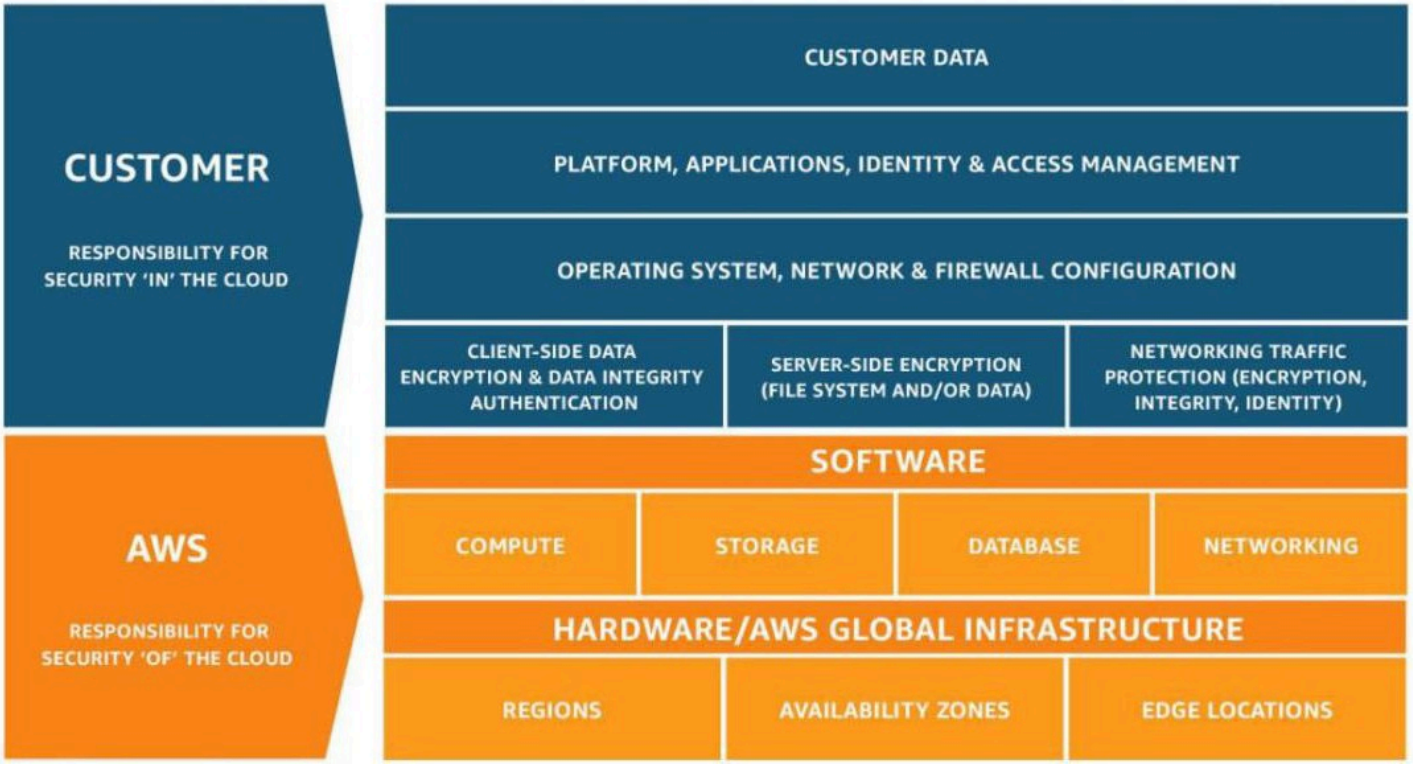
The latest CSP now requires a community-standard assessment for all users and all assessments submitted from 2021 onwards will require an independent assessment.

This document provides guidance for SWIFT connectivity deployed on the AWS Cloud and is structured on the 7 Requirement sections described in the CSP.

The latest v2022 guidance includes five changes from v2021. Three of the changes—Control 2.9, Control 6.2, and Control 6.3—are out of scope for Cloud Providers according to Appendix G of the CSP. A new advisory control(Control 1.5A) was added. It is nearly identical to Control 1.1 and the guidelines are the same. Finally, the scope of Control 1.2 has been extended to a new architecture type, but there are no changes to the AWS guidance.

AWS Shared Responsibility model

Security and Compliance is a shared responsibility between AWS and the customer. This shared model can help relieve the customer's operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates.



The AWS Shared Responsibility security model

AWS is responsible for the security and compliance **of** the cloud, or the infrastructure that runs all of the services offered in the AWS Cloud. Cloud security at AWS is the highest priority. AWS customers benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations and compliance frameworks. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services. This includes controls that maintain separation between customer resources and data, along with numerous other administrative, compliance, and security-related controls.

Customers are responsible for the security and compliance **in** the cloud, or the customer-configured systems and services provisioned on AWS. The customer assumes responsibility and management of the guest operating system (including updates and security patches) and other

associated application software, as well as the configuration of the AWS-provided security group firewall. This includes, but is not limited to, the following, as the customer's responsibility will depend on the services used, the integration of those services into their IT environment, and applicable laws and regulations:

- Customers are responsible for the compliant configuration of all system components, to include AWS resources and services, included in or connected to their cardholder data environments (CDE).
- Customers are responsible for the operating systems and installed applications on Amazon Elastic Compute Cloud (Amazon EC2), and network routing and configuration of associated virtual networking components.

