Architecting for PCI DSS Scoping and Segmentation on AWS

Identify and Minimize Your Scope Using Appropriate Segmentation Controls

May 2019

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities 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

This paper provides guidance on how to properly define the scope of your Payment Card Industry (PCI) Data Security Standard (DSS) workloads running on the AWS Cloud platform and how to define segmentation boundaries in between your in-scope and out-of-scope resources using cloud native Amazon Web Services (AWS) services.

The paper also discusses how to validate the implemented segmentation controls as mandated by PCI DSS requirement 11.3.4 of PCI DSS v3.2.1[1]. The paper is based on the PCI Security Standards Council (PCI SSC) published Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation.

This paper is intended for engineers and solution builders, but also serves as a guide for Qualified Security Assessors (QSAs) and internal security assessors (ISAs) to better understand the different segmentation controls available within the AWS platform and the associated scoping considerations.

Introduction

Software-defined-networking on AWS transforms the scoping process for applications, compared to on-premises environments. Additional segmentation controls available on AWS go above and beyond just network segmentation. Therefore, thoughtful design of the applications and selection of security impacting services for implementing required controls can substantially reduce the number of systems and services in the cardholder data environment (CDE). To severely limit and even eliminate connected-to systems and services, leverage Amazon Elastic Compute Cloud (Amazon EC2) instances to granularly define Amazon Virtual Private Cloud (VPC) Security Groups and web service interfaces with incorporated firewalls.

PCI DSS Scoping Process on AWS

Similar to a traditional scoping process, scoping an AWS application begins with the cardholder data (CHD) flow.[2] However, there is an immediate difference because many hops for CHD are purpose-built services, such as Amazon API Gateway or AWS WAF. These services have well-defined connection configurations and have been assessed as part of the AWS PCI DSS Level 1 Service Provider assessment. Additionally, these AWS endpoints are RESTful web service interfaces that are protected by firewall functionality (part of the AWS PCI DSS scope) and serve as segmentation boundaries for services not receiving CHD.

General purpose Amazon Elastic Compute Cloud (Amazon EC2) instances have much more clearly defined network connections than on-premises servers. EC2 instances are treated as standalone network hosts, not as routers or a network gateway. Security groups operate not only between subnets but also on each instance interface, providing interface-level network rule granularity as opposed to the subnet-level granularity of traditional network firewall appliances. Network configuration through security groups can enforce much greater granular network access control that cannot be circumvented by the instance network configuration.

There are further differences on how PCI DSS scope fans out within the AWS platform based on the nature of used AWS services.

AWS services can be broadly categorized into one of the following groups:[3]

- Infrastructure services, such as Amazon EC2 instances
- Container services, such as Amazon Relational Database Service (Amazon RDS)
- Abstracted services, such as Amazon Simple Storage Service (Amazon S3)

The strategies for establishing a minimal PCI DSS scope cover the preceding three broad categories of AWS services. It also addresses the scoping and segmentation consideration for a hybrid environment, where the in-scope PCI DSS workloads may be spread across your on-premises data center and AWS Cloud.

Security Controls

Any organization that stores, processes, and transmits cardholder data (CHD) is generally required by the acquirer agreement to meet and demonstrate compliance with PCI DSS requirements. For a service provider, this may be a customer requirement. This global security standard maintained by PCI Security Standards Council includes a prescriptive set of IT security requirements (also known as *controls*) to protect sensitive credit card information. Organizations are responsible for correctly identifying and defining their Cardholder Data Environment (CDE), also referred to as the scope of the PCI DSS assessment. The CDE consists of people, processes, and technologies that interact with CHD or have a security impact and thus are in scope for PCI. In this paper, we only focus on the scope of the technology aspect of the CDE.

Segmentation

Segmentation is colloquially referred to as *PCI DSS Requirement 0*. In other words, segmentation is first used to limit scope to the systems that are critical to the payment flow before addressing any other PCI DSS requirement. Traditionally, organizations use network segmentation as a key control to limit their PCI DSS in-scope environment and protect it from the rest of their IT infrastructure. This approach does not imply that organizations must not secure their out-of-scope infrastructure; rather, they have more flexibility in their choice of security controls as well as different validation requirements for their out-of-scope infrastructure.

The Guidance for PCI DSS Scoping and Network Segmentation, categorizes IT infrastructure and resources into the following segments with respect to PCI DSS Scope:

- **CDE Systems**: These system components store, process, or transmit CHD or sensitive authentication data (SAD). These components are in scope of PCI DSS and tend to bring other resources in scope (i.e., any resources communicating with these CDE systems also become in-scope systems).
- **Connected-to and/or security impacting systems:** These system components have direct or indirect restricted connection to CDE systems and provide some sort of management and security services to CDE systems. These components can additionally assist in fulfilling one or more PCI DSS requirements or help establish segmentation boundaries. These are in-scope systems; however, they do not extend the scope to other resources, merely from a connection perspective.
- **Out-of-scope systems:** These system components do not meet the preceding criteria and do not have an impact on the security or configuration of CDE systems. These systems are not considered in scope.

AWS Cloud Platform Considerations for Solutions

AWS Cloud platform has certain characteristics, such as scalability, disposable resources, traceability, security controls automation, continuous validation, and testing. These unique characteristics, along with various business advantages like the Operating Expenditure (OPEX) model, make cloud adoption advantageous for most organizations.[4] The benefits of the cloud are realized when you use these cloud-specific characteristics to design payment application infrastructure.

Shared Responsibility Model

Security and Compliance is a shared responsibility between AWS and the customer. This shared model can help relieve 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. Primarily, AWS is responsible for the security *of* the cloud and you are responsible for protecting the confidentiality, integrity, and availability of the data *in* the cloud, and for meeting specific business requirements for information protection.

Make sure that you understand the shared responsibility model before you embark on your compliance journey. The shared responsibility varies depending on the abstraction incorporated in that specific AWS service. Service abstraction is inversely proportional to your responsibility. Make sure to evaluate every service used in your environment to understand the impact of using the service on your overall PCI DSS scope. This approach helps you understand exactly what you must do to meet your compliance obligations. AWS suggests adopting a Service Adoption Framework (SAF) to formally evaluate each AWS service and document the decision process and required controls that you are responsible to meet as per organizational security and compliance requirements.

