Architecting for PCI DSS Scoping and Segmentation on AWS

Identify and Minimize Your PCI DSS Scope Using Appropriate Segmentation Controls

First published April 2019

Last updated November 2024

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 current Amazon Web Services (AWS) 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. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Security Assurance Services, LLC (AWS SAS) is a fully owned subsidiary of Amazon Web Services (AWS). AWS SAS is an independent PCI QSA company (QSAC) that provides AWS customers and partners with specific and prescriptive information on PCI DSS compliance. As a PCI QSAC, AWS SAS can interact with the PCI Security Standards Council (SSC) or other PCI QSAC under the confidentiality and contractual framework of PCI.

Contents

Abstract

This whitepaper empowers engineers, solution builders, Qualified Security Assessors (QSAs) and Internal Security Assessors (ISAs) to navigate Payment Card Industry Data Security Standard (PCI DSS) compliance within the AWS Cloud. It details best practices for scoping PCI workloads and implementing robust segmentation strategies using native AWS services. Find guidance on a layered security approach to achieve optimal PCI isolation of in-scope and out-of-scope resources, network segmentation, and consistent PCI policy enforcement across your accounts. This comprehensive guide provides you with knowledge that can help you safeguard customer data, mitigate fraud risk, and align with PCI DSS requirements in the ever-evolving digital landscape.

Introduction

Payment Card Industry Data Security Standard (PCI DSS) compliance and network segmentation is a critical but complex undertaking in the ever-changing digital commerce landscape. Although essential for safeguarding sensitive cardholder data, achieving PCI DSS compliance can be challenging, especially in dynamic cloud environments like Amazon Web Services (AWS). This whitepaper describes how AWS software-defined networking and robust security services can significantly reduce the number of systems and services within your cardholder data environment (CDE). This contributes to minimizing your compliance cost and effort. The paper guides you through the implementation of a multi-account structure that uses AWS Organizations, aligned with organizational units (OUs) for optimal segmentation. The reference architecture provides a foundational blueprint that you can tailor to your specific organizational needs. For details and guidance on how to apply security best practices and recommendations for the design, delivery, and maintenance of secure AWS workloads, see the AWS Well-Architected Security Pillar.

The guidance in this whitepaper can help you to align with PCI DSS requirements while benefiting from the agility and scalability of the cloud. Additional information on planning for and documenting the compliance of your AWS workloads can be found in the supporting PCI DSS v4.0 on AWS Compliance Guide.

PCI DSS scoping process on AWS

It is critical to understand the complete flow of account data within applications and the rest of your environment, including interactions with procedures and application code. The evaluation of data flow in the environment, as well as all *connected to* and supporting system components,

determines the applicability of the PCI DSS requirements and defines the boundaries and components of a CDE and the scope of a PCI DSS assessment.

Understand scoping

The requirements of the PCI DSS are mandatory for organizations that store, process, or transmit account data. Account data consists of cardholder data (CHD) and sensitive authentication data (SAD). This data security standard maintained by the PCI Security Standards Council includes a prescriptive set of IT security requirements—known as *controls*—to protect sensitive credit card information. CHD comprises the primary account number (PAN) and can also include the cardholder's name, card expiration date, and service code, if included with the PAN. SAD includes full magnetic-stripe or track data, card verification codes, and PINs or PIN blocks. The PAN is the defining factor for cardholder data and is the basis for determining the associated CDE.

For a service provider, complying with the PCI DSS might also be a customer requirement. Organizations are responsible for correctly identifying and defining their CDE, which is referred to as the *scope* of their PCI DSS assessment. The CDE comprises people, processes, and technologies that interact with or impact the security of account data and thus are in-scope for the PCI DSS. In this paper, we focus on the technology aspect of the CDE scope and not on the people or processes. Additional information on the applicability of the PCI DSS and account data can be found in PCI DSS v4.0.

The PCI DSS uses a term called *system components* to identify resources that are considered in-scope and part of the CDE. *System components* is a general term for technology resources that store, process, or transmit account data, have unrestricted connectivity to the aforementioned PCI resources, or could otherwise impact the security of the environment. On AWS, this includes AWS resources and services that handle AWS account data or support the account and resources that are PCI in-scope. This can include services that are actively involved in storing or processing cardholder data, such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), or AWS Lambda, as well as supporting services, such as AWS Config, AWS Identity and Access Management (IAM), and AWS Organizations.

Segmentation

Segmentation is colloquially referred to as *PCI DSS Requirement 0,* and it is not a formal PCI DSS requirement. Segmentation is used to limit the PCI DSS scope to the systems involved in the payment flow before addressing other PCI DSS requirements. 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 and different validation requirements for their out-of-scope infrastructure.

The PCI SSC 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 account data (CHD and SAD). This can also include system components on the same network segment that have unrestricted network access. These components are in-scope of PCI DSS and can bring other resources in-scope.
- ***Connected to* and *security impacting* systems**: These system components have a direct or indirect restricted connection to CDE systems, or provide some sort of management or security services to CDE systems. These components might help fulfill one or more PCI DSS requirements or help establish segmentation boundaries. These are in-scope systems, but they do not extend PCI DSS scope to other resources.
- **Out-of-scope systems**: These system components do not meet the preceding criteria and do not impact the security or configuration of CDE systems. For a system to be considered out-of-scope, controls must be in place to provide assurance that the out-of-scope system cannot be used to compromise an in-scope system component. These systems are not considered in-scope.

Figure 1 shows the considerations for scoping system components as provided by the PCI SSC. The diagram shows how system components can be categorized into the three groups of systems: CDE, *connected to*, and *security impacting*.

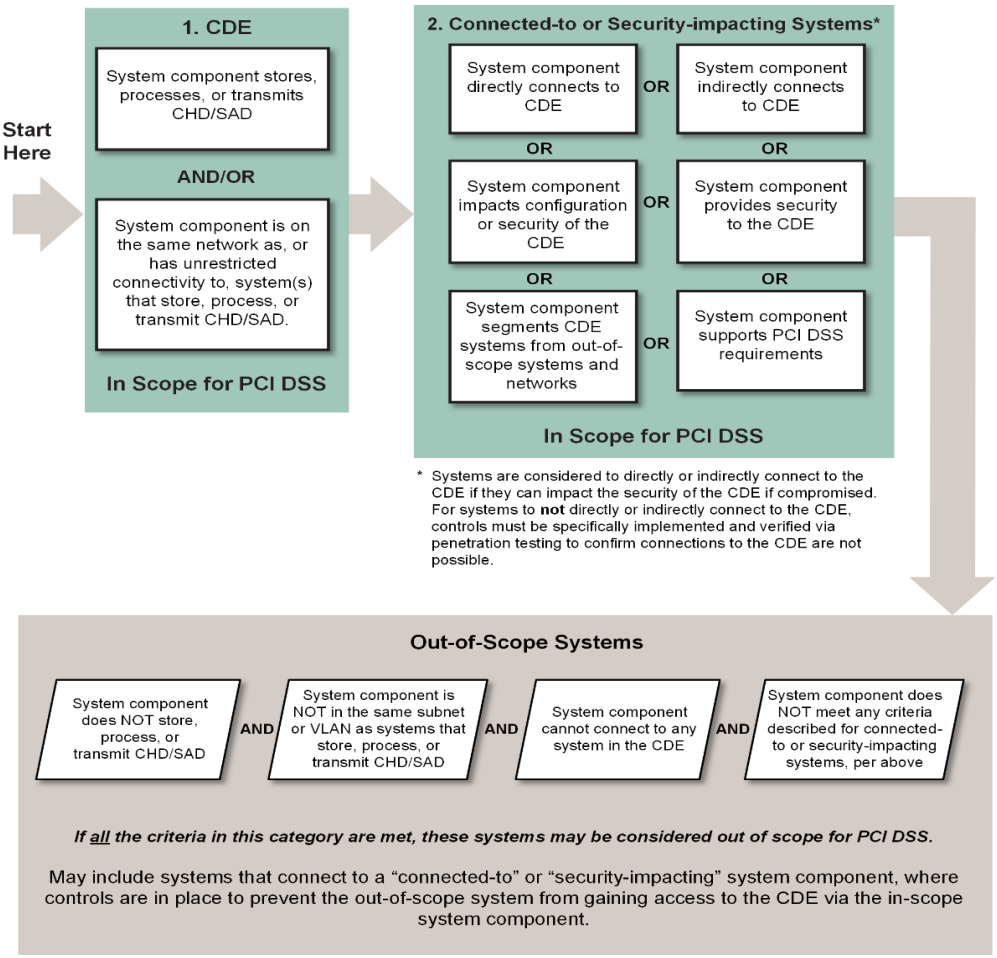

Figure 1: Overview of PCI DSS scoping (Source: PCI SSC Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation, page 11)

Strategies for minimizing PCI DSS scope cover these three categories. This whitepaper also addresses scoping and segmentation considerations for a hybrid environment, where the in-scope workloads might be spread across your on-premises data center and the AWS Cloud.

Similar to an on-premises scoping process, scoping an AWS PCI DSS environment begins with the account data flow. The significant difference with the AWS Cloud is that many network segments in the CHD and SAD flow could include AWS purpose-built services, such as Amazon API Gateway or AWS WAF. These services have well-defined connection configurations and are assessed as part of the AWS PCI DSS Level 1 Service Provider assessment. The AWS endpoints for these services are RESTful web service interfaces that are protected by firewall functionality as part of the AWS PCI DSS scope and serve as segmentation boundaries for services not receiving account data. You need to carefully review the segmentation capabilities

provided by AWS services and networking features against your documented data flows to determine an accurate assessment scope. You will learn about these segmentation capabilities in this whitepaper.

Infrastructure services

For infrastructure services where you control both the data and network layer, you need to take additional scoping steps. First, you need to identify instances that store, process, or transmit account data. Then you must identify the *connected to* and *security impacting* AWS resources. For example, an Amazon EC2 instance running a web service that handles account data would be in-scope. However, this EC2 instance can bring many other *connected to* resources in-scope because you are responsible for defining and managing the network connections. An AWS security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. If security groups attached to EC2 instances are not adequately configured and restricted, other instances that can establish a network connection to and from these instances in the same subnet or virtual private cloud (VPC) are potentially *connected to* systems and are considered in-scope for PCI DSS. These *connected to* systems might include systems such as a monitoring server polling non-CHD data for reporting. EC2 instances and other AWS resources that are considered *connected to* could also be considered *security impacting*. *Security impacting* systems can include EC2 instances that host security tooling, such as antivirus protections for the EC2 instances, or other AWS services such as AWS Directory Service for Microsoft Active Directory that provide authentication and authorization. This same scoping guidance also applies to other AWS services where you manage the underlying EC2 instance, such as Amazon Elastic Kubernetes Service (Amazon EKS) and Amazon Elastic Container Service (Amazon ECS) with the EC2 launch type.