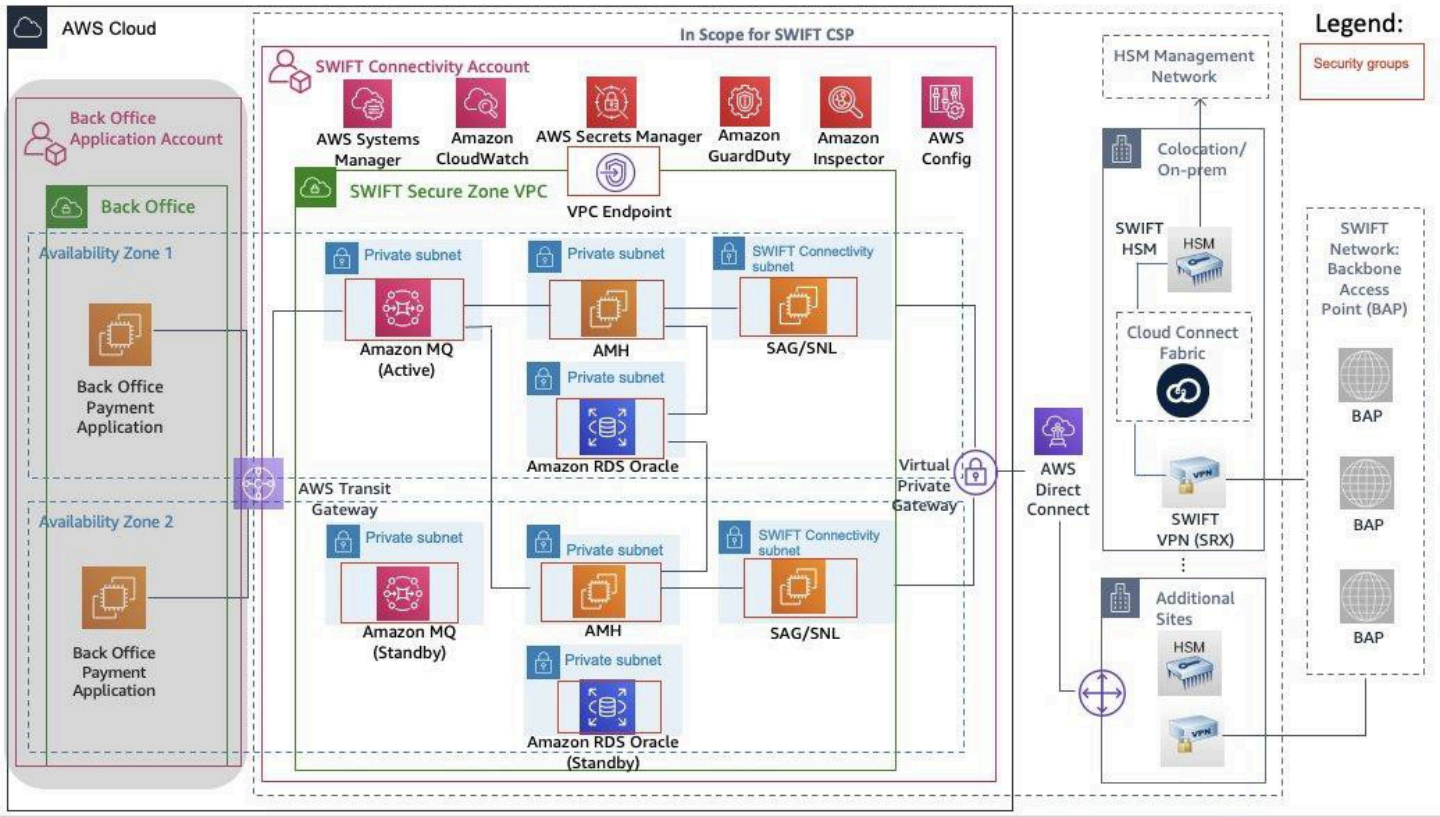
Requirement 1 - SWIFT environment protection

SWIFT environment protection

Overall design for environment segmentation

SWIFT mandates the various connectivity components (messaging interface, SwiftNet link (SNL), hardware security model (HSM), SWIFT connector, jump server, operator PC) to be deployed in a “secure zone”: a segmented and controlled environment that is bounded to the CSP control framework. Several AWS services can help you design and implement the secure zone. The following sections detail recommendations and guidance to help you meet the CSP control objectives using a combination of different AWS services.

Scope of the secure zone



Example of a SWIFT secure zone architected by AWS

SWIFT provides general guidance on which components should be in scope for the secure zone depending on the architecture the customers select to satisfy their business requirements. The preceding diagram shows an example of a SWIFT secure zone architected on AWS.

From the perspective of environment segregation, customers should use a dedicated AWS Account that is governed by AWS Organizations for running the SWIFT secure zone in a production environment. From the Well-Architected Security Pillar:

"We recommend that you organize workloads in separate accounts and group accounts based on function, compliance requirements, or a common set of controls rather than mirroring your organization's reporting structure. In AWS, accounts are a hard boundary, zero trust container for your resources. For example, account-level separation is strongly recommended for isolating production workloads from development and test workloads."

Account separation helps to ensure the ease of auditing and clear environment separation from back-office applications and other SWIFT connectivity stacks (Dev/Test).

Another important aspect of the scoping is that the SWIFT secure zone should contain only the resources that are intended to be deployed and run. On the preventative side, you can leverage AWS Organization Service Control Policies (SCP) to restrict the resource instantiation to the intended AWS services. You can create a dedicated deployment AWS Identity and Access Management (IAM) role with a least privilege policy and assign it to the infrastructure deployment pipeline to perform the infrastructure deployment.

On the detective side, you can use AWS Config to help you detect drifts and alterations that have happened in the SWIFT secure zone, and you can monitor AWS CloudTrail for any unintended changes. If you use AWS CloudFormation for infrastructure as code deployment, you can leverage the drift detection functionality in CloudFormation to detect unintended deployment in the SWIFT secure zone.

You can leverage a landing zone to manage and govern the SWIFT secure zone accounts. A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organization can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. Building a landing zone involves making technical and business decisions across account structure, networking, security, and access management in accordance with your organization's growth and business goals for the future.

You have a few options for creating your landing zone on AWS. You can choose a managed service to orchestrate your environment, or work with an AWS Partner to build your own. AWS offers AWS Control Tower, a managed service that sets up a landing zone based on multi-account best practices, centralizes identity and access management, and establishes pre-configured governance rules for security and compliance.

Protection of the secure zone — boundary protection

Amazon Virtual Private Cloud (Amazon VPC) enables you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. Security Groups act as a stateful firewall for resources within an Amazon VPC, controlling both inbound and outbound traffic at the virtual network interface. Security Groups can be used to restrict traffic by IP address, port, and protocol, and can help satisfy elements of CSP 1.1. By default, Security Groups allow all outbound connections; you are responsible for configuring specific outbound connection rules for CSP compliance. Network access control lists (ACLs) are an additional layer of security for VPCs that acts as a stateless firewall for controlling traffic in and out of one or more subnets.