Virtualization of Traditional Network Controls

Traditional on-premises network controls like VLANs are implemented differently on AWS because layer 2 networking is transparent to end users. The network on AWS is a software-defined network (SDN) mimicking the traditional network construct.

Within the AWS platform, Amazon Virtual Private Cloud (Amazon VPC) represents a logically isolated section of the AWS Cloud where resources can be launched in a virtual network. Additionally, it is common for resources providing similar functionality or under similar scope to span across multiple subnets across different Availability Zones providing redundancy. Therefore, VPCs and subnets are more grouping constructs than segmentation controls.

Elasticity

AWS resources allocated to an application such as compute or storage, can be scaled horizontally based on demand. AWS Auto Scaling monitors your applications and automatically adjusts compute capacity to maintain steady, predictable performance at the lowest possible cost. Such AWS resources tend to be ephemeral in nature and can be short lived. Other AWS Managed Services, such as AWS Lambda and Application Load Balancers (ALBs), scale vertically to accommodate the resource requirement. The segmentation controls you design must be able to accommodate the elastic and transient nature of the cloud environment. Design these controls so that they remain enforced as the infrastructure changes; otherwise, it may lead to incorrect scope definition.

Abstracted Services and API-based infrastructure

Many AWS offerings are provided as a managed service, meaning AWS manages the infrastructure for you. Among these, the abstracted AWS services only communicate

over web service API calls. Web service APIs use inherent network segmentation controls in addition to other controls, such as authentication, authorization, and data integrity. This ensures that only data from authorized entities are exchanged between the calling system and the service. By design, these abstracted services are secured to ensure that data is not shared in between different instances of the service unless otherwise explicitly allowed. These services communicate among themselves and other services over access-controlled APIs. This configuration is provided as part of the service and meets the layer 3–4 network controls provided by firewalls. You must design application layer-based segmentation and traffic filtering controls as part of your share of the overall shared responsibility model.

Whenever possible, use web API or loosely coupled services for application functionality and controls. For example, select a log consolidation service, like Amazon CloudWatch, that makes web API calls to forward log data rather than an agent that maintains an open TCP/IP connection.

Automation

With automation, most infrastructure and application changes can be implemented without any manual intervention. This provides agility, decentralizes the change management process, and expedites the deployment process. The segmentation controls must also be automated to the extent possible so that the segmentation controls are applied in tandem with the infrastructure and application change controls. This approach preserves the proper scope boundary. By eliminating manual checks, automation also assists in detecting when segmentation controls are modified and when proper remediation steps can be implemented, such as reinforcing the controls or alerting someone in near-real-time to analyze the cause and effect of changes.

Hybrid Infrastructure

As your organization migrates workloads to the AWS Cloud platform, you may run a hybrid infrastructure for a period of time. In other words, you may have workloads running both on your on-premises data center and on AWS Cloud platform. Your segmentation controls must consider this hybrid infrastructure and any communication between the on-premises and AWS Cloud platform.

Decision Flow - PCI DSS Scope Identification

A decision flow assists in correctly identifying the PCI DSS scope within your organization. This process starts with correctly identifying the flow of CHD within your organization environment.

Once the CHD flow has been correctly identified, the next step is to identify all in-scope resources in your environment.

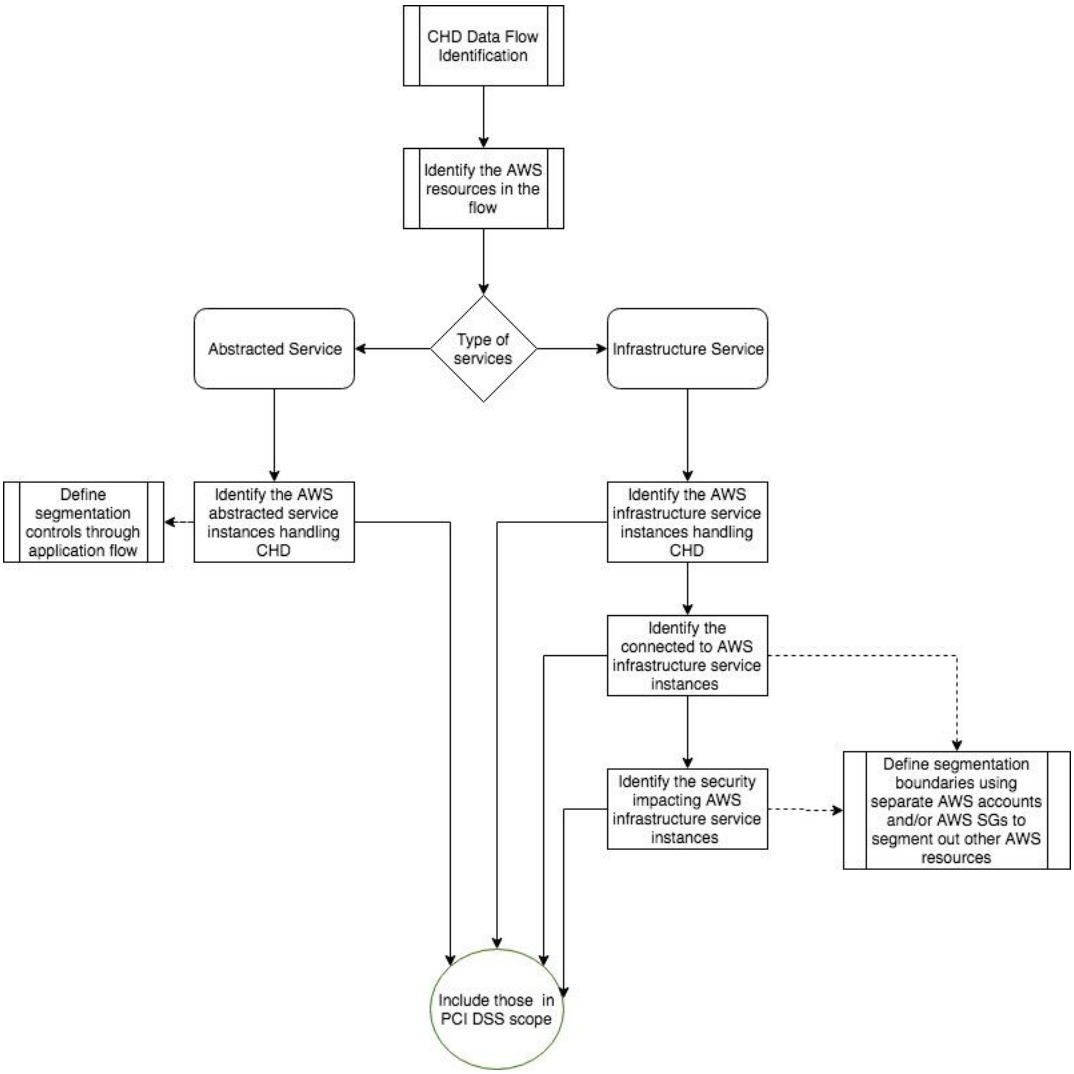

Figure 1: Decision flow for correctly identifying the PCI DSS scope and associated segmentation boundaries