Managed services

Managed services require a similar consideration as infrastructure services regarding scoping and network connectivity. However, with AWS managed services, AWS manages more of the underlying resource infrastructure. This abstracts the responsibility of maintaining operating systems from you and reduces your PCI DSS scope at the resource level. However, the non-abstracted portion of the service is your responsibility and includes network security. For example, an organization setting up Amazon RDS needs to verify network segmentation by applying appropriate security groups to the elastic network interface in the VPC and by restricting user access to the service with IAM. Managed service instances are isolated by design to help ensure that data is not shared between different instances of the service unless otherwise explicitly allowed through a configuration change that you manage.

Abstracted services

By design, abstracted services are secured to verify that data is not shared between different instances of the service unless otherwise explicitly allowed. For abstracted services that store, process, or transmit account data, the component in-scope is the particular instantiation of the AWS service that handles account data. Also, in-scope are specific resources that both connect to or are connected to from the abstracted service instance. For example, an organization might have many Amazon DynamoDB tables provisioned but only use a subset of those tables to store or process account data. In this case, the tables that are used to store account data are part of the organization's PCI DSS scope. The same is true for Lambda instances that handle account data. The parties calling the function are in-scope, as well as the resources that the Lambda function connects to as part of the data flow. *Connected to* in-scope resources are both the systems and services that pass account data to the abstracted service resource in question, and also those that in some form connect to the CDE even if the data flow does not contain account data. An example is an Amazon Simple Storage Service (Amazon S3) bucket that receives operational logs from a payment application server hosted on an EC2 instance. The S3 bucket is considered *connected to* even if no account data exists in those logs. In this example, you can use Amazon Macie, a fully-managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data on AWS. To verify that your PCI DSS scope does not extend past the S3 log bucket, you can configure Macie to monitor objects for account data in those buckets and provide an alert if sensitive data is identified.

Some abstracted services, such as Elastic Load Balancing (ELB), do maintain a persistent network connection to the resources in the target groups, and you must evaluate these services as part of your scoping process.

Unlike traditional virtual servers in a subnet, which have default network connections available to each other, abstracted service instances are segmented by default. The connections between these services are on-demand data connections rather than persistent network connections. For scoping, the focus is placed on the type of data traversing the connection, and the configuration and permissions of the abstracted service to allow data in or out. For example, a Lambda function will only communicate based on the code that you provision. That same function can only be communicated to, or *connected to*, using explicitly allowed IAM permissions and configurations.

If an abstracted service does not handle account data, does not impact the security of the CDE, and does not communicate with a resource that handles account data, you can demonstrate that you have content-aware controls to sufficiently show that account data cannot travel beyond your CDE through these abstracted services. This will allow you to exclude it from your PCI DSS scope. Abstracted services that communicate to an account data storage location,

such as collecting system logs or metrics from a CHD-containing database, would still be considered *connected to*.

You must adopt proper architectural designs, including data filtering and monitoring, to help ensure that these services do not store, process, or transmit account data, and therefore can be safely considered out-of-scope for the PCI DSS.

Scoping guidance for hybrid environments

As your organization migrates workloads to the AWS Cloud, you might run a hybrid infrastructure. You might have workloads that are running both in your on-premises data centers and on AWS. Your segmentation controls must consider this hybrid infrastructure and communication between on-premises and AWS Cloud environments.

Consider the following scenarios:

1. **Scenario 1:** PCI DSS resources hosted on AWS have network connections to resources in an on-premises data center.
2. **Scenario 2:** PCI DSS resources hosted in an on-premises data center have network connections to resources on AWS.
3. **Scenario 3:** PCI DSS resources are hosted both in an on-premises data center and on AWS.

For scenarios 1 and 2, the scope of the non-CDE resources (regardless of where they are hosted) is determined by the type of connectivity that these non-CDE resources have with in-scope systems.

- If the non-CDE resources have direct connectivity to CDE resources, then those on-premises or AWS resources are in-scope for PCI DSS and become part of the CDE.
- If the non-CDE resources have a connection to systems that are considered *connected to*, then those on-premises or AWS resources can be considered out of scope for PCI DSS as long as they do not provide security or management services. Further, if those out-of-scope systems are compromised, there should be no way for those systems to be used to further compromise the CDE. For example, AWS Systems Manager can manage both AWS Cloud and on-premises servers and in-scope EC2 instances. In this scenario, the on-premises servers are potentially out of scope provided they do not interact with account data or have other connections to the CDE.

To limit scope, consider the following guidelines:

- Restrict network connections from on-premises networks to only non-CDE resources.

- If connectivity to CDE resources is required, implement proper security controls and designs to help prevent transitive network connections, which might bring unwanted network and system components in-scope.
- Implement multiple segmentation controls to help ensure that scope boundaries are not altered erroneously. You can implement these controls through a combination of access control rules defined and implemented by using on-premises stateful firewall technologies, security groups, and network access control lists (network ACLs), and backed up by IAM permissions to achieve defense in depth.

For scenario 3, you should group CDE resources into either on-premises or AWS Cloud environments for simplicity of scope determination. If this grouping is not workable, carefully verify that CDE, *connected to*, and *security impacting* systems are identified both on-premises and on AWS and that the connections to these systems are correctly identified.

Decision flow – PCI DSS scope identification

A decision flow can help you correctly identify the PCI DSS scope in your organization. This process starts with correctly identifying the flow of account data in your organization's environment.

When you have identified the account data flow, the next step is to identify in-scope resources in your environment.

Figure 2 shows the decision flow that you can use to identify the PCI DSS scope. This flow can also help you correctly define segmentation boundaries at different stages of the flow to segregate other AWS resources from in-scope resources and reduce the PCI DSS scope.

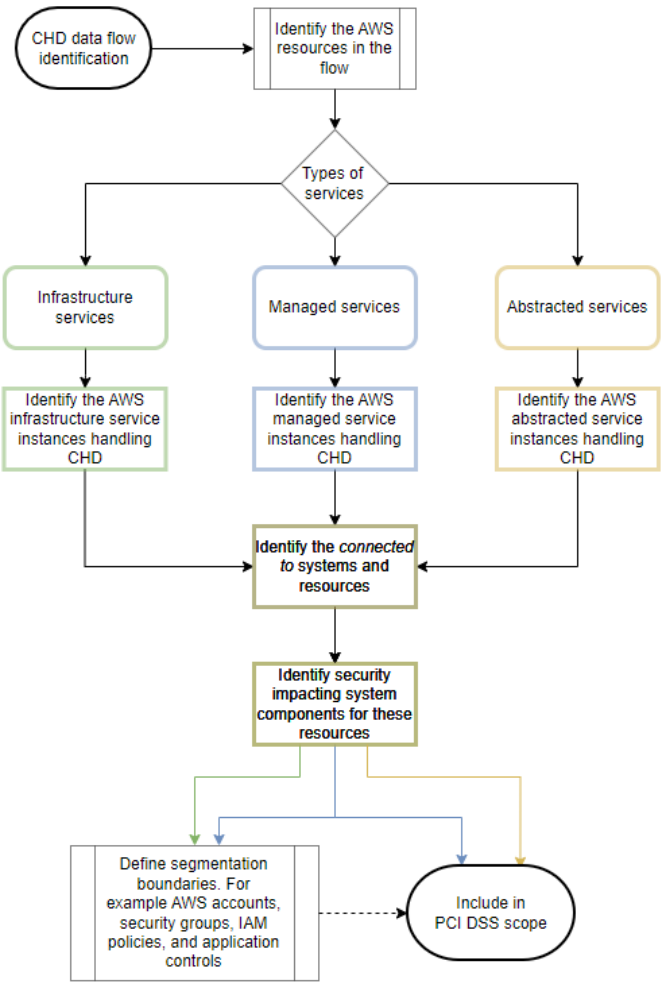

Figure 2: Decision flow to correctly identify the PCI DSS scope and associated segmentation boundaries

The sections that follow go into more detail about the decision flow process.

Step 1: Identify the CHD and SAD flow

Before you can define your PCI DSS CDE and design segmentation boundaries, you must have an accurate understanding of the CHD and SAD flow in your organization. To correctly identify the account data flow, you must identify and define the entire lifecycle of account data in your organization. This includes the entry of account data into your environment; the subsequent transmission, processing, and storage of the account data; and the eventual secure destruction, devaluation, or exit of the account data from your environment.

Step 2: Identify in-scope resources in your environment

Identify the various AWS resources that make up the account data flow. These resources are those that receive, process, store, or transmit account data. It is critical to define what is and is not in-scope as part of your analysis so that your company's ISA, external QSA, or both can clearly understand what services their assessment should be limited to when conducting their audit.

Step 3: Categorize the systems and services

This step categorizes systems and resources into infrastructure, managed, and abstracted services. The scope identification and segmentation of these resources are based on different connections. Infrastructure and managed services primarily communicate with each other over a network (OSI layer 3-4) connection, and communication to abstracted services is often through the service's API (OSI layer 7). After you identify the different AWS resources, you can identify the PCI DSS scope.

Step 4: Identify *connected to* resources

After you have defined the systems and services involved with the account data flow, identify resources that have connections to or from them. These connections might be for purposes like operational monitoring, logging, or configuration management. Both persistent and on-demand connections must be reviewed, such as from an infrastructure service like an EC2 instance or to an abstracted service like an S3 bucket. These *connected to* resources are part of the CDE.

Step 5: Identify security impacting resources

After you have defined *connected to* system components, you should identify resources and AWS services that provide security services or could otherwise impact the security or configuration of the CDE. This includes a wide range of potential system categories, such as system components that are used to satisfy a particular PCI DSS requirement like logging, or services like DNS that are used to support an environment. Many types of system components could fall into this category. For more information, see page 12 of the PCI SSC Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation.

Step 6: Design segmentation boundaries

After you have identified the in-scope AWS resources, you should design segmentation boundaries to help ensure that other AWS resources are properly segmented to exclude them from the PCI DSS scope. For AWS abstracted services, this segmentation is primarily controlled by the application and associated application code controlling the flow of the account data. For

infrastructure service resources like those of Amazon EC2, and managed services resources like those of Amazon RDS, you must also design network-level segmentation along with application-level segmentation.

AWS Cloud considerations for solutions

The AWS Cloud has benefits such as scalability, disposable resources, traceability, security control automation, and continuous validation and testing. AWS also offers various business advantages like the ability to shift from a fixed cost model to a variable cost model, and benefits such as the ability to pay only for the individual services that you need and only for as long as you use them, and no long-term business contracts. These unique characteristics make cloud adoption advantageous for most organizations.