For example, the AWS Quick Start guide for SWIFT client connectivity provides these Security Groups:

Table 1 – Security Groups in SWIFT on AWS Quick Start reference deployment

Security Group	Inbound (Port)	Outbound (Port)
Alliance Message Hub (AMH)	AMH Admin Desktop IP (8443)	SAG/SNL (48002/3), Amazon MQ (61617), RDS (1521)
Alliance Gateway (SAG) / SNL	AMH (48002/3)	SWIFT IP range (0-65536), HSM IPs (1792)
Amazon MQ	AMH (61617)	n/a
Amazon RDS	AMH (1521)	n/a
VPC Endpoint	AMH (443), SAG/SNL (443)	n/a

Access to the secure zone systems

Traditionally, jump servers are deployed to the SWIFT secure zone for accessing various SWIFT components. Creating, maintaining, managing access, security hardening, and patching on these jump servers becomes undifferentiated heavy lifting. Access to the secure zone systems is simplified with AWS System Manager Session Manager, as it removes the need for the jump server in the secure zone. Session Manager is a fully managed AWS capability that enables you to manage your Amazon EC2 instances, on-premises instances, and virtual machines (VMs) through an interactive, one-click, browser-based shell, or through the AWS Command Line Interface (AWS CLI) without the need for a bastion or jump server.

If you prefer a traditional jump server setup and GUI access, Amazon WorkSpaces is the preferred choice. Amazon WorkSpaces enables you to provision virtual, cloud- based Microsoft Windows or Amazon Linux desktops, called Workspaces, for your users. Amazon WorkSpaces eliminates the need to procure and deploy hardware or install complex software. You can quickly add or remove users as your needs change. Users can access their virtual desktops from multiple devices or web browsers. This Amazon Workspace virtual desktop would be deployed to the secure zone VPC and act as a bastion host to access the rest of the secure zone's components. This virtual desktop can be restricted such that it can be accessed through the corporate network only. For example, you could use IP access control to only allow certain IP addresses to access the virtual desktop. Refer to How to secure your Amazon WorkSpaces for external users. Various components in the SWIFT secure zone can be accessed from the virtual desktop.

Segregation from general enterprise IT services

The guidance of using your AWS account to achieve segregation is detailed in the the section called "Scope of the secure zone" section of this document.

Operating system privileged account control

The goal for this control is to prevent individual users from having excess privilege to exploit the operating system on the host. The general best practice is to prevent human access to the EC2 host directly, except in a break glass situation. ("Break glass" refers to a quick means for a person who does not have access privileges to gain access when necessary.)

Ideally, installation and configuration of the SWIFT software should not be run manually on the EC2 host itself. Software installation should be done through an automated AMI pipeline. Per the Well-Architected Framework: Financial Service Industry Lens:

"Adopt immutable infrastructure practices with no human access to better meet your audit and compliance needs. You will be able to version control your infrastructure and handling failure will be a routine and continuous way of doing business."

You can leverage Amazon EC2 Image Builder as part of the AMI pipeline implementation for building the SWIFT application AMIs. If regular operation and maintenance tasks are required to be performed on the EC2 host, AWS Systems Manager Document Automation and Run Command can be used. Proper IAM role and least privilege policies should be assigned for individuals or groups who need to perform such functions.

Virtualization platform protection

Customer environments are logically segregated to prevent users and customers from accessing resources not assigned to them. You maintain full control over who has access to your data. Services such as EC2, that provide virtualized operational environments, ensure that customers are segregated from one another and prevent cross-tenant privilege escalation and information disclosure via hypervisors and instance isolation.

Different instances running on the same physical machine are isolated from each other via the hypervisor. The Amazon EC2 firewall resides within the hypervisor layer, between the physical network interface and the instance's virtual interface. All packets must pass through this layer, so an instance's neighbors have no more access to that instance than to any other host on the internet, and can be treated as if they are on separate physical hosts. The physical random access memory (RAM) is separated using similar mechanisms.

Customer instances have no access to physical disk devices, but instead are presented with virtualized disks. The AWS proprietary disk virtualization layer automatically erases every block of storage before making it available for use, which protects one customer's data from being unintentionally exposed to another. Customers can further protect their data using traditional file system encryption mechanisms, or, in the case of Amazon Elastic Block Store (Amazon EBS) volumes, by enabling AWS managed disk encryption.

The AWS Nitro System is the underlying platform for the next generation of EC2 instances that enable AWS to innovate faster, further reduce cost for customers, and deliver added benefits such as increased security and new instance types.

The Nitro System provides enhanced security that continuously monitors, protects, and verifies the instance hardware and firmware. Virtualization resources are offloaded to dedicated hardware

and software, minimizing the attack surface. Nitro System's security model is locked down and prohibits administrative access, eliminating the possibility of human error and tampering.

Nitro is currently available across many different EC2 instance types.

Restriction of internet access

This control objective is designed to limit exposure to an internet-based attack. This is a common security control, and best practice for many organizations. There are many well-established patterns to restrict internet access.

One simple measure that can help you meet this control objective is to completely disable internet access for the SWIFT secure zone. The SWIFT secure zone can be operated without internet access. In AWS terms, the VPC in the secure zone does not have an internet gateway attached to the VPC. An SCP can be implemented to disallow internet gateway and prevent additional components for getting internet access. Refer to the example service control policy, prevent any VPC that doesn't already have internet access from getting it.

If internet access is required, customers can consider leveraging different traffic filtering solutions like Gateway Load Balancer, AWS Network Firewall, or open-source Squid proxy. These solutions can filter unnecessary egress traffic to the internet and prevent unintentional internet ingress traffic.

Advanced adversaries may attempt to use DNS-based data exfiltration to leak data or perform command and control within a VPC that is otherwise removed from the internet. Use Route 53 Resolver to control internet name resolution, in conjunction with Amazon GuardDuty rules for DNS tunneling. Amazon Route 53 Resolver DNS Firewall is a managed firewall that enables customers to block DNS queries made for known malicious domains and to allow queries for trusted domains. DNS Firewall provides more granular control over the DNS querying behavior of resources within your Amazon VPCs.