The preceding figure captures the decision flow you must follow to correctly identify the PCI DSS scope. It also assists in correctly defining segmentation boundaries at different

stages of the flow to segregate other AWS resources from in-scope resources and minimize the PCI DSS scope.

Step 1: Identify the CHD Flow

Before you can start the process of defining your PCI DSS and designing segmentation boundaries, you must have an accurate understanding of the CHD flow within your organization. To correctly identify the CHD flow, you must identify and define the whole lifecycle of CHD within your organization. These include the path of consumption or entry of CHD in your environment, the subsequent processing and storing of the CHD and eventually to the secure destruction, devaluation, or exit of the CHD from your environment.

Step 2: Identify All In-Scope Resources in Your Environment

Identify the various types of AWS resources making up the CHD flow. These resources are those receiving, processing, storing and/or transmitting CHD. It is critical to define what is and what is not in scope as part of your analysis so that your ISA and QSA can clearly understand what services their assessment should be limited to when conducting their audit.

Step 3: Categorize the Systems

This step categorizes systems into abstracted and infrastructure services. The scope identification and segmentation of those resources are based on different types of connection. Although the infrastructure services primarily communicate with each other over a network (OSI Layer 3–4) connection, the only form of communication for abstracted services are the data connections established over some form of API (OSI layer 7).

After identifying those different types of AWS resources, you can identify the exact PCI DSS scope.

Abstracted Services

For AWS abstracted services, the resources in scope are not the endpoints of the AWS services that are used for accessing the service. The only resource in scope would be the particular instantiation of the AWS service handling CHD. For example, an organization may have many Amazon DynamoDB tables provisioned, but only a subset

of those tables is used to store/process CHD. In this case, only those RDS tables used for storing CHD would fall into the PCI DSS scope of the organization.

Containerized Services

Like abstracted services, the endpoints of containerized services are not in scope. The instantiation of the service handling CHD is in scope. However, the instantiation may include additional layers. For example, and RDS instance may have multiple tables and only a subset of those may be handling CHD. Those tables are in scope along with the RDS schema under which the tables reside. This is because for abstracted service, you are responsible for the security of the platform, which in this case is the database platform of the RDS instance.

Infrastructure Services

For infrastructure services, the scope identification process includes additional steps. In addition to identifying the AWS resources handling CHD, you must also identify the connected-to and the security impacting AWS resources. For example, an EC2 instance running a web service handling CHD would definitely be in scope. However, it may bring other connected-to resources in scope, such as a reporting server having a network connection for fetching non-CHD data reports. It would additionally bring other AWS resources having an impact on its security in scope, which may include directory resources providing authentication and authorization services.

Step 4: Design Segmentation Boundaries

After all of the in-scope AWS resources have been identified, design segmentation boundaries to ensure that all other AWS resources are properly segmenting them to exclude them from the PCI DSS scope. For AWS abstracted services, this segmentation is primarily controlled by the application and associated application code that controls the flow of the CHD. For infrastructure-based resources along with application-level segmentation, you must also design network level segmentation.

Design Segmentation for the Cloud

This section explains the various segmentation boundaries that you can design based on the principle of using cloud features to protect cloud services and achieving defense in depth. You can achieve these boundaries at various layers of the AWS platform and then combine them with each other to reduce the PCI DSS in-scope systems to the minimum as required for a secure and functional CDE.

Segmentation is not a PCI DSS requirement; however, you can use segmentation to restrict the PCI scope to the minimum number of resources possible. Therefore, after identifying resources that fall under the PCI DSS scope, design the segmentation boundaries to ensure out-of-scope systems are properly segregated.

AWS Account Layer

An individual AWS account provides the highest level of segmentation boundary that can be achieved on the AWS platform. By design, all resources provisioned within an AWS account are logically isolated from resources provisioned in other AWS accounts, even within your own AWS Organizations. Using an isolated account for PCI workloads is a core best practice when designing your PCI application to run on AWS. It provides a level of security that is not possible in on-premises implementations.

You may reduce the PCI DSS scope by segmenting in-scope and out-of-scope resources into multiple AWS accounts. This would ensure avoiding accidental scope creep as logical account level isolation can only be changed by establishing explicit communication channels in between resources from separate AWS accounts. This approach also helps reduce the impact by not allowing changes to architecture and controls in out-of-scope AWS accounts from adversely affecting the security of in-scope resources in other AWS accounts. The following diagram shows a proposed AWS multi-account architecture with respect to designing PCI DSS scope:

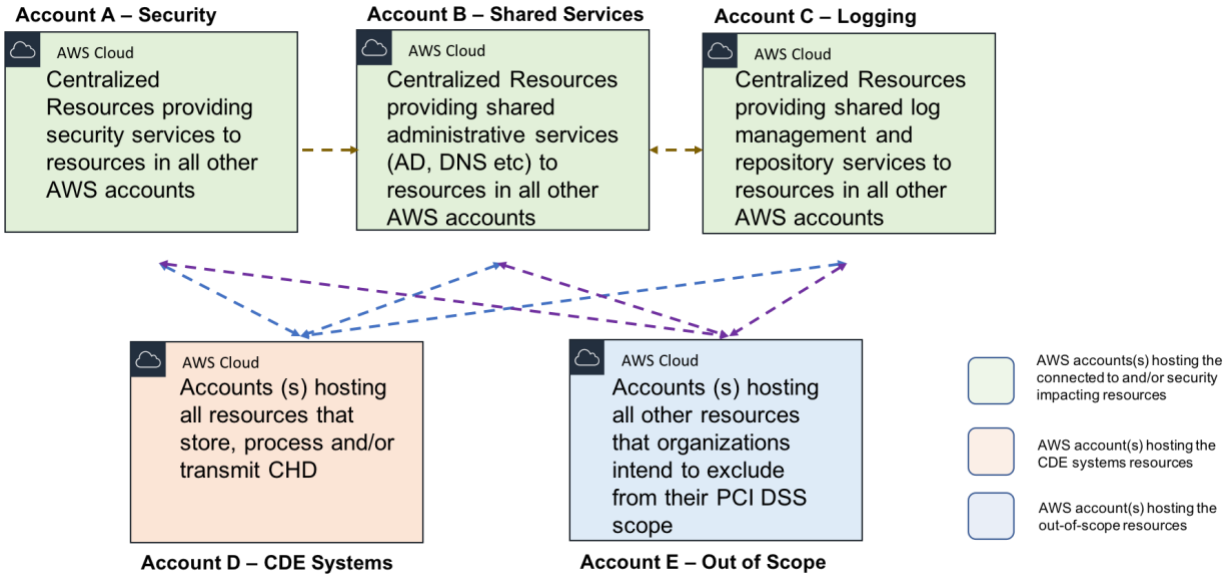

Figure 2: Multi-account architecture for restricting PCI DSS scope

Components of Multi-Account Architecture

Figure 2 shows multi-account architecture designed per AWS best practices, as published through the AWS landing zone architecture, where resources providing similar functionality are grouped within AWS accounts. This grouping enables sharing resources that provide security and management functionality among in-scope and out-of-scope AWS resources, while ensuring the PCI DSS scope is restricted. In this scenario, the accounts are grouped together based on the PCI DSS scope of the resources, CDE systems, connected-to and/or security impacting systems, and out-of-scope systems.

CDE Systems Account (Account D in Figure 2)

Provision your IT infrastructure that would handle CHD in one or more dedicated AWS accounts. Do not use this account to provision any other IT resources.

CDE system accounts include the following examples:

- Compute and network resources receiving and/or transmitting CHD, from and to external entities
- Compute resources processing CHD
- Storage resources storing CHD at rest

Consider the CDE systems account as the most sensitive part of the PCI DSS infrastructure. System components in this account tend to bring other resources in scope. This means that any system components that they connect to are also considered to be in scope for PCI, irrespective of the functionality or the data involved in the communication. Communication outside of this account should only be highly restricted (port and protocol restriction) to only necessary resources in other AWS accounts, for system management or business requirement purposes. Implement strict change management processes to ensure that changes in this account do not negatively impact the segmentation boundaries and the overall PCI DSS scope of the organization. AWS Config can be used to track all changes to a particular AWS resource, which could then trigger an AWS Lambda function to generate an alert, auto remediate security control deviations or orchestrate a variety of other steps to implement change control processes.

Connected-to and/or Security Impacting Systems Account (Account B and C in Figure 2)

Use separate AWS accounts to provision other system components that are required to manage the CDE systems and/or provide security functionality to the CDE systems.

Security impacting systems accounts include the following examples:

- jump server or bastion host
- directory services
- antivirus and anti-malware services
- vulnerability scanning services
- any other services that help manage the CDE systems or are required to fulfill one or multiple PCI DSS requirements.

Connected-to systems accounts include the following examples:

- applications or services extracting data from the CDE systems to fulfill some business purpose, such as business intelligence applications that extract non-CHD data.
- accounts used for provisioning such connected-to and/or security impacting systems.
- central security and shared services accounts

The resources in these accounts are not required to be isolated from other non-CDE AWS resources of the organization. Services provided by resources in these accounts can be consumed by other AWS resources that do not fall into PCI scope. These augmented security controls, in turn can assist in centralizing security services and controls and help maintain the same baseline across all of your workloads.

Per the AWS published landing zone architecture, you should provision a dedicated AWS account for centralized logging, separate from the security or shared service account. This account should collect all types of logs from all systems and applications from all AWS accounts. This helps implement centralized management and restricted access of all log data, including security logs. In Figure 2, *Account C - Logging account* represents the central log management AWS account. Only systems in this account that are part of log management and are used to satisfy PCI DSS requirement 10[3] would be in scope for PCI DSS.

Out-of-scope systems account (Account E in Figure 2)

Use separate AWS accounts to provision IT infrastructure that neither require any form of connectivity and do not provide any service to CDE systems. This separation ensures that they are adequately isolated from PCI in-scope systems by design, through inherent AWS account isolation.

Although communication in between CDE systems and connected-to systems is permitted, you must ensure that there is no CHD flowing through those channels. Otherwise, the connected-to systems must be re-categorized into CDE systems. This scenario can result in a cascading scope creep. Monitoring also helps prevent bad actors from using the connected-to systems for data exfiltration from a compromised CDE system.

The multi-account architecture also helps group AWS accounts for implementing standard security controls across similar accounts. These accounts can be clubbed together to form individual Organization Units (OU) as part of AWS Organizations and policy-based account management. You can apply service control policies (SCPs) to the OUs that centrally control AWS service use across multiple AWS accounts. Service Control Policies (SCPs) restrict the permissions that AWS Identity and Access Management (IAM) policies can grant to entities in an account, such as IAM users and roles. For example, you can use SCPs to prohibit provisioning of non-PCI DSS compliant AWS services within in-scope-systems accounts.

Network Layer (OSI Layer 3–4)

A security group is an Amazon Virtual Private Cloud (VPC) feature that provides stateful network layer traffic filtering that works on the principle of an explicit deny. Security groups can equate to a host-based firewall and are associated with network interfaces of an EC2 instance. Security groups are the segmentation boundaries at the network layer and you should design PCI DSS scoping strategy at network layer around security groups. Security groups can be further used to restrict traffic flow in between instances to help meet with PCI DSS v3.2.1 requirements 1.2[5] and 1.3.[6]

Use security groups to restrict network communication based on required port, source, and destination addresses required to segment CDE systems from other connected-to systems.