For more information on AWS economic benefits, see the Cloud Economics Center. You can realize these benefits when you use these cloud-specific characteristics to design your payment application infrastructure. To learn about additional benefits of AWS, see the AWS Well-Architected Framework and Overview of Amazon Web Services.

Shared Responsibility Model

Security and compliance are shared responsibilities between AWS and their customers. This shared model can help relieve your operational burden; AWS operates, manages, and controls the components from the hypervisor 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 your data *in* the cloud. Your responsibility includes meeting your 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 level of abstraction provided by a particular AWS service. As service abstraction increases, your responsibility decreases. Evaluate every service used in your environment and understand the impact of using the service on your overall PCI DSS scope. This approach can help you understand what you must do to meet your compliance obligations.

Virtualization of traditional network controls

The software-defined networks (SDN) that AWS provides mimic traditional networking constructs, but require a new way of viewing and managing your networks. For example, traditional on-premises network controls like virtual local area networks (VLANs) are implemented differently on AWS because layer 2 networking is transparent to you.

On AWS, Amazon Virtual Private Cloud (Amazon VPC) represents a logically isolated section of the AWS Cloud where you can launch resources in a virtual network. It is common for resources that provide similar functionality or similar scope to span across multiple subnets across different Availability Zones (AZs) to provide redundancy. Therefore, Amazon VPCs and subnets should be used as grouping constructs and not 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 can be short-lived, potentially measured in minutes or hours rather than months or years like their physical on-premises counterparts. Other AWS managed services or features, such as Lambda and ELB, scale vertically to accommodate the resource requirement. The segmentation controls that 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 could lead to an incorrect scope definition. It is important to note that when resources scale horizontally, such as EC2 instances that are part of an AWS Auto Scaling group, the population of PCI DSS in-scope resources scales with it, either up or down.

Abstracted services and API-based infrastructure

Many AWS offerings are provided as managed or abstracted services. This means that AWS manages much of the underlying infrastructure for you, and you do not have control over it. Many abstracted AWS services are only communicated with through the API endpoint of the service over HTTPS. AWS service APIs include inherent network segmentation controls, in addition to other controls such as authentication, authorization, and data integrity. This verifies that only data from authorized entities is exchanged between the calling system and the service. When configured to do so, these services communicate among themselves and other AWS services over access-controlled APIs. This configuration is provided as part of the service and aligns with the layer 3-4 network security control provided by firewalls. You should use AWS abstracted services to reduce the *connected to* scope of your environment. For example, select a log consolidation service like Amazon CloudWatch that makes encrypted API calls to forward log data, rather than an agent that maintains an open TCP/IP connection.

When you configure abstracted services like Lambda or ELB for account data flows, you must also configure secure connectivity such as HTTPS using TLS version 1.2 or later. If an AWS abstracted service instance stores, processes, or transmits account data, the resources that it is configured to connect to are also PCI DSS in-scope. You must design application layer-based

segmentation and traffic filtering controls as part of your responsibility under the shared responsibility model.

Automation

With automation, you can implement most infrastructure and application changes without manual intervention. This provides agility, decentralizes the change management process, and expedites the deployment process. You must also automate the segmentation controls to the greatest extent possible so that the segmentation controls are applied in tandem with changes to the infrastructure and applications. This approach preserves the defined scope boundary. Automation can also help detect when segmentation controls are changed so that you can implement proper remediation steps, such as re-establishing the controls or alerting someone in near real time to analyze the cause and effect of changes.

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 achieve defense in depth. You can create these boundaries at various layers on AWS and then combine them with each other to reduce the PCI DSS in-scope systems to the minimum required for your secure and functional CDE.

AWS account layer

An individual AWS account provides the highest level of segmentation that you can achieve on AWS. By design, the resources provisioned within an account are logically isolated from the resources provisioned in other accounts, even within your own organization in AWS Organizations. When you design your AWS environment, it's a best practice to use isolated accounts for PCI DSS workloads. You can reduce your PCI DSS scope by segmenting in-scope and out-of-scope resources into their own accounts.

An AWS account acts as an identity and access management isolation boundary. This helps prevent accidental scope creep because logical account-level isolation can be changed only by establishing explicit communication channels between resources in separate accounts. This approach also helps reduce the impact of security incidents and segmentation misconfigurations because, by default, it is designed not to allow changes to architecture and controls in out-of-scope accounts to adversely affect the security of in-scope resources in other accounts. There are some additional operational benefits to using multiple AWS accounts. This includes distributing the AWS service quotas (limits) and request rate limits (throttling) because these are allocated for each account. For more details, see Benefits of using multiple AWS accounts.

Figure 3 shows a proposed multi-account architecture in relation to PCI DSS scope and segmentation. The sections that follow provide more detail about this architecture.

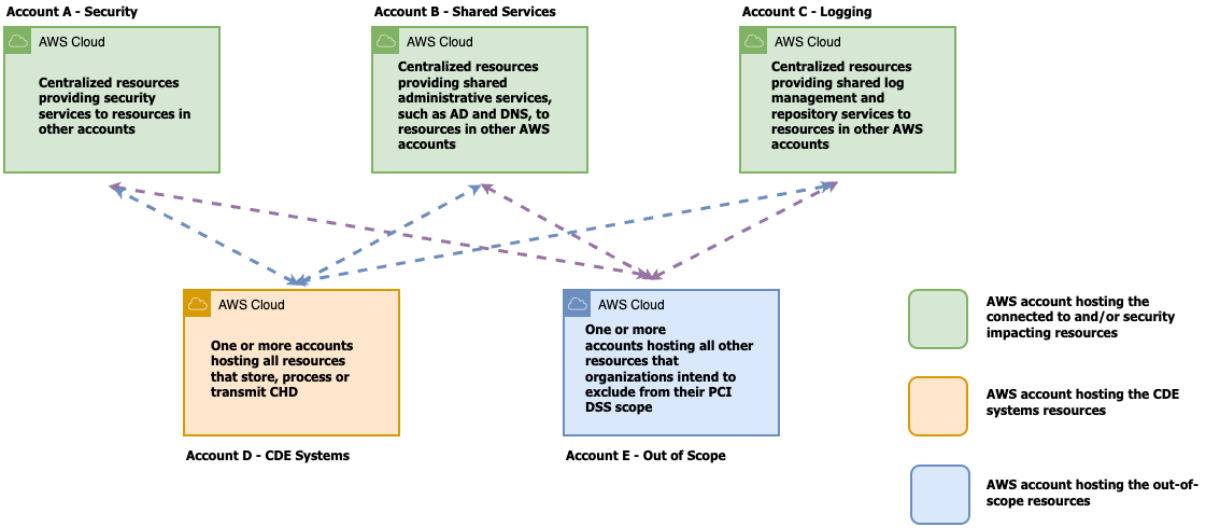

Figure 3: Multi-account architecture to restrict PCI DSS scope

Components of a multi-account architecture

The AWS Organizations service allows you to manage multiple AWS accounts in a centralized and organized manner. It provides security boundaries through organizational units (OUs) and accounts. OUs and accounts are logically isolated, with no access allowed between them by default. This segmentation also helps with PCI compliance because it sets the first level of segmentation for PCI workloads. AWS Organizations also allows for centralized governance and management of accounts, including centralized identity and access management, setting guardrails using service control policies (SCPs), and configuring AWS services and resources.

For organizations that need to comply with the PCI DSS, AWS provides a recommended framework for setting up Organizations for a multi-account environment. The use cases where the framework will be most effective are medium to larger organizations with separate accounts for development, testing, and production environments as shown in Figure 4.

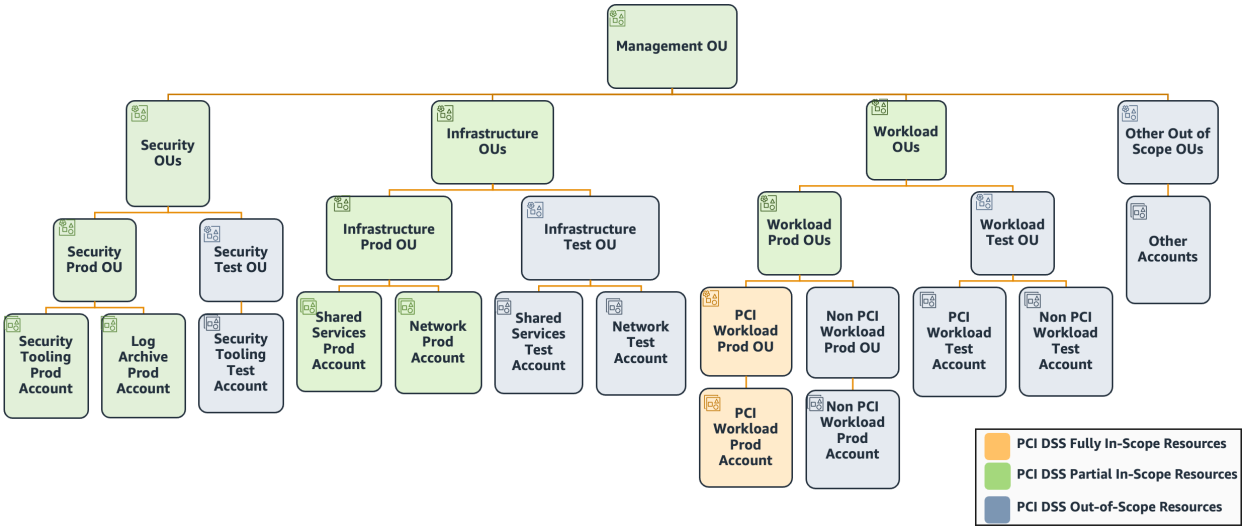

Figure 4: Block diagram for AWS Organizations OUs and account structure for PCI DSS

OUs in AWS are used to create logical isolation for resources and workloads within accounts. This structure allows for more granular application of policies, controls, and access permissions to help you meet security and compliance requirements effectively. The hierarchy of OUs helps in organizing and managing resources in the AWS environment. Following AWS best practices, a multi-account architecture groups accounts with similar functionalities within OUs under a root. This grouping facilitates resource sharing, restricts PCI DSS scope, supports separation of duties, and enforces least privilege. OUs simplify the application of common policies such as AWS Organizations SCPs, which restrict permissions granted by IAM policies to entities like IAM users and roles. For example, SCPs can prohibit the provisioning of AWS services that aren't PCI DSS-compliant within an OU that contains PCI DSS in-scope accounts.