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following data sources:

- VPC Flow Logs
- AWS CloudTrail management event logs
- CloudTrail S3 data event logs
- DNS logs

If you use AWS DNS resolvers for your EC2 instances (the default setting), then GuardDuty can access and process your request and response DNS logs through the internal AWS DNS resolvers. For example, the `Trojan:EC2/DNSDataExfiltration` (https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-ec2.html#trojan-ec2-dnsdataexfiltration) finding informs you that the listed EC2 instance in your AWS environment is running malware that uses DNS queries for outbound data transfers.

Requirement 2 - Reduce attack surface and vulnerabilities

Internal data flow security

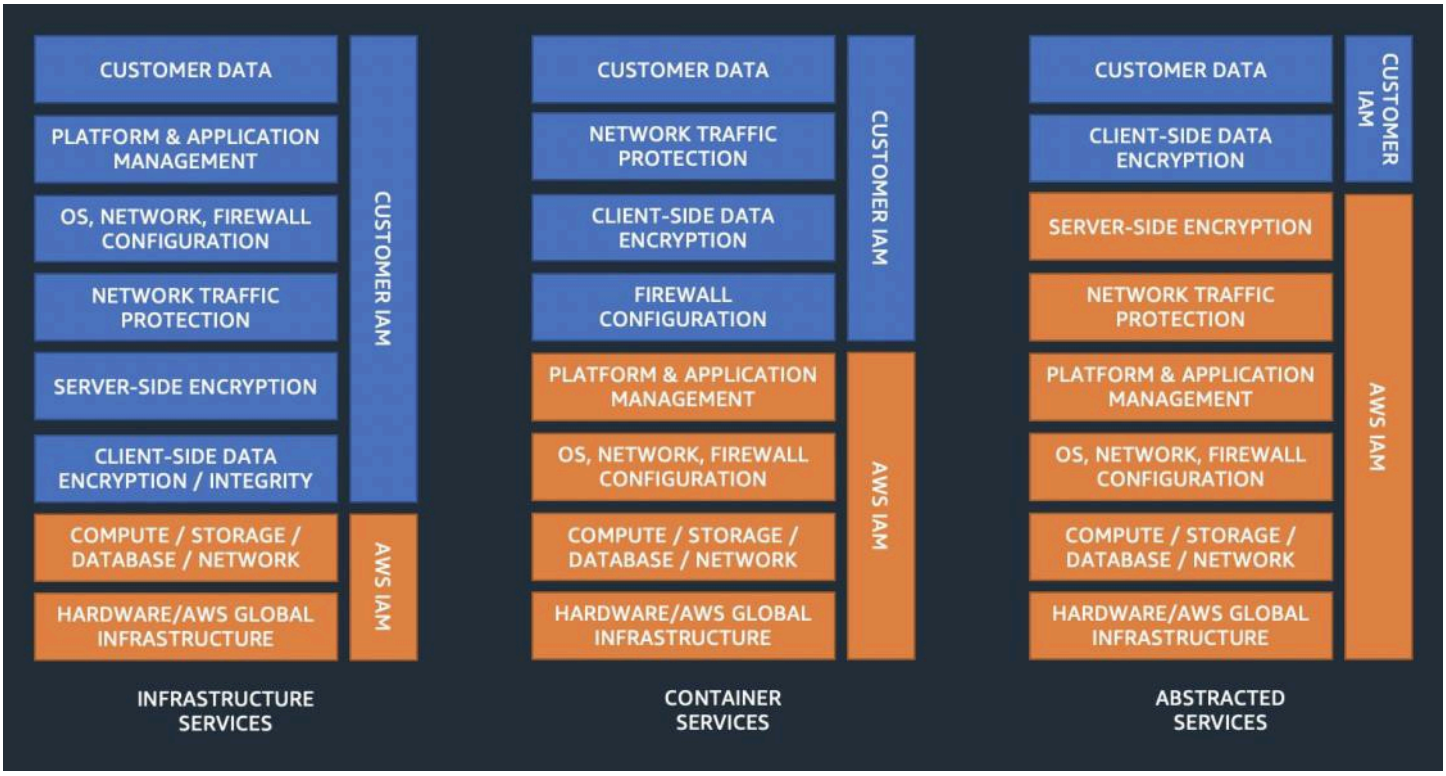
The internal data flow security control objective can be accomplished by securing server-to-server traffic using authentication with one-way Transport Layer Security (TLS), or two-way TLS. This control is applicable for the communications between all applications, messaging interfaces, and database connections. The communication between SWIFT applications, Alliance Messaging Hub (AMH), Alliance Access (SAA), Alliance Gateway (SAG), and SWIFTNet Link (SNL), have the mechanism enforced in the application configuration. From the AWS perspective, here is the list of guidance by architecture components:

- **Amazon MQ** — Amazon MQ requires a user ID and password to connect to the message broker. Amazon MQ protocols have TLS 1.2 enabled. Refer to **Encryption in Transit for Amazon MQ**.
- **Amazon RDS for Oracle** — Amazon RDS for Oracle supports authentication and TLS 1.2 and higher encryption in transit. Refer to Oracle Secure Sockets Layer.
- **AWS Secrets Manager** — The user ID, password, and connection information can be safely stored in AWS Secrets Manager. The password should be rotated regularly. AWS Secrets Manager supports password rotation natively.
- **AWS Certificate Manager Private Certificate Authority** (CA) — For issuing, managing, and renewal of certificates when using two-way TLS for communication.
- **AWS Systems Manager Session Manager** — Communication between EC2 instances and AWS Systems Manager Session Manager is always encrypted with TLS. An IAM assume role with MFA and source IP condition enforced should be used for gaining access to the AWS Systems Manager.
- **Security Group** — Security groups act as a stateful firewall for the components in the SWIFT secure zone. It should be set up so that only the intended traffic is allowed between components.