You can attach security groups to Auto Scaling groups so that they are applied to and removed from instances in the group as the groups scale out and scale in. In between two peered VPCs, the security groups can be chained together so that one security group can reference the other security group as the source or destination, instead of referring to hardcoded IP addresses. This design helps automate the security group architecture and provides scalability by controlling peering traffic via security group membership instead of CIDR ranges.

By default, security groups do not meet PCI DSS requirement 1.2.x because they permit all outbound traffic. Rather you should configure security group defaults to limit inbound and outbound traffic only to which is necessary and deny all other traffic.

You can enable connection in between your EC2 instances in out-of-scope AWS accounts with a connected-to systems account without impacting the overall PCI DSS scope. Further, as VPC peering connections are non-transitive, the peering connection between CDE systems accounts and connected-to systems account does not extend the connectivity into out-of-scope systems accounts as long as there is no peering connection in between CDE systems and out-of-scope systems accounts.

The following diagram demonstrates the use of security groups for achieving segmentation. As shown, security groups primarily define the network segmentation irrespective of other network constructs, such as VPCs and network segments.

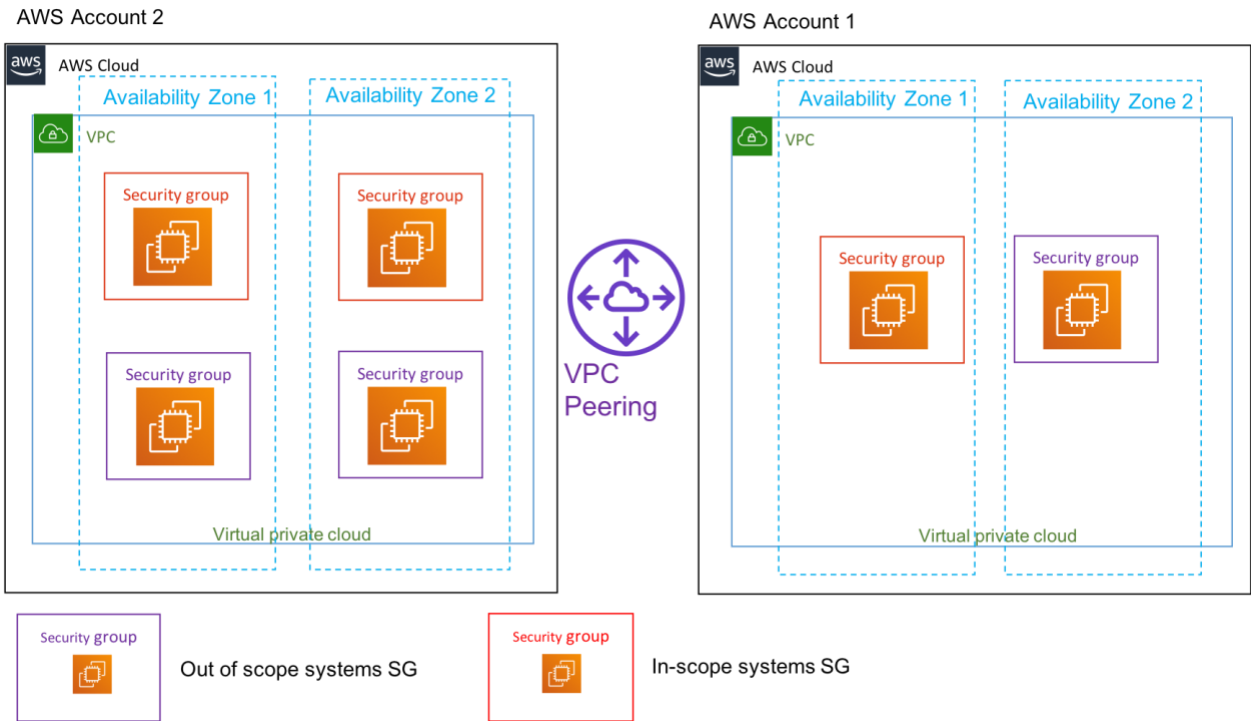

Figure 3: Network segmentation using security groups

Application Layer (OSI Layer 7)

At this layer, the application handling CHD must manage the CHD flow and defining segmentation boundaries. AWS provides numerous web API-based abstracted services, such as AWS Lambda, Amazon S3, and Amazon DynamoDB, that your organization can consume to enable business functionality without having to worry

about managing servers and EC2 instances. The only form of communication that can take place with these abstracted services are web service API calls to their endpoints. API calls happen over the application layer or OSI layer 7.

Segmentation for abstracted AWS services

Abstracted services implement network isolation by design. The connections between these services are data connections rather than network connections. For scoping, the focus is placed on the type of data traversing the connection.

Provided an abstracted service does not handle CHD, you can use content aware controls to sufficiently prove that CHD cannot traverse over from your CDE to these abstracted services, and exclude them from the PCI scope.

Adopt proper architectural designs, including data filtering and monitoring, to ensure that these services do not store, process, or transmit CHD, and therefore can be safely considered out-of-scope for PCI.

Segmentation with Amazon API Gateway

For customer-defined web API-based services, such as serverless applications, you can use Amazon API Gateway to broker connections in between the CDE resources/services and other web-based services. Amazon API Gateway can act as a "front door" to applications for accessing data, business logic, or functionality from backend services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, other supported AWS services, or any web/mobile applications. You can use API Gateway to front end the communication from CDE systems and services hosted on AWS. In this case, the connections from CDE systems and services are terminated by API Gateway and a new connection is established from API Gateway to the destination system or services. Then, any web API-based resource outside AWS that communicates with CDE systems via API Gateway can be excluded from the PCI DSS scope as long as the resource does not handle CHD or provide any security service to CDE systems. The instance of API Gateway service is in scope as a connected system. Since this is a PCI DSS validated service, you do not have to worry about maintaining and validating the service for PCI DSS. Although API Gateway provides a secure, access-controlled web service API mechanism, applications are responsible for validating incoming data, including monitoring for unexpected CHD from a CDE.