Recommended OUs and accounts

To implement AWS Organizations, we recommend that you assess how your business, governance, security, audit team, infrastructure, and operational requirements related to PCI segmentation can be met in AWS. The recommended reference architecture of the AWS Organizations structure is shown in Figure 5. This architecture incorporates the isolation requirements of workloads (such as PCI, non-PCI, production, and non-production) and places them in appropriate accounts and OUs for isolation. It also considers the role and accessibility boundaries of your organization team before assigning AWS services to each account under specific OUs. Additionally, AWS services are placed in various accounts based on security and operational requirements, with a focus on limiting access to specific roles or groups and minimizing cost and operational overhead. You can use this strategic guidance on the AWS Organizations structure to align with PCI compliance and segmentation requirements.

Figure 5 provides the reference architecture details for recommended OUs and accounts, based on the AWS multi-account strategy guidance, while adjusting the grouping names to help align with PCI DSS scoping. You should name your OUs according to your organization's naming conventions.

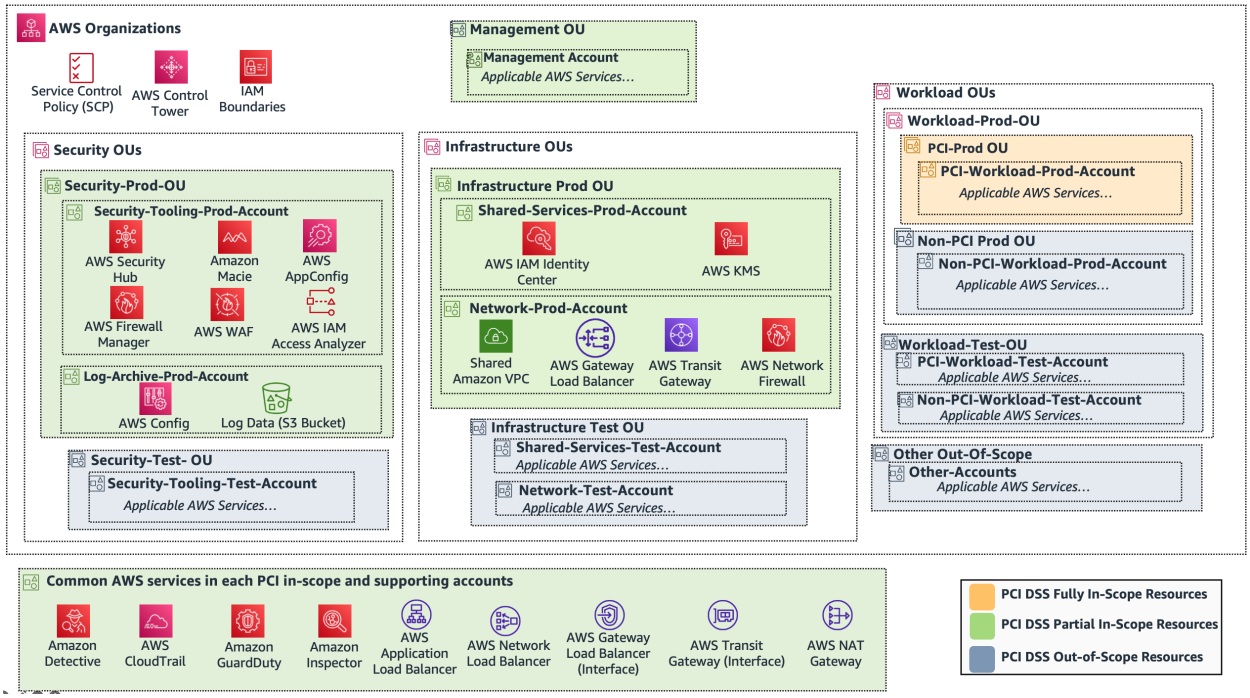

Figure 5: AWS account layer segmentation recommended architecture for PCI DSS scope

Although the recommended OUs are aligned toward common use cases, you can define your own OU structure that best meets your needs. The guidance here should align with the needs of most customers; however, it is not one-size-fits-all. Depending on your requirements, you might tailor your organization structure differently.

The recommended OU categories with respect to PCI DSS scoping include the following:

- **PCI** – This is the OU category that groups accounts that host the CDE, workloads, and resources that directly store, process, or transmit account data, along with resources that are categorized as *connected to* systems. This OU is usually a child OU under the top-level workloads OU.
- **Security-impacting** – This is the OU category that groups accounts that provide common security and shared service capabilities to help secure and support your overall AWS environment, including your CDE.

- **Out-of-scope** – This is the category for other OUs that group accounts that host resources and workloads you deemed to be out-of-scope for CDE.

AWS SCP controls

AWS Organizations enables centralized governance rules for security, accessibility, operations, and compliance that you can define by using AWS SCPs combined with AWS Identity and Access Management (IAM) access boundaries and AWS Control Tower. SCPs are a means of implementing guardrails in your organization. You can apply an SCP to the OU that restricts the access and actions allowed on resources within an account or within all of the accounts under a specific OU. You can also use IAM to restrict access to the resources in that OU or account. For example, you can create an SCP that denies access to the resources in the PCI-Prod-OU, and then create exceptions to the SCP to allow specific users or roles to access the PCI resources. This approach simplifies the process of achieving least privilege access. Also, by attaching policies to OUs rather than to individual accounts, you can simplify management of policies across groups of similar accounts. As the number of accounts in a specific organization scales up or down, the policies attached to OUs will automatically apply to the accounts within those OUs.

Following is an example of an SCP that you can use to restrict access to PCI-compliant resources:

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "ForAllValues:aws:ResourceOrgPath": {
      "aws:ResourceOrgPath": "[PCI-Prod-OU]"}
    }
  }
 ]
}
```

Following is an example of an IAM policy that you can use as an exception role to restrict access to PCI-compliant resources:

```
{
"Version":"2012-10-17",
"Statement":[
  {
     "Effect":"Deny",
     "Action":"*",
     "Resource":"*",
     "NotPrincipal":{
        "AWS":"arn:aws:iam::[PCI-Workload-Prod-Account-
id]:role/[pci-compliant-resources-role]"
       }
  }
  ]
}
```

See the blog post How to control access to AWS resources based on AWS account, OU, or organization for details on how you can use various conditions with SCPs to restrict traffic.

AWS Control Tower, landing zone controls

AWS Control Tower helps you set up and manage a secure, multi-account AWS environment known as a *landing zone* by using AWS Organizations. It simplifies the process of provisioning new accounts that adhere to organizational policies. AWS Control Tower applies preventive and detective controls called guardrails to help prevent a drift from best practices. It is considered in-scope for PCI DSS compliance because it orchestrates the multi-account environment where cardholder data resides. AWS Control Tower offers prepacked governance rules for PCI compliance and segmentation requirements, allowing users to create a landing zone template with specific restrictions. The AWS Account Factory feature within AWS Control Tower can create pre-provisioned accounts that adhere to PCI compliance requirements, including AWS services, IAM roles and policies, SCPs, and other preconfigurations.

Important: AWS Control Tower names the account that is under the security OU the *Audit Account* by default. You can rename this account during the AWS Control Tower setup.

For more information, see Best practices for AWS Control Tower administrators.

At the organization level in AWS Organizations, segmentation boundaries play a crucial role in dividing the structure into distinct network segments. This segregation serves to isolate PCI in-scope workloads, effectively reducing unauthorized accesses at the account layer. The combination of SCPs and IAM policies works in tandem to define granular access controls, specifying who can access specific resources.

Management account

AWS Organizations is an account management service that you can use to merge multiple AWS accounts into an *organization* that you create and centrally manage. By using the AWS Organizations account management and merged billing capabilities, you can better meet the budgetary, security, and compliance needs of your business.

You should consider the management account in-scope for PCI DSS because it impacts the security of your organization and your CDE. We recommend that you use the management account only for tasks that can only be performed by that account. Store your AWS resources in other accounts in the organization and keep them out of the management account.

One important reason to keep your resources in other accounts is because AWS Organizations SCPs do not work to restrict users or roles in the management account. For more information, see Best practices for the management account.

Workload OU

Your PCI application accounts are the most sensitive part of your PCI DSS infrastructure. The Workload OU is where you host business-specific workloads, including production and non-production environments. It contains multiple child OUs grouped by business unit, team, or software development lifecycle (SDLC) environments. At least one child OU, like PCI-Prod OU, should be dedicated to hosting PCI production in-scope workloads. This OU is the focal point for your PCI DSS scope, and organization policies are enforced by using SCPs.

You can segregate your CDE and *connected to* resources into separate PCI application accounts or combine them in the same account. You should base this design on your workload requirements. If you are mixing these types of resources, you can use additional network- and application-level controls to segregate these resources from each other. For this whitepaper, we are using only one account. You should only use PCI application accounts to provision resources that are in-scope for PCI DSS, unless required by your workload design.

The general resource types hosted in these accounts are the following:

- Compute and network resources that receive or transmit account data, from and to external entities
- Compute resources that process account data
- Resources that store account data at rest
- Resources that establish a network- or application-level connection to the preceding resources

System components in the PCI workload accounts can bring other resources in scope. This means that resources that these accounts connect to have the potential to be considered in-scope, regardless of the functionality or the data involved in the communication. Communication to and from these accounts should be highly restricted (in terms of both logical and network access) to only the necessary resources in other accounts, and based on system management or business requirement purposes. AWS Config can track changes to a particular AWS resource, which can cause an Amazon CloudWatch event to generate an alert, triggering a Lambda function that remediates security control deviations to orchestrate a variety of steps to restore or reconfigure security controls.

You can group your AWS accounts that host non-PCI workloads into one or more separate child OUs without bringing them in-scope for PCI. The resources within the scope of PCI DSS compliance for accounts associated with the workload OU meet both of the following criteria:

- Impact the security of the CDE systems accounts
- Have connections (logical access or network) with resources in CDE systems accounts

As noted previously, AWS accounts, by design, provide logical isolation, and therefore, these accounts isolate your resources related to non-PCI workloads from resources related to PCI workloads (unless you explicitly introduce a cross-account network or application connection). In this whitepaper, we refer to these OUs as *non-PCI OUs*; and the accounts, as *non-PCI application accounts*.

You can have additional child OUs to host test accounts that your application team owns and uses to develop, test, and stage workloads. The significant change in the PCI DSS Requirements 6.5.1 and 6.5.2 is a requirement that all changes to system components in your CDE environments be tested to verify that they do not adversely impact system security. However, you can safely exclude these non-PCI production resources from your PCI DSS scope if you make sure that resources within the accounts do not have connections to CDE resources or impact the security of CDE resources.

Figure 6 shows an example structure of a workload OU with associated accounts as described in this section. The Workload OUs host Workload-Prod-OU for PCI in-scope and out-of-scope workloads, and Workload-Test-OU for PCI test accounts for PCI (accounts that don't host any production data or CHD or SAD data) and non-PCI test workloads. PCI-Prod OU hosts in-scope PCI workloads, whereas Non-PCI Prod OU hosts out-of-scope PCI workloads.

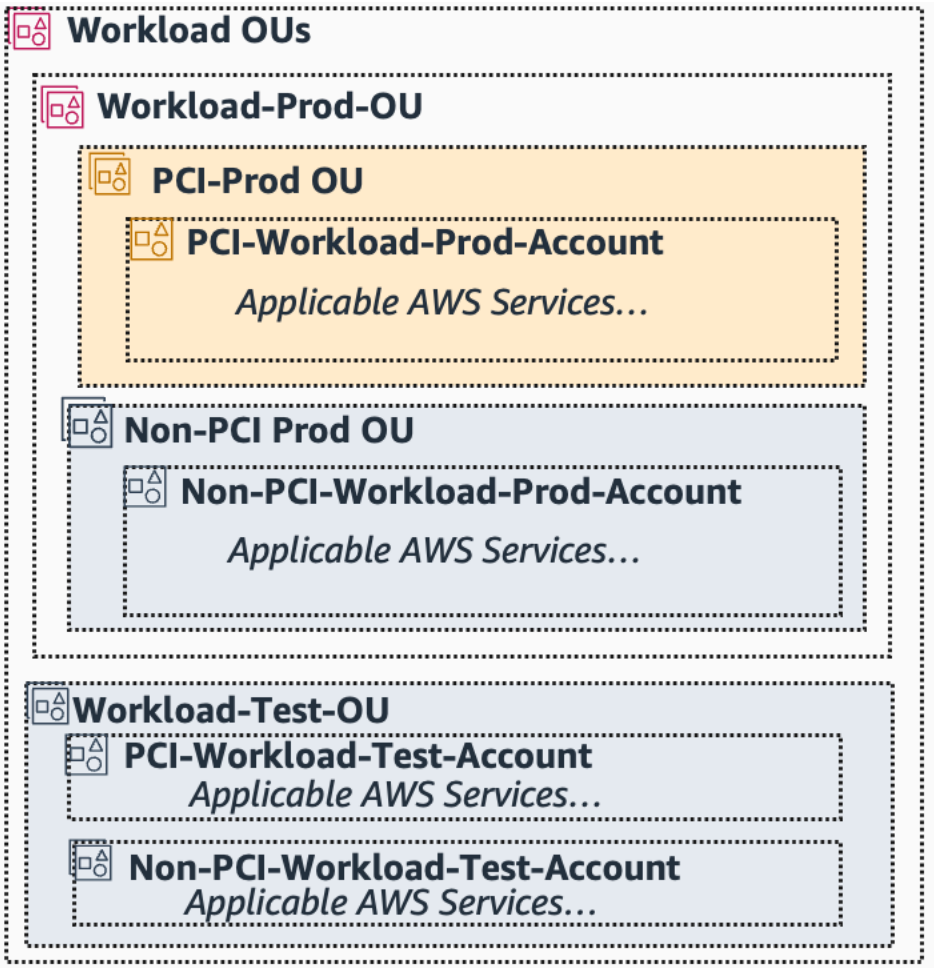

Figure 6: Recommended AWS Organizations member account structure for the workload OU

Foundational OU

This is a structured grouping of OUs and sub-OUs to manage accounts that provide security and management services to CDE resources. Centralized cloud environment or cloud engineering teams, made up of cross-functional representatives from your security, infrastructure, and cloud engineering teams, oversee these accounts, workloads, and data within the foundational OUs. The AWS multi-account recommendation outlines two distinct OUs in the foundational OU: security and infrastructure.

Security OU

Group the accounts that provide security services to the CDE accounts together in the security OU. Your security organization, if it doesn't share operations with other teams, should own and

manage this OU, along with child OUs and associated accounts. Use the security OU to host resources that provide security services to CDE systems. We recommend that you have sub-OUs within the security OU to host non-production accounts for testing.

The following are the recommended accounts in the security OU:

- **Log Archive** – This account acts as the audit trail consolidation point for log data collected from the accounts in the organization. This includes logs from the CDE system accounts. You should consider the log archive account to be in-scope because this account helps fulfill several controls outlined under the PCI DSS Requirement 10.
- **Security Tooling** – This account is the administrator account for security services in AWS accounts, managed through AWS Organizations delegated administrator functionality. Services in the AWS Security Response Automation (SRA) that support delegated administrators include AWS Config, AWS Firewall Manager, Amazon GuardDuty, IAM Access Analyzer, Amazon Macie, AWS Security Hub, Amazon Detective, AWS Audit Manager, Amazon Inspector, AWS CloudTrail, and AWS Systems Manager. The security team is responsible for managing these services and monitoring security events or findings. This account groups security-oriented workloads like anti-virus, vulnerability scanning, intrusion detection system (IDS), and intrusion prevention system (IPS) workloads, depending on the specific requirements and architecture of your organization. It is important to consider this account in scope for fulfilling portions of the PCI DSS Requirements 2, 3, 5, 6, 7, 8, and 11.
- The AWS Firewall Manager and AWS WAF services are often managed by the networking team and are typically located in a Networking account within the Infrastructure OU.
- **Security Test OU** – The test sub-OU and associated accounts are considered out of PCI DSS scope.

Figure 7 shows an example structure for the security OU. The Security OUs host Security-Prod-OU and Security-Test-OU for *connected to* in-scope accounts and the test account. The Security-Prod-OU hosts Security-Tooling-Prod-Account and Log-Archive-Prod-Account with security tooling and log controls using AWS services.

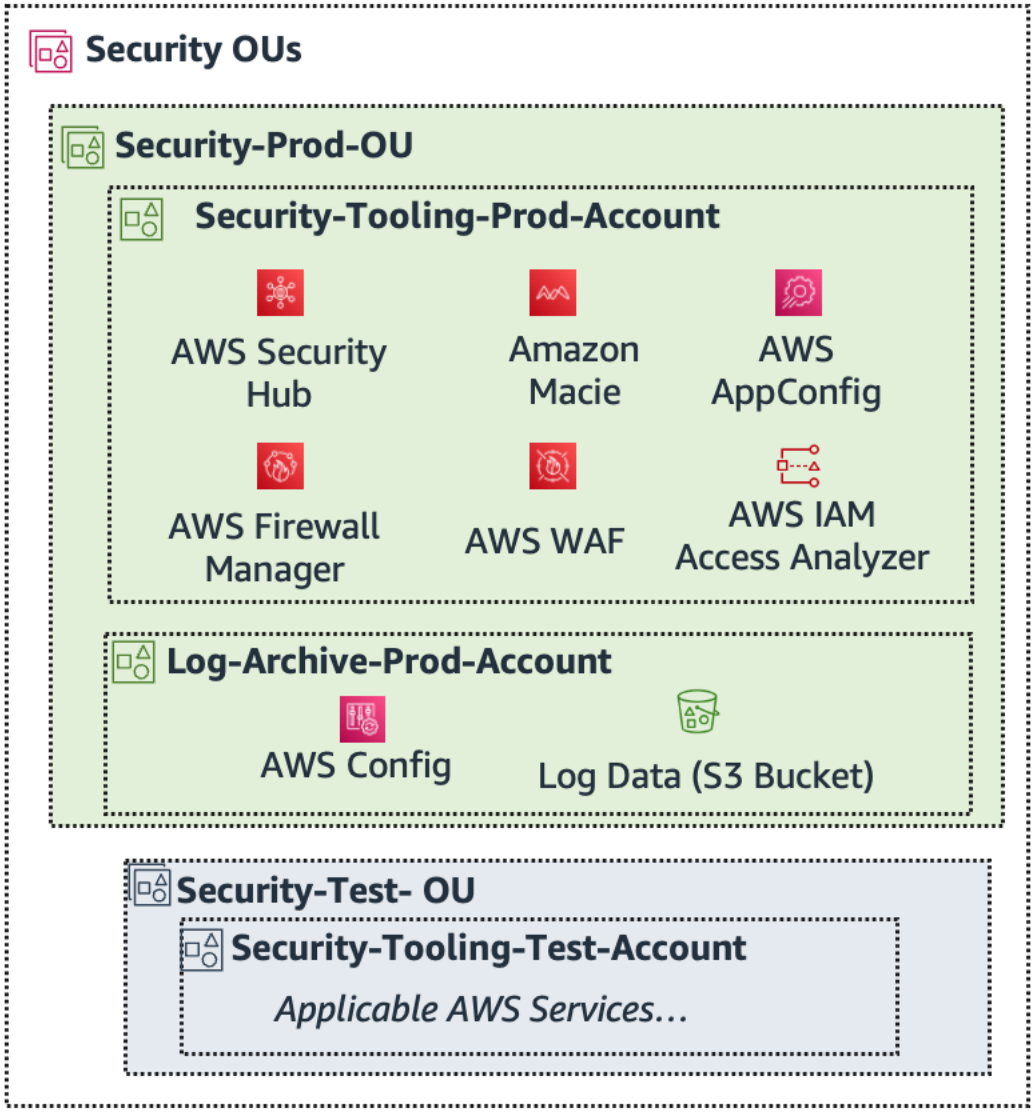

Figure 7: Recommended AWS Organizations member account structure for the security OUs