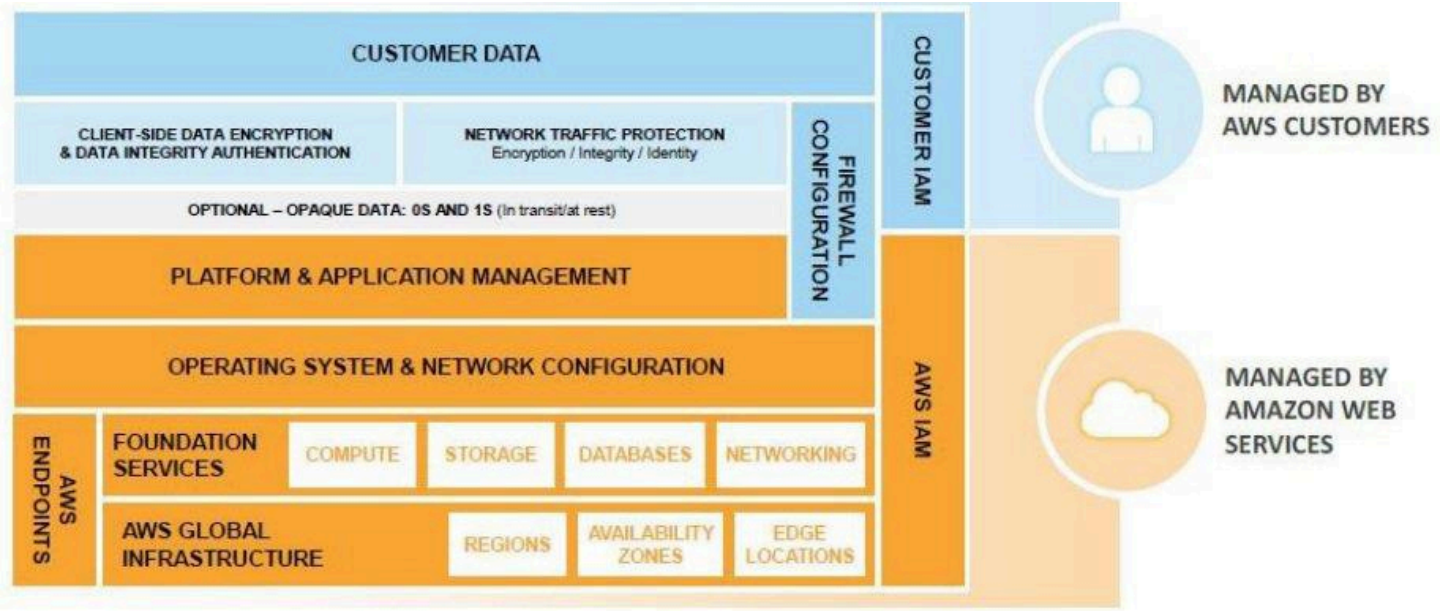
Security updates

Leveraging container services greatly simplifies the security updates process for many customers. By using Amazon MQ and Amazon RDS for Oracle in the SWIFT secure zone, you do not need to worry about maintaining and patching the underlying EC2 instances. AWS is responsible for

maintaining updates for the underlying EC2 instances for AWS Managed Services. In this model, it is your responsibility to upgrade the Amazon MQ broker version and Amazon RDS for Oracle major and minor versions. Refer to the following diagrams:



The AWS Shared Responsibility Model for infrastructure services, container services, and abstracted services



AWS/customer management plan

You are responsible for maintaining security updates for the EC2 instances in the SWIFT secure zone. You can opt to use immutable infrastructure and blue/green deployment strategy for deploying security updates for EC2 instances. In this topology, you would have a "golden" AMI pipeline to create the up-to-date operating system (OS) image, and another AMI pipeline to bundle the golden image with SWIFT applications and third-party libraries. The updated AMI is deployed and tested in the Dev / Test environment, and is subsequently promoted to the production environment. This testing and promotion process can be orchestrated in a pipeline created with AWS CodePipeline, or your existing CI / CD pipeline.

System hardening

Customers are responsible for maintaining the security configuration standards for their resources provisioned on AWS. These standards must be consistent with industry- accepted system hardening standards, and include the customer's configuration of AWS services. AWS has published extensive security guides for the platform and individual services. The base set of these are:

- Center for Internet Security (CIS) Benchmark for AWS
- CIS Benchmarks for EC2 instance types
- AWS Trusted Advisor
- AWS Security Checklist
- AWS Well-Architected Framework: Security Pillar

There are various options to archive this control objective. For example, you can launch the pre-hardened EC2 instance using a CIS-provided AMI in the AWS Marketplace.

Another option is to leverage the AMI pipeline to build the hardened EC2 AMI. If you are using EC2 Image Builder for the AMI pipeline, EC2 Image Builder provides EC2 Image Builder STIG components for EC2 hardening.

Back-office data flow security

This control objective is similar to the section called "Internal data flow security", but this is an advisory control and primarily focuses on the edge connection to the secure zone. A backend

payment application on-premises connecting to the Amazon MQ that resides in the SWIFT secure zone VPC is an example for such a connection. The principles and the requirements of the two controls are the same, so the guidance is the same as 2.1.

If you use a hybrid architecture, such as one in which the back-office applications reside in an on-premises data center and the SWIFT secure zone is on AWS, consider using an internet protocol security (IPsec) VPN tunnel or Media Access control Security (MACsec) to encrypt the networking traffic in between. Refer to:

- AWS Site-to-Site VPN
- AWS Direct Connect MAC Security

Besides encryption in transit using IPsec and MACsec, it is important to use Security Groups and Network ACLs (NACLs) to control the connection to the edge of the SWIFT secure zone. For example, if the entry point of the secure zone is Amazon MQ, the Security Group for Amazon MQ should allow incoming connections from the on- premises IP CIDR range, while other components in the secure zone should not allow incoming connection from the same range. NACLs can be applied in addition to Security Groups to provide an additional layer of access control.

External transmission data protection

AWS KMS is a managed service that makes it easy for you to create and control AWS KMS keys (CMKs), the encryption keys you can use to encrypt your data. AWS KMS CMKs are protected by hardware security modules (HSMs) that are validated by the FIPS 140-2 Cryptographic Module Validation Program, except in the China (Beijing) and China (Ningxia) Regions.