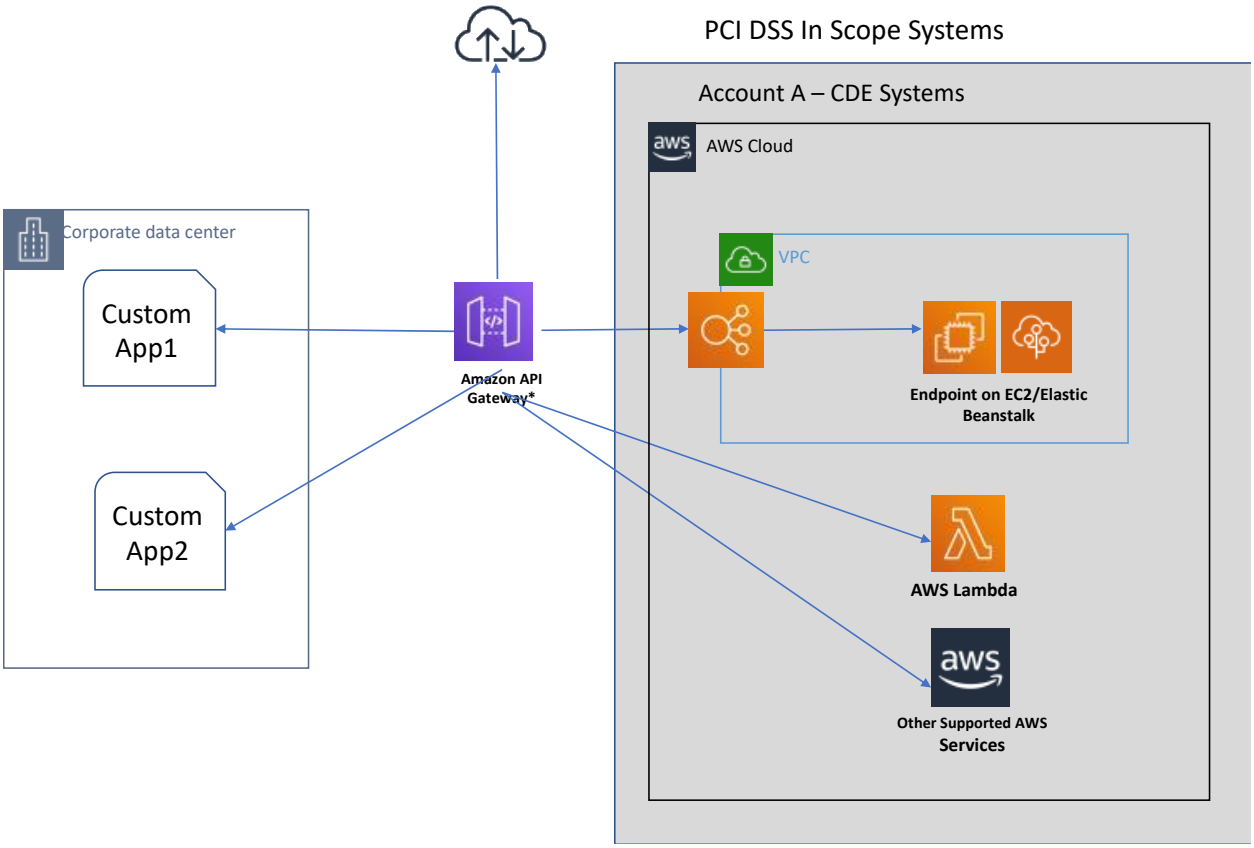

Figure 4: Segmentation using Amazon API Gateway for abstracted service

Scoping and Segmentation for Docker Containerized Workloads

Organizations are using containers to quickly, reliably, and consistently deploy applications that are transparent in the deployment environment. Containers allow you to easily package an application's code, configurations, and dependencies into building blocks that deliver environmental consistency, operational efficiency, developer productivity, and version control. Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS. If you are running or architecting for running containerized PCI in-scope applications, you must ensure that they are properly scoped and segmented.

A task is a logical group of running containers. Amazon ECS tasks supports two task launch types:

- Amazon EC2 instance type: This type allows you to run your containerized applications on a cluster of Amazon EC2 instances that you manage.
- AWS Fargate type: This type allows you to run your containerized applications without the need to provision and manage the backend infrastructure.

Container Scoping

A combination of task definition and task launch types determines scoping of containers. Ideally, create separate tasks to group PCI in-scope and out-of-scope containers. For EC2 instance launch types, assign them to separate clusters. This assignment ensures all EC2 instances in the cluster running PCI in-scope container tasks are only in scope for PCI. You can assign a security group to the ECS cluster and design security group ACLs to restrict network communication to only other in-scope systems and/or isolate the cluster from any out-of-scope system components. After you isolate the cluster, design task level isolation or segmentation. ECS supports task networking, which allows you to define security group rules for individual tasks by isolating container tasks to their own network interface.

For details on how to create a task definition using `awsvpc` mode, see the AWS Documentation on Task Networking.