You can set up your security OUs in the multi-account environment by using the AWS Control Tower to create two OUs: Security-Prod-OU and Security-Test (non-production)-OU. Within the production OU, we recommend that you create two accounts: log archive and security tooling.

The log archive account is a consolidation account of your security audit log data aggregated from your PCI accounts. The primary use of the log account is to serve as a centralized storage location for copies of every account's audit, configuration compliance, application/OS logs, and operational logs. For example, AWS API access logs recorded in AWS CloudTrail, logs for changes to AWS resources recorded in AWS Config, and other logs that have security implications. This log data should be locally available to use for authorized IAM users and roles.

The access to the log account and data should be restricted to read-only and for specific roles or users who consume data for auditing, security analysis, troubleshooting, and similar tasks. This helps limit who needs to access your PCI workload account. The log data housed in the log archive account will be immutable and protected from being changed or deleted.

The security tooling account contains broadly applicable security-oriented workloads in the form of security services and tools. The account provides centralized delegated admin access to AWS security services. The account aggregates security data from other services across the AWS organization accounts, including AWS Security Hub, Amazon GuardDuty, Amazon Macie, AWS AppConfig, AWS Firewall Manager, Amazon Detective, Amazon Inspector and IAM Access Analyzer. The security policy of the security tooling OU restricts access to authorized IAM users and roles to access security data and administrate security services. The SCP and IAM policies, in many cases, need to address common operational tasks and roles, such as AWS Firewall Manager and AWS WAF service capabilities that are assigned to a different team. These services may reside in different accounts, such as the infrastructure OU.

Infrastructure OU

The infrastructure OU serves as another foundational OU, intended to host the services that support a shared infrastructure. The team members responsible for the shared infrastructure and networking segmentation services require access to the infrastructure OU and its child OUs to manage associated accounts. In your CDE environment, use this OU to host resources that provide management functions to CDE systems. Common use cases for this OU include shared services such as hybrid DNS infrastructure and directory services. Similar to the security OU, the recommended best practice is to separate the production and test resources through production and test child OUs.

The following is the recommended account for the infrastructure OU:

- **Shared Services** – This account supports the services that multiple applications and teams use to deliver their outcomes. For example, directory services like AWS Directory Service for Microsoft Active Directory, messaging services, and metadata services. To maintain separation of duties, the shared services account is kept separate from the network account. This account is important for adhering to PCI DSS Requirements 2, 3, 5, 6, 7, 8, and 11.

Important: The AWS multi-account guidance whitepaper recommends a centralized networking account under the infrastructure OU. This is recommended to be the central hub for your network on AWS, hosting network resources that route traffic between accounts and egress or ingress traffic to the internet. To reduce PCI DSS scope, do not centralize network traffic to and from your CDE systems, but rather host it locally as part of your CDE accounts. If you centralize network resources that handle account data traffic, you could extend the PCI DSS scope into and through the centralized network account.

- **Network Services** – The Network Services account acts as a gateway between your application and the internet, and it requires protection. It isolates networking services from individual application workloads and infrastructure, limiting connectivity and data flow. By separating inbound and outbound VPCs, you can help to safeguard sensitive infrastructure. The inbound network is considered higher risk and needs proper routing, monitoring, and issue mitigation. Infrastructure accounts inherit permission guidelines from the AWS Organizations management account and the infrastructure OU. Networking and security teams typically manage the majority of infrastructure in this account. The services in the accounts control both the data and network layer segmentation and security, helping to fulfill PCI DSS Requirements 1.2, 1.3, 1.3.1, 1.3.2, and 1.4.1 by implementing segmentation controls between the CDE and other systems, networks, and environments.

Figure 8 shows an example structure for the infrastructure OU. The infrastructure OU hosts the infrastructure production and non-production OUs for the Shared Services and Network Services accounts with associated AWS services for the *connected to* PCI DSS scope environment.

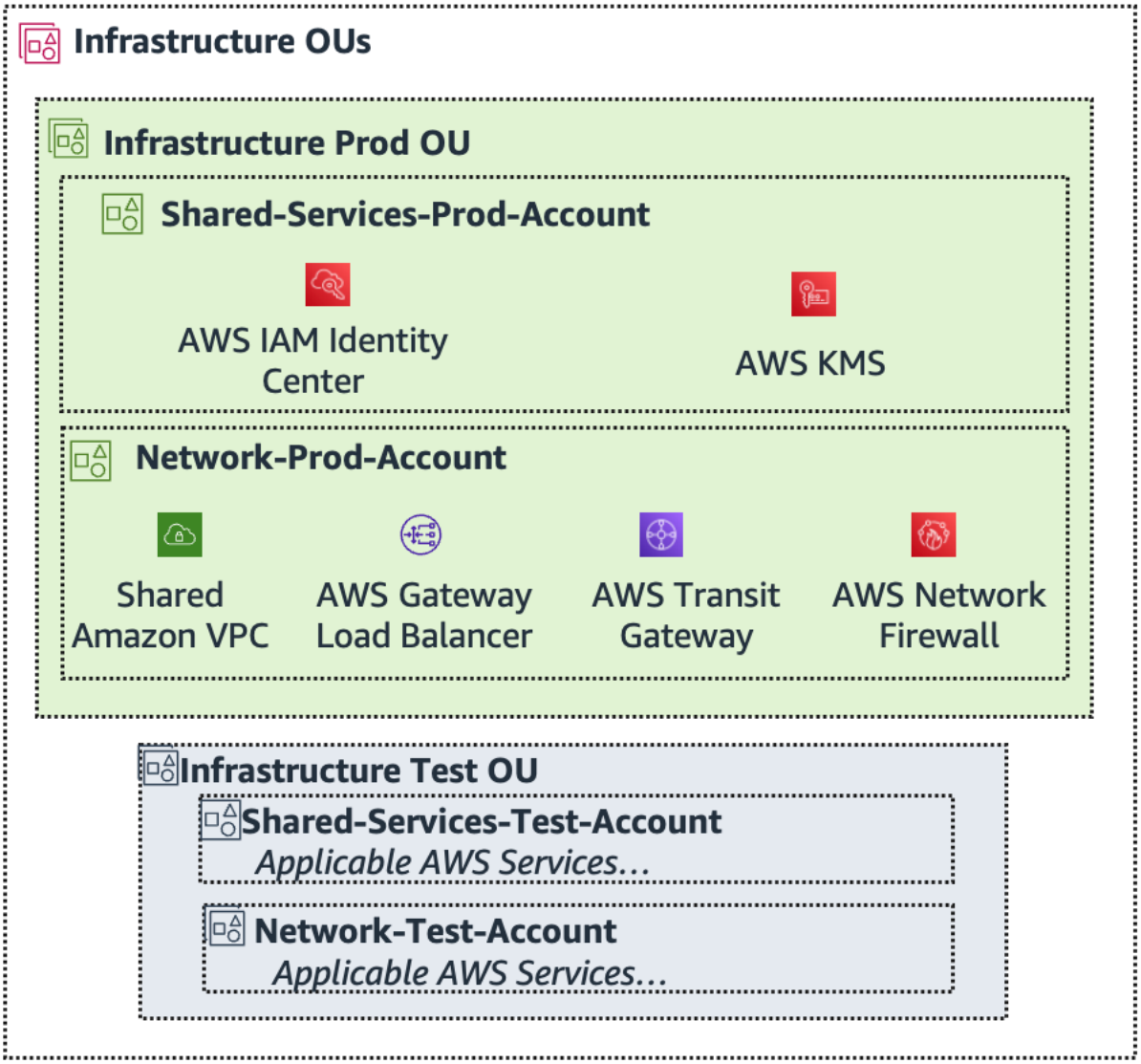

Figure 8: Recommended AWS Organizations member account structure for the infrastructure OU

In some cases, security and infrastructure teams configure shared services across the accounts, including PCI (provided that there is no CHD or SAD data storage or transmission), non-PCI, and non-production accounts. For example, you can use AWS Resource Access Manager (AWS RAM) to introduce share services such as Amazon VPC across accounts that are configured and operated by the infrastructure team, or AWS Network Firewall services that are configured and operated by the networking team.

Your organization may have other non-PCI workloads in the organization, and you may have a dedicated team that manages and governs them. You may create separate OUs and accounts for other non-PCI purposes.

PCI DSS segmentation is often referred to as PCI DSS Requirement 0. We recommend that you refer to the Guide for PCI DSS Compliance on AWS for the different PCI DSS requirements and how AWS services can be used to support your compliance needs. See Streamlining PCI compliance using AWS serverless architecture for the AWS serverless services that need to be included in your segmentation architecture. See Design principles for your multi-account strategy for AWS account best practices and structure. The Root and Nested Organizational Unit Support for Customizations for AWS Control Tower page includes additional benefits and best practices similar to those described in this whitepaper.

Out-of-scope OUs

Use separate AWS accounts to provision resources that don't require connectivity to or don't provide services for CDE systems. This separation helps ensure that resources in these accounts are adequately isolated from PCI in-scope systems by design, through inherent AWS account isolation. You can group these accounts under separate individual OUs according to business functions and needs. For recommendations on other types of OUs, see Organizing Your AWS Environment Using Multiple Accounts.

Network, application, and data security layers

At the organization level in AWS Organizations, segmentation boundaries play a crucial role in dividing the structure into distinct OU and account segments. This segregation serves to isolate PCI "in scope" workloads, effectively reducing unauthorized accesses at the account layer. The combination of SCPs and IAM policies works in tandem to define granular access controls, specifying who can access specific resources within accounts. This section describes how to implement network, application, and data segmentation layers with clear access boundaries to help with reducing compliance scope.

Network layer (OSI layers 3–4)

A security group is a feature of Amazon VPC that provides stateful network layer traffic filtering that works on the principle of an implicit deny. Only traffic defined in a security group can pass. A security group is similar to a host-based firewall and is associated with the network interface of an Amazon EC2 instance. You can use security groups to restrict network communication based on the port, and the source and destination addresses required to segment CDE systems from other *connected to* systems. Security groups are the core segmentation boundaries in an Amazon VPC, and you should design your network layer PCI DSS scoping strategy around

these security groups. With AWS software-defined networking capabilities, you can bring these stateful firewalls “closer” to your EC2 instances, and attach them directly to your network interfaces. Moving resources “closer” on the network means reducing the length of the network path between two resources. This reduces the number of network interconnection possibilities that might impact the *connected to* scope and potentially reduces the network latency. This means that neighboring EC2 instances might not be considered *connected to* and in-scope for PCI DSS if you have configured the security groups to restrict traffic. In traditional on-premises environments, a single host that contains account data would bring its entire subnet into scope because networking boundaries were present further up the network stack. You can also use security groups to restrict traffic flow between instances to help meet PCI DSS Requirements 1.2, 1.3, 1.3.1, 1.3.2, and 1.4.1.

You can attach security groups to Auto Scaling groups so that they apply to, and are removed from, instances in the group when the groups scale out and in. Between peered Amazon VPCs, you can chain together security groups so that one security group can reference the other security group as the source or destination, instead of using hard-coded IP addresses or ranges. This design helps automate the security group architecture and provides scalability by controlling peering traffic through security group membership instead of Classless Inter-Domain Routing (CIDR) ranges.

The default outbound configuration for a newly created security group does not meet PCI DSS Requirement 1.2.1 and 1.3.2 because it permits all outbound traffic. You must modify the default configuration for the security group to limit your outbound traffic to allow only necessary traffic.

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