This control objective concerns the compromising of trusted backup data and loss of sensitive data confidentiality.

In an AWS environment, data backup is automatically encrypted by the same Key Management Service (KMS) encryption key as your data store. For more information, refer to Encryption for backups in AWS. Encryption of the data stores in the SWIFT secure zone (such as RDS Oracle and EBS volumes) can satisfy this requirement.

Operator session confidentiality and integrity

There are two aspects to consider for meeting control objectives:

- Ensure access to the jump server / SWIFT component hosts are properly authenticated and encrypted.
- Implement proper timeout / inactivity lockout on the operator sessions so it limits the minimal timeframe necessary to perform business-as-usual duties.

When AWS Systems Manager Session Manager is used in place of the jump server for operator access, the first implementation guidance is satisfied. All AWS-provided services, including AWS Systems Manager Session Manager, require authentication, and the traffic is required to be encrypted using TLS. Choose one of the options for logging session activity in your AWS account with the appropriate encryption mechanism enabled.

In addition to providing information about current and completed sessions in the Systems Manager console, Session Manager provides you with options for logging session activity in your AWS account. This enables you to:

- Create and store session logs for archival purposes.
- Generate a report showing details of every connection made to your instances using Session Manager over the past 30 days.
- Generate notifications of session activity in your AWS account, such as Amazon Simple Notification Service (Amazon SNS) notifications.
- Automatically initiate another action on an AWS resource as the result of session activity, such as running an AWS Lambda function, starting an AWS CodePipeline pipeline, or running an AWS Systems Manager Run Command document.

AWS System Manager Session Manager supports idle session timeout. It enables you to specify the amount of time to allow a user to be inactive before a session ends. By default, sessions time out after 20 minutes of inactivity. You can modify this setting to specify that a session times out between one and 60 minutes of inactivity.

Vulnerability scanning

You can employ Amazon Inspector to help you achieve the vulnerability scanning control objective. Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. With Amazon

Inspector, you can automate security vulnerability assessments throughout your development and deployment pipelines, or on EC2 instances. Amazon Inspector also offers an agent that you can install on Linux or Windows EC2 instances to scan for vulnerabilities. The Amazon Inspector agent also monitors the behavior of the EC2 instances, including network, file system, and process activity.

In addition, there are many AWS Partner solutions available in this space which you can use.

Application hardening

Guidance for securing Messaging and Communication interfaces for SWIFT applications (AMH, SAA, SAG, SNL) are in SWIFT Knowledge Centers.

Requirement 3 - Physically secure the environment

Physical security

AWS manages the physical infrastructure for the hosted environments, and physical security requirements are inherited from the AWS global infrastructure. Customers are responsible for the physical security and data classification of media exported or transferred out of the AWS environment, but not for the physical security of data stored within AWS. Under CSCF Control 3.1, customers are also responsible for the physical security and management of any physical HSM devices they use that connect to resources provisioned in the AWS Cloud. Customers are also still responsible for the physical security of any physical locations in which they store, process, or transmit messages. These might include corporate offices, call centers, or retail locations.

Requirement 4 - Prevent compromise of credentials

Password policy

AWS Secrets Manager is the recommended service to safely store the passwords that are utilized in the SWIFT secure zone.

Refer to *Physical and logical password storage* in this document for details on AWS Secrets Manager. It is your responsibility to define the password policy and enforce it.

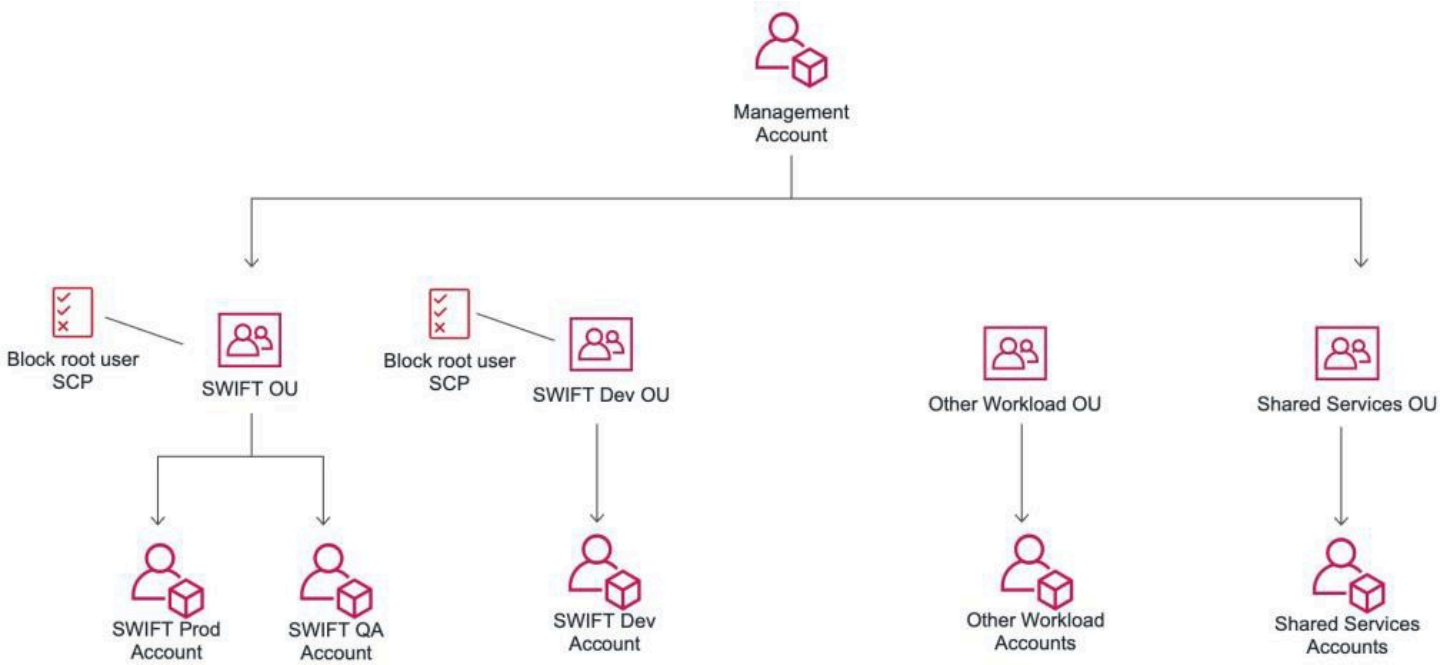
To access AWS in a typical enterprise environment, use a federated login to AWS from the corporate identity provider. In this scenario, the authentication and password policy of the SWIFT operator PC / Accounts is enforced within the corporate security policy.