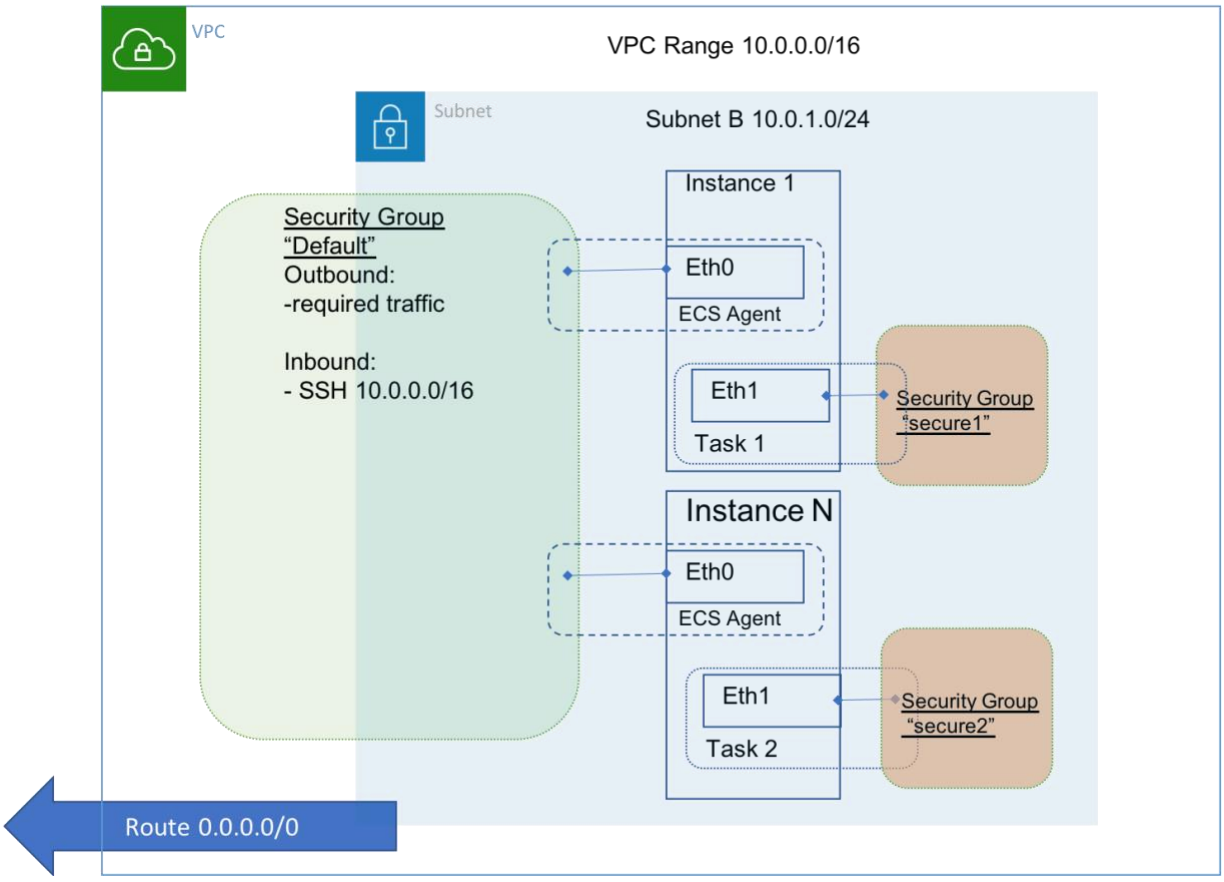

Figure 5: ECS task level segmentation

Although network segmentation can be achieved within tasks, as a best practice, avoid running both in-scope and out-of-scope applications as part of a single ECS task. Doing so may introduce overheads around defining proper scope boundaries.

For the AWS Fargate task launch type, tasks are the lowest construct so you do not have to worry about cluster assignment and scope. Group tasks running in-scope containers and use `awsvpc` network mode in combination with security group rules to segment and/or restrict communication in between in-scope and out-of-scope tasks.

Scoping Guidance for Hybrid Environments

Organizations running a hybrid architecture, (i.e., workloads running both on the AWS platform and on-premises data centers) must consider the connections to and from their on-premises system components and AWS CDE resources.

Consider the following scenarios:

Scenario 1: PCI resources hosted on AWS have network connections to resources in an on-premises data center.

Scenario 2: PCI resources hosted in on-premises data center have network connections to resources on AWS.

Scenario 3: PCI resources hosted both in on-premises data center and on AWS.

For Scenario 1 and 2, the scope of the non-CDE resources, irrespective of where they are hosted, is determined by the type of connectivity these non-CDE resources have with in-scope systems.

- If the non-CDE resources have a direct connectivity to CDE resources, then those on-premises or AWS resources are in scope for PCI DSS.
- If the non-CDE resources have a connection to connected-to systems, then those on-premises or AWS resources can be considered out of scope for PCI DSS, provided they do not provide any security and/or management services CDE systems. Further, if those out-of-scope systems are compromised, there is no way for that system to be used to further compromise their CDE. For example, AWS Systems Manager can manage both on-premises servers and in-scope EC2 instances. In this scenario, the on premises servers are out of scope provided they do not interact with CHD or have any other connection to CDE.

To limit scope, consider the following guidelines:

- Restrict network connections from on-premises networks only to non-CDE resources.
- If connectivity is required with CDE resources, implement proper security controls and designs to prevent transitive network connections that may bring unwanted network and system components in scope.
- Implement multiple segmentation controls to ensure that the scope boundaries are not altered erroneously. These controls can be implemented through a combination of access control rules defined and implemented using on-premises stateful firewall technologies, security groups, and backed up by network access control list (network ACL) for achieving defense in depth.

For Scenario 3, group CDE resources into either on-premises or AWS for simplicity of scope determination. If this grouping is not feasible, carefully ensure all CDE and

connected-to systems are identified both on-premises and on AWS and the connection to those systems are correctly identified.

Scoping and Segmentation Validation

Per PCI DSS requirement 11.3.4[7], you must perform network and application layer testing. This step corresponds to the “Assess and authorize controls” phases of the system lifecycle approach for security and privacy as outlined in NIST SP 800-37.

As per the PCI DSS published Information Supplement: Penetration Testing Guidance, the scope for penetration testing includes the entire CDE perimeter and any critical systems.[8] This applies both to the external perimeter (public-facing attack surfaces) and the internal perimeter of the CDE (LAN-LAN attack surfaces). Additionally, include any remote access vectors, such as VPN connections to CDE hosted on AWS.

This form of penetration testing to validate segmentation boundaries is only required if your CDE consists of infrastructure services such as EC2 instances. For abstracted services, scanning and penetration testing of AWS services and endpoints is covered by the PCI DSS service provider assessment of AWS.

For infrastructure services, such as EC2 instances, security groups form the base segmentation boundaries. If the configuration of the security group is reliable, then carrying out a penetration testing exercise to validate security group segmentation is valueless. To validate penetration boundaries, a better approach is to try penetrate into an in-scope EC2 instance from an out-of-scope EC2 instance since security groups define the segmentation boundaries and they are associated with network interfaces on EC2 instances.

For a hybrid environment, the source of penetration testing can be an out-of-scope network segment of the physical on-premises network and the target is the in-scope EC2 instances.

For in-scope AWS abstracted services, the CHD flow and segmentation boundaries are controlled by the application and the application code. Hence, the focus should be on application testing to validate the segmentation boundaries as designed within the application. This focus ensures that the CHD flow and the PCI DSS scope is maintained by the application as depicted in the CHD flow diagram.