Figure 9 shows an example of how to use security groups to achieve segmentation. As shown, security groups primarily define the network segmentation regardless of other network constructs, such as VPCs and network segments.

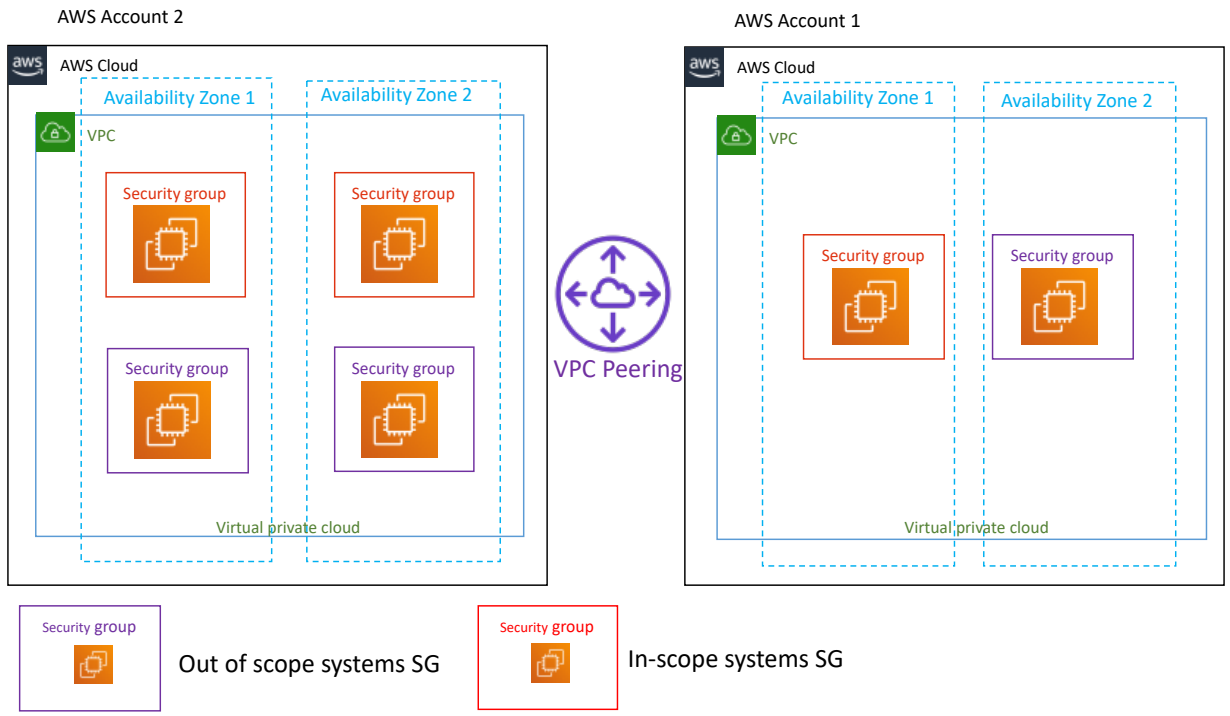

Figure 9: AWS VPC reference architecture

You should use one AWS account to provision your PCI DSS in-scope resources. However, you should control security group rules centrally for your CDE to help ensure that a change does not negatively affect the PCI DSS scope. You can achieve this by using AWS Firewall Manager, which provides you with the ability to centrally manage security groups across multiple accounts. With Firewall Manager, you can group resources by account, resource type, and tag to help you group your in-scope security groups and enforce policies across them. Firewall Manager helps you apply policies hierarchically, so you can delegate the creation of application-specific rules while retaining the ability to enforce certain rules centrally. Firewall Manager constantly monitors centrally-applied rules for accidental removal or mishandling to verify that the rules are applied consistently. With Firewall Manager, you can create policies that set guardrails defining which security groups are allowed or disallowed across your VPCs. You can get notifications of accounts and resources that are noncompliant or set up Firewall Manager to act directly through auto-remediation. Incorporating the ability of Firewall Manager to identify unused, redundant, or overly permissive security groups into your semi-annual firewall rule review process can help you meet PCI DSS Requirement 1.1.7 and 1.2.7.

If your organizational security controls require more advanced firewall capabilities or the centralization of ingress and egress traffic inspection within a VPC, consider using AWS Network Firewall. Network Firewall is a stateful, managed network firewall and intrusion

detection and prevention service for your VPC. With Network Firewall, you can filter traffic at the perimeter of your VPC. This includes filtering traffic to and from an internet gateway, NAT gateway, or over VPN or AWS Direct Connect. Network Firewall uses the open source intrusion prevention system (IPS) engine Suricata for stateful inspection. Usage of Network Firewall is not mandatory to meet PCI DSS requirements for network segmentation. However, the service provides the following additional security benefits for your VPC traffic:

- Only allows traffic from known AWS service domains or IP address endpoints, such as Amazon S3
- Uses custom lists of known bad domains to limit the domain names that your applications and resources can access
- Performs deep packet inspection on traffic entering or leaving your VPC
- Uses stateful protocol detection to filter protocols like HTTPS, independent of the port used

Application layer (OSI layer 7)

At OSI layer 7, the application that handles account data must manage the data flow and define segmentation boundaries. AWS provides many API-based abstracted services, such as Lambda, Amazon S3, and DynamoDB, that your organization can use to enable business functionality without the need to manage servers and Amazon EC2 instances. You can only communicate to most abstracted services through the service's encrypted API over HTTPS. For communication to be possible, you must also explicitly allow access to a service by using IAM policies. When you use AWS abstracted services, IAM policies become application layer logical segmentation boundaries based on the implicit deny in IAM policies and permissions. It is the customer's responsibility to design application-layer segmentation to reduce the PCI DSS scope when using abstracted AWS services.

Segmentation for abstracted AWS services

Abstracted services implement network isolation by design. Unlike traditional virtual servers in a subnet that have default network connections available to each other, abstracted service instances are segmented by default and only establish connectivity based on permitted events. For scoping, the focus is placed on the type of data traversing the connection, and the configuration and permissions of the abstracted service to allow data in or out. For example, a Lambda function will only ever communicate based on the code that you provision, and connect to only those resources that you explicitly configure. That same function can only be communicated to, or *connected to*, using explicitly allowed IAM permissions and configurations.

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 between the CDE resources and services and other web-based services. As shown in Figure 10, API Gateway can act as a "front door" to applications for accessing data, business logic, or functionality from backend services. This could include workloads running on Amazon EC2, code running on AWS Lambda, other supported AWS services, or web or mobile applications. You can use API Gateway as the CDE boundary interface to broker communication from CDE systems and services hosted on AWS. Here, the connections from CDE systems and services are terminated by your custom API instance, and a new connection is established from your API instance to the non-CDE destination system or services. Then, you can exclude from your PCI DSS scope resources outside of your AWS CDE that communicate with CDE systems through API Gateway, as long as the resource does not handle account data or provide security services to CDE systems. Only the API Gateway instance is in-scope as a connected system. Because this is a PCI DSS validated service, you do not need to maintain and validate the service itself and its ability to act as a segmentation boundary for PCI DSS compliance. You are still responsible for other aspects of the implementation as part of the Shared Responsibility Model. This includes, but is not limited to, requirements for access management with IAM and logging with AWS CloudTrail. Although API Gateway provides a secure, access-controlled web service API mechanism, applications validate incoming data, including monitoring for unexpected account data from a CDE.

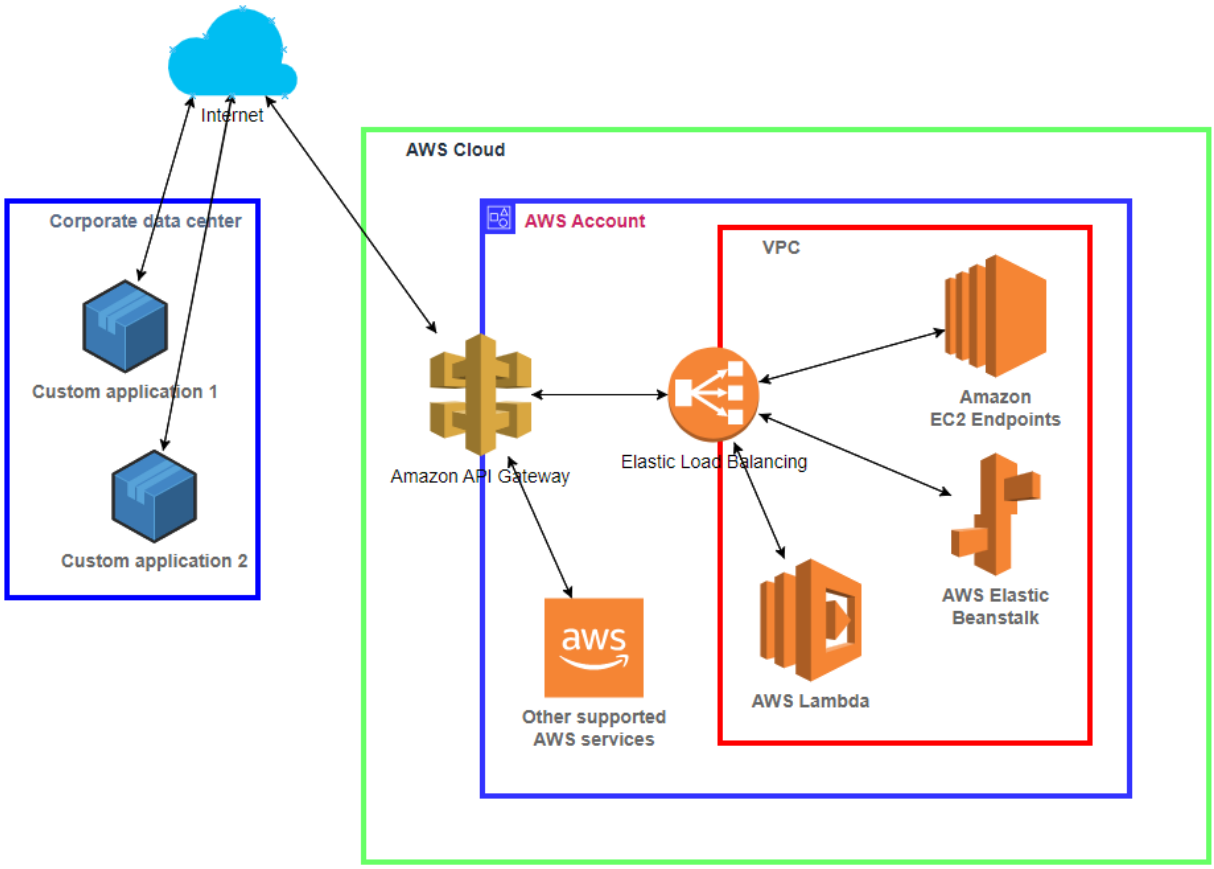

Figure 10: Segmentation using API Gateway for abstracted services

Scoping and segmentation for containerized workloads

Organizations use containers to quickly, reliably, and consistently deploy applications independent of the underlying infrastructure.

Containers allow you to create a self-contained standardized unit of software development that includes an application's code and related configurations and dependencies. You can use containers as scalable building blocks for a larger business application, to help deliver consistency in environments, support robust version control, and increase developer and operational efficiency. Amazon ECS and Amazon EKS are highly scalable, highly performant container orchestration services that you can use to run and scale containerized applications on AWS. AWS Fargate is a serverless compute engine, compatible with both Amazon ECS and Amazon EKS, that provides a greater level of abstraction, scope reduction, and segmentation than EC2 instances. If you are running or architecting containerized PCI DSS in-scope applications, you must verify that they are properly scoped and segmented. For specific guidance, see the following AWS whitepapers:

- Architecting on Amazon ECS for PCI DSS Compliance
- Architecting Amazon EKS for PCI DSS Compliance

Container scoping

Amazon ECS tasks and Kubernetes Pods on Amazon EKS are logical groups of running containers for the respective services. Both Amazon ECS and Amazon EKS support two launch types:

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

As described previously, you should use the PCI application account under the PCI OU to deploy your containerized workloads that handle account data. Then use a separate account under out-of-scope OUs to host containerized workloads that are isolated from in-scope containers. If you have containerized workloads that need to connect to in-scope workloads, but whose communication does not involve account data, you should consider hosting them in accounts under the PCI OU or in shared services accounts that contain other *connected to* or *security impacting* resources.

You can use AWS Cloud Map to dynamically discover resources and maintain an inventory of them, and assist with defining the scope of your containerized environment. AWS App Mesh can also help define your scope by maintaining the traffic flow and assisting with your data flow diagrams. You can use App Mesh to dynamically update traffic routing between your CDE services.

When technical or business constraints prevent you from using account-level segregation of containerized workloads, you should create separate ECS and EKS clusters and VPCs to group in-scope and out-of-scope containers. Customers should be aware that having in-scope and out-of-scope resources in the same AWS account imposes additional administrative and logical constraints in order to successfully keep the desired resources in that PCI account out of scope. Processes and resources that support these out-of-scope resources themselves are potentially in-scope because they have the ability to impact the security of the CDE. This includes, but is not limited to, access provisioning procedures for IAM permissions in the account, deployment processes that provision the out-of-scope resources that exist adjacent to in-scope resources, change management procedures for changes to out-of-scope resources in the in-scope account, and change detection for those supporting resources. The AWS account that contains in-scope clusters is part of the CDE, so you will need to demonstrate that even if the supporting processes that maintain the out-of-scope resources are compromised they *could not* impact the security of the CDE. This might be a difficult conversation with your QSA. They must agree that

your segmentation and separation efforts sufficiently rule your desired resources as out-of-scope after viewing evidence that supports your assertion.