You are responsible for configuring operating system level access to EC2 instances. By leveraging AWS Systems Manager Session Manager, you would no longer have to manage passwords for human operators on your EC2 instances. For the service accounts that run applications on the OS, it is your responsibility to maintain the password policy for the users.

Multi-factor authentication

Multi-factor authentication (MFA) is mandated for the jump server for accessing the SWIFT secure zone. In the case where AWS Systems Manager Session Manager is utilized for accessing the SWIFT secure zone, you can create an AssumeRole trust policy to enforce MFA when federation is used. Refer to MFA-for-SAML.

The AWS root user of the AWS account also has access to the SWIFT secure zone using AWS Systems Manager Session Manager by default. You can disable the root user using SCPs to block the root user. Refer to the example, “Block service access for the root user” on the Example service control policies page. You should enable MFA for the root user. Refer to Enable MFA on the AWS account root user. The following diagram illustrates an example multi-account structure with SWIFT accounts and OU with “Block service access for the root user” SCP applied.



Example of a multi-account structure with SWIFT accounts and OU with "Block service access for the root user" SCP applied

Requirement 5 - Manage identities and segregate privileges

Logical access control

Here are two sample roles for managing the SWIFT infrastructure and components:

- The SWIFT instance operator role gives privilege to individuals who require EC2 access to install and troubleshoot SWIFT software like AMH, SAG and SNL.
- The SWIFT infrastructure role enables you to control the states of the infrastructure components like EC2, Amazon MQ, and Amazon RDS for Oracle, and the ability to view the CloudWatch Logs.

When you create IAM policies, follow the standard security advice of granting *least privilege*, or granting only the permissions required to perform a task. Determine what users (and roles) need to do, and then craft policies that allow them to perform *only* those tasks.

Start with a minimum set of permissions, and grant additional permissions as necessary. This is more secure than starting with permissions that are too lenient and trying to tighten them later.

IAM provides several options to help you refine the permissions that you grant.

- **Understand access level groupings** – You can use access level groupings to understand the level of access that a policy grants. Policy actions are classified as List, Read, Write, Permissions management, or Tagging. For example, you can choose actions from the List and Read access levels to grant read-only access to your users. To learn how to use policy summaries to understand access level permissions, refer to Use access levels to review IAM permissions.
- **Validate your policies** – You can perform policy validation using IAM Access Analyzer when you create and edit JSON policies. We recommend that you review and validate all of your existing policies. IAM Access Analyzer provides over 100 policy checks to validate your policies. It generates security warnings when a statement in your policy allows access it considers to be overly permissive. You can use the actionable recommendations that are provided through the security warnings as you work toward granting least privilege. To learn more about policy checks provided by IAM Access Analyzer, refer to IAM Access Analyzer policy validation.
- **Generate a policy based on access activity** – To help you refine the permissions that you grant, you can generate an IAM policy that is based on the access activity for an IAM entity (user or

role). IAM Access Analyzer reviews your AWS CloudTrail logs and generates a policy template that contains the permissions that have been used by the entity in your specified time frame. You can use the template to create a managed policy with fine-grained permissions, then attach it to the IAM entity. That way, you grant only the permissions that the user or role needs to interact with AWS resources for your specific use case. To learn more, refer to Generate policies based on access activity.

- **Use last accessed information** – Another feature that can help with least privilege is *last-accessed information*. View this information on the **Access Advisor** tab on the IAM console **Details** page for a user, group, role, or policy. Last-accessed information also includes information about the actions that were last accessed for some services, such as Amazon EC2, IAM, Lambda, and Amazon Simple Storage Service (Amazon S3). If you sign in using AWS Organizations management account credentials, you can view service last- accessed information in the **AWS Organizations** section of the IAM console.

  You can also use the AWS CLI or AWS API to retrieve a report for last-accessed information for entities or policies in IAM or Organizations. You can use this information to identify unnecessary permissions so that you can refine your IAM or Organizations policies to better adhere to the principle of least privilege. For more information, refer to Refining permissions in AWS using last accessed information.
- **Review account events in AWS CloudTrail** – To further reduce permissions, you can view your account's events in AWS CloudTrail **Event history**. CloudTrail event logs include detailed event information that you can use to reduce the policy's permissions. The logs include only the actions and resources that your IAM entities need. For more information, refer to Viewing CloudTrail Events in the CloudTrail Console in the *AWS CloudTrail User Guide*.

Token management

The customer is responsible for having a controlled process for distributing, assigning, and revoking physical tokens.

Physical and logical password storage

AWS Secrets Manager is the recommended service to safely store the passwords that are utilized in the SWIFT secure zone, such as user IDs and passwords for connecting Amazon RDS Oracle and Amazon MQ, and the password for connecting SWIFT AMH to the SAG cluster. AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources.

The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. All secrets stored in AWS Secrets Manager should be encrypted with AWS KMS, and have well-defined resource policies for the secrets.

Secrets Manager integrates with AWS KMS to encrypt every version of every secret with a unique data key that is protected by an AWS KMS key. This integration protects your secrets under encryption keys that never leave AWS KMS unencrypted. It also enables you to set custom permissions on the AWS KMS key and audit the operations that generate, encrypt, and decrypt the data keys that protect your secrets.

Requirement 6 - Detect anomalous activity

Malware protection

AWS is responsible for the deployment and management of antivirus and anti-malware solutions on AWS managed services such as Amazon RDS, Amazon ECS, and AWS Fargate. Customers inherit the security and compliance for AWS managed operating systems. You are responsible for configuring and running appropriate antivirus software on any applicable EC2 instance in which you have access to and responsibility for the underlying operating system. The AWS Marketplace offers numerous products.