> Penetration testing is not required for APIs of abstracted AWS services that are PCI compliant. Penetration testing of AWS endpoints is an AWS responsibility.

When preparing for penetration testing on the AWS Cloud platform, make sure that you understand the AWS acceptable usage policy. AWS recommends vetting potential penetration testing vendors/third-parties to determine their overall cloud experience and penetration testing experience on AWS platform and technologies. For more information on conducting vulnerability assessments or penetration testing to and from AWS resources, see the Vulnerability and Penetration Testing guidelines.

Preventive Controls

In addition to the periodic penetration testing exercise as mandated by PCI DSS, you must also have proactive security controls in place to prevent any unauthorized modification of the segmentation controls.

This section provides information on monitoring the status of segmentation controls implemented. This monitoring helps ensure that the defined PCI DSS scope is not violated intentionally or erroneously. In case of any violation, the preventive controls must be designed so that respective stakeholders are notified as early as possible and remediation steps can be taken immediately. As the security posture matures, automate the majority of responses so that deviations can be remediated without any human intervention in a near-real-time basis. The following are some of the ways you can monitor your segmentation boundaries:

- Validate all security group rules against the planned CDE scope periodically.
- Manage all security group and network configurations with a change control process.
- Monitor all security group rule changes in the CDE systems account.
- Monitor all VPC peering connections to the CDE systems account.
- Monitor all configuration changes to in scope API gateways.
- Implement data loss prevention controls on any connected-to systems to prevent CHD leakage and validate scope boundary.

You can automate responses using AWS Config. AWS Config provides you with resource configuration history and configuration change notifications to help enable

governance and security. You can create custom AWS Config rules to monitor all changes to security groups, VPC peering connections, API gateways, and any other resources within AWS that enforce segmentation boundaries. Attach AWS Config rules to appropriate AWS Lambda responders to evaluate deviations and trigger auto remediation once a change violates the defined PCI DSS segmentation boundaries.

You can use AWS CloudTrail to monitor all configuration changes for your AWS resources. Additionally, you can configure Amazon CloudWatch Events to trigger Lambda responders to take remediation action on your behalf. These are a sample of various AWS services that you can use to design and automate security controls for your CDE in AWS.

Feedback Loop

Your business is constantly changing and in turn the design and functionality fo your PCI in-scope application and associated AWS service choices can change over time. Apart from business requirement changes, you will also be evolving your AWS infrastructure to make it more secure, efficient, and easy to manage. This constant change makes it imperative to establish a feedback loop in the security and segmentation controls design process. Establish channels to gather feedback from the previous phases and from industry peers. Use this feedback to make the current process better and more secure. The feedback loop and the associated changes may require revaluation of current segmentation controls implemented to define the PCI DSS scope. This re-evaluation can also stem from a change in your organization CHD flow. Irrespective of the reasons, there must be at a minimum a yearly process of validating your established PCI DSS scope and re-categorizing systems in the scope, if necessary.

Conclusion

This paper addressed various architectural patterns that can be adopted to design proper segmentation boundaries to help restrict the PCI DSS scope to systems components necessary for secure functioning of the CDE resources hosted on the AWS platform. The services and/or features used to design segmentation boundaries include AWS accounts and security groups. Both are cloud native and thus resilient and elastic in nature. The infrastructure can be implemented as software code and then automated through your organizations existing CI/CD pipeline to ensure continuous compliance.

The segmentation philosophy defined by PCI DSS of isolation and restricted communication does not vary for resources on AWS platform; the variation is in the way those controls are achieved, which is unique to the AWS Cloud platform. In addition, as part of the shared responsibility model, using PCI DSS validated AWS services does not imply that use of those services in itself leads to achieving PCI DSS compliance for your environment. Use and architect those services in a PCI DSS complaint manner. Your organization is always responsible for the scope design and determination, but the design becomes more agile on AWS.

Contributors

Contributors to this document include:

- Avik Mukherjee, Security Architect, AWS
- Balaji Palanisamy, Senior Consultant, AWS
- Tim Winston, Sr. Assurance Consultant, AWS Security Assurance Services LLC

Further Reading

For additional information, see:

- PCI Compliance in AWS Technical Workbook
- AWS Whitepapers page
- SP 800-37 Rev. 2. Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy
- PCI Security Standards Council Penetration Testing Guidance
- Architecting for the Cloud: AWS Best Practices Whitepaper
- Introducing Cloud Native Networking for Amazon ECS Containers
- AWS Multiple Account Security Strategy
- PCI SSC Cloud Computing Guidelines
- PCI Security Standards Council Penetration Testing Guidance
- PCI DSS Virtualization Guidelines

Document Revisions

Date	Description
May 2019	First publication.

Notes

[1] See *Requirement 11: Regularly test security systems and processes* in the *PCI DSS Quick Reference Guide: Understanding the Payment Card Industry Data Security Standard version 3.2.1* found at https://www.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf

[2] Cardholder data (CHD): At a minimum, cardholder data consists of the full PAN. Cardholder data may also appear in the form of the full PAN plus any of the following: cardholder name, expiration date and/or service code See Sensitive Authentication Data for additional data elements that may be transmitted or processed (but not stored) as part of a payment transaction.

[3] For details around the various types of AWS services and their associated shared responsibility model, see the AWS Security Best Practices whitepaper.

[4] For more detailed analysis of the benefits of the cloud, see the Architecting for the Cloud: AWS Best Practices and Overview of Amazon Web Services whitepapers.

[5] PCI DSS v3.2.1 requirement 1.2 - 1.2 Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment.

[6] PCI DSS v3.2.1 requirement 1.3.5- 1.3.5 Permit only “established” connections into the network.

[7] PCI DSS requirement 11.3.4 - If segmentation is used to isolate the CDE from other networks, perform penetration tests at least annually and after any changes to segmentation controls/methods to verify that the segmentation methods are operational and effective, and isolate all out-of-scope systems from systems in the CDE.

[8] PCI DSS Requirement 10 – Track and monitor all access to network resources and cardholder data

Logging mechanisms and the ability to track user activities are critical in preventing, detecting, or minimizing the impact of a data compromise. The presence of logs in all environments allows thorough tracking, alerting, and analysis when something does go wrong. Determining the cause of a compromise is difficult without system activity logs.