Software integrity

Software integrity checks are generally embedded in SWIFT software components (AMH, SAA, SAG, SNL). If additional software and components are required to run in the SWIFT secure zone, consider getting a third-party file integrity monitoring tool in the AWS Marketplace.

The immutable infrastructure mentioned in the the section called “Security updates” section of this document also plays a part in this security control objective. In an immutable infrastructure environment setup, no individuals should be allowed to perform any software changes or modification directly on the live SWIFT system.

Database integrity

Similar to the software integrity requirement, database integrity checks are enabled for SWIFT software components (AMH, SAA, SAG).

From the environment perspective, use a dedicated database instance for SWIFT connectivity purposes. The database should be encrypted with KMS keys. You should have designated users and roles for ensuring separation of duty for the database tables and schemas. Use AWS Secrets Manager to store the password for the database user login, and use the password rotation capability to rotate the database password periodically. You can implement detective controls for password checkout from AWS Secrets Manager.

Regarding database integrity, Amazon RDS creates and saves automated backups of your database (DB) instance during the backup window of your DB instance. Amazon RDS snapshots

are automatically encrypted with the same encryption key that was used to encrypt the source Amazon RDS database. For the IAM permission aspect, grant least-privileged IAM policy to authorized roles that are required to perform infrastructure operations on the database instances. For details, refer to Identity-based policy examples for Amazon RDS.

Logging and monitoring

The overall goal is to capture security-related logs, configure alarms for suspicious events, and establish a plan to remediate the incident. Per SWIFT CSP implementation guidelines, enable logging on jump servers, firewall logs, databases, messaging interfaces, and command line history.

GuardDuty can help you detect unauthorized and unexpected activity in your AWS environment. You can use it to analyze and process data from AWS CloudTrail event logs, VPC Flow Logs, and DNS logs to detect anomalies involving the following AWS resource types:

- EC2 instances
- S3 buckets

From the application and middleware components perspective, this control objective can be covered by configuring logging and monitoring for different services in the SWIFT secure zone.

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts, workloads, and data stored in Amazon S3. It also uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning (ML) to identify threats more accurately.

Logging in SWIFT secure zone by component:

- **Amazon RDS Oracle** — Alert logs, trace log, audit logs, trace files, listener logs, and Oracle Management Agent logs. Refer to Oracle database log files.
- **Amazon MQ** — General logging and audit logging. Refer to Configuring ActiveMQ logs.
- **AWS Systems Manager Session Manager** — SSM Session Manager Session activity. Refer to Logging session activity.
- **VPC Flow Log** — Capture information about the IP traffic going to and from network interfaces in your VPC. Refer to Log and View Network Traffic Flows.
- **AWS CloudTrail** — Log account activity-related action across the AWS infrastructure. Refer to Turning on CloudTrail in Additional Accounts.

- **SWIFT Application Logs** — AMH, SAA and SAG / SNL logs.

All logging mechanisms have integration with Amazon CloudWatch Logs, which can be used to store, access, and monitor the behavior in the SWIFT secure zone. You can leverage create metrics from log events using filters to be alerted for suspicious activities. If you want to perform analytics on the logs generated, you can use Amazon CloudWatch Logs Insights, Amazon Athena, or Amazon OpenSearch Service. You can choose to integrate all AWS logs into your existing security information and event management (SIEM) and log archival solutions.

Intrusion detection

Amazon GuardDuty (described in the the section called “Logging and monitoring” section of this document) can be leveraged to help you meet this control objective. This can not only detect anomalies in the networking traffic that is happening in the VPC, but can also detect suspicious activities on the AWS account level.

AWS Network Firewall can also be used to act as a network intrusion prevention system/intrusion detection system (IPS / IDS) within AWS as well as other IPS / IDS offerings from the AWS Marketplace.

Requirement 7 - Plan for incident response and information sharing

Cyber incident response planning

Preparation is critical for a successful incident response program. The *AWS Security Incident Response Guide* whitepaper provides you with an overview of the fundamentals of responding to security events within a customer's AWS Cloud environment. AWS provides a number of security tools and services to allow organizations to track, monitor, analyze, and audit events.

The Well Architected Security pillar also provides guidance to customers on Incident Response.

Security training and awareness

AWS offers training to develop critical security skills to simplify your organization's journey to the AWS Cloud, protect data and applications, and innovate with confidence.

AWS Training and Certification has created the AWS Ramp-Up Guide: Security for AWS Cloud Security, Governance and Compliance Professionals and other AWS Ramp-Up Guides to help build your knowledge of the AWS Cloud. Each expertly curated guide features free training, classroom courses, videos, whitepapers, certifications, and other information.

Penetration testing

The AWS Acceptable Use Policy describes permitted and prohibited behavior on AWS, and includes descriptions of prohibited security violations and network abuse. AWS customers are welcome to carry out security assessments or penetration tests against their accounts on AWS infrastructure without prior approval for eight services, listed in Penetration Testing under "Permitted Services." All penetration testers and vulnerability scan managers must understand and comply with the AWS Customer Support Policy for Penetration Testing.

AWS Security Competency Partners offer an array of security offerings like network and infrastructure security, vulnerability and configuration analysis, application security, and security engineering.

Conclusion and contributors

Conclusion

In summary, this document provides comprehensive best practice guidance for SWIFT connectivity stack deployed on the Amazon Web Services (AWS) Cloud. You can leverage the best practices to help you achieve your agility, security, and cost saving goals, while gaining confidence in deploying a SWIFT CSCF-CSP compliant environment.

Contributors

Contributors to this document include:

- Jack Iu, Global Account Solution Architect, Amazon Web Services
- Syed Shareef, Security Solutions Architect, Amazon Web Services
- Henry Su, Customer Solutions Manager, Amazon Web Services
- Liam Wadman, Security Solutions Architect, Amazon Web Services

Further reading

For additional information, refer to:

- AWS Quick Start: SWIFT on AWS
- SWIFT CSCF (Requires login)
- Architecting SWIFT Connectivity on Amazon Web Services (AWS) (blog post)

Document Revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Update whitepaper	Updated for v2022 changes from SWIFT	October 4, 2022
Initial publication	Whitepaper published.	July 21, 2021

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.