Customers should place clusters using the EC2 launch type in separate VPCs to enhance the ability to segment at the network layer. These clusters are your lowest construct for PCI DSS scoping. Use further network controls, as described in the next section, to restrict inter-cluster communication to segment out out-of-scope clusters within the same account.

Segment ECS clusters

An Amazon ECS cluster is a logical grouping of tasks or services. Your tasks and services are run on infrastructure that is registered to a cluster. You can assign security groups to the Amazon ECS cluster and design security group rules to restrict network communication to only in-scope systems, isolate the cluster from out-of-scope system components, or both. Security groups act as a firewall for associated container instances, controlling both inbound and outbound traffic at the container instance level. An Amazon ECS cluster is the lowest construct that you can use to define your PCI DSS scope boundary.

For the AWS Fargate task launch type, tasks are the lowest scoping construct, so you don't have to worry about cluster assignment and scope. Each Fargate task has its own isolation boundary and does not share the underlying kernel, CPU resources, memory resources, or network interface with another task. You should group tasks running in-scope containers and use the awsvpc network mode in combination with security group rules to restrict communication between in-scope and *connected to* tasks.

Segment Amazon EKS clusters

An EKS cluster with the EC2 launch type has the following primary components:

- EKS control plane
- EKS nodes that are registered with the control plane

The control plane runs in an account managed by AWS, and the Kubernetes API is exposed through the EKS endpoint associated with your cluster. Each EKS cluster control plane is single-tenant and unique, and it runs on its own set of Amazon EC2 instances. EKS nodes run in your account and connect to your cluster's control plane through the API server endpoint and a certificate file that is created for your cluster. Your EKS cluster is created in a VPC. The Amazon VPC Container Network Interface (CNI) plugin provides Pod networking.

When you create a cluster, Amazon EKS creates a security group named `eks-cluster-sg-my-cluster-uniqueID`. The default rules allow all traffic to flow freely between your cluster and nodes and allow all outbound traffic to any destination. If you need to limit the open ports between the cluster and nodes, you can remove the default rules and add some minimum rules.

Use this cluster security group to segment network traffic from other EKS clusters. An EKS cluster is the lowest construct to define your PCI DSS scope boundary.

For the AWS Fargate task launch type, Kubernetes Pods are the lowest scoping construct, so you do not need to perform cluster assignment and design segmentation between Pod clusters. Each Pod that runs on Fargate has its own isolation boundary. It doesn't share the underlying kernel, CPU resources, memory resources, or network interface with another Pod. You should group Pods running in-scope in containers and use Pod networking in combination with security group rules to restrict communication between in-scope and *connected to* Pods.

Data segmentation layer

This layer uses AWS cryptographic services to help protect PCI in-scope application data in transit, in use, and at rest. These AWS cryptographic services can be used in combination to implement a robust data protection strategy for PCI applications. For example, you can use AWS Key Management Service (KMS) to encrypt data at rest and in transit, AWS CloudHSM for secure key storage and cryptographic operations, AWS Certificate Manager (ACM) for SSL/TLS certificate management, AWS Secrets Manager for secure storage of application secrets, and AWS Security Hub and AWS CloudTrail for monitoring and auditing. Using AWS cryptographic capabilities in conjunction with network and application segmentations, access controls, and security services can help you to achieve comprehensive data protection, auditing abilities, and alignment with compliance requirements.

Firewall rules

To bolster security, you can implement additional measures such as network, application, and data firewall rules within the segmentation boundaries. These rules contribute an extra layer of protection, allowing or denying permission within the organization structure to access and operate PCI "in scope" workloads. Their role extends to inspecting and controlling requests and responses, helping to ensure the security of sensitive data and services.

In essence, the implementation of the firewall rules described in this section serves as a robust defense mechanism, helping you to safeguard PCI workloads from both internal and external threats. By reducing the risk of unauthorized access to data and services, this comprehensive approach contributes significantly to maintaining the integrity and security of your organization structure.

Network firewall rules

Network firewall rules act as gatekeepers, regulating the flow of traffic in and out of a network, with the primary goal of safeguarding PCI workloads from unauthorized access and malicious activity. These rules play a pivotal role in determining the sources that are allowed to connect to

the network and specifying which protocols and ports are either permitted or blocked. The following is a list of firewall rules designed for network segmentation purposes:

- Block all inbound traffic, excluding traffic from known and trusted sources
- Use intrusion detection and preventive (IDP/IPS) rules to block malicious traffic
- Restrict outbound traffic, allowing communication only with known and trusted destinations
- Establish distinct firewall policies for various traffic types, such as PCI traffic, production traffic, or non-production traffic, thereby isolating PCI workloads from other workloads and reducing the risk of attacks
- Prohibit all traffic from identified malicious IP addresses and sources
- Only permit inbound traffic from the public internet on ports 80 and 443, which helps to prevent unauthorized users from accessing other ports
- Allow inbound traffic exclusively on port 443 if the traffic is encrypted
- Restrict outbound traffic on ports 22 and 3389 solely to known and trusted IP addresses, which helps to prevent unauthorized users from accessing PCI workloads through SSH or RDP

Application firewall rules

Firewall rules in this context actively defend against common web application exploits, such as SQL injection, cross-site scripting, and parameter tampering. These rules scrutinize incoming traffic, identifying and blocking requests that exhibit suspicious patterns or attempt to exploit known vulnerabilities. Following is a list of recommended application-level firewall rules:

- Use web application firewall (WAF) rules to protect against web-based exploits, including SQL injection and cross-site scripting
- Block application-level ingress or egress traffic from or to a specific AWS Region or country, as applicable
- Employ a content filtering service to block access to known malicious websites
- Implement IDP/IPS rules to handle known attack vectors and vulnerabilities
- Make TLS encryption mandatory for all incoming traffic through TLS listener configuration
- Use path patterns to route traffic to different backend targets, isolating PCI workloads from other workloads on the backend
- Implement sticky sessions to maintain user connections to the same backend target throughout the user's session, helping to prevent threat actors from performing session hijacking
- Restrict inbound traffic to allow only HTTPS traffic from known and trusted sources on port 443
- Prohibit all inbound traffic on ports 25 and 110 to help prevent spam and phishing
- Restrict outbound traffic to allow only communication with known and trusted destinations, such as payment processors, or other services that adhere to PCI standards

Data firewall rules

The primary objective of data firewall rules is to safeguard sensitive data, both during storage and transmission. You can accomplish this by implementing data encryption, access control policies, and data loss prevention measures. These precautions are instituted to thwart unauthorized access, modification, or theft of PCI data. Encrypt PCI data at rest, in motion, and in use.

Access policies

IAM policies play a critical role in determining who has access to security controls and the authority to operate the controls. These policies establish specific access controls, specifying that only authorized users and roles can access designated resources that manage security rules. Following are some recommendations for access policies:

- Apply the principle of least privilege access by using IAM to restrict access to workloads that adhere to PCI standards, allowing only essential access
- Employ security groups or network ACLs to restrict access to the Application Load Balancer (ALB) based on specific IP addresses or IP address ranges

Monitoring and auditing

Regular monitoring and auditing are essential for detecting and addressing potential security issues promptly. Following are some monitoring and auditing recommendations:

- Use AWS CloudTrail to establish a comprehensive audit trail that logs activities within the organization
- Implement AWS Security Hub for centralized security management and visibility across multiple AWS accounts

It's important to note that these guidelines and examples are general in nature. The specific firewall rules that you require will vary based on your unique environment and organizational needs. Additionally, it's essential to regularly review and update firewall rules to verify their ongoing effectiveness and relevance.

Scoping and segmentation validation

PCI DSS Requirement 11.3.4 states that you must perform penetration and segmentation testing at least annually for merchants and every six months for service providers. This step, as shown in Figure 11, corresponds to the “Assess and authorize controls” phase of the system lifecycle approach for security and privacy, as outlined in National Institute of Standards and Technology (NIST) SP 800-37. You should review the AWS Customer Service Policy for Penetration Testing before conducting this kind of security testing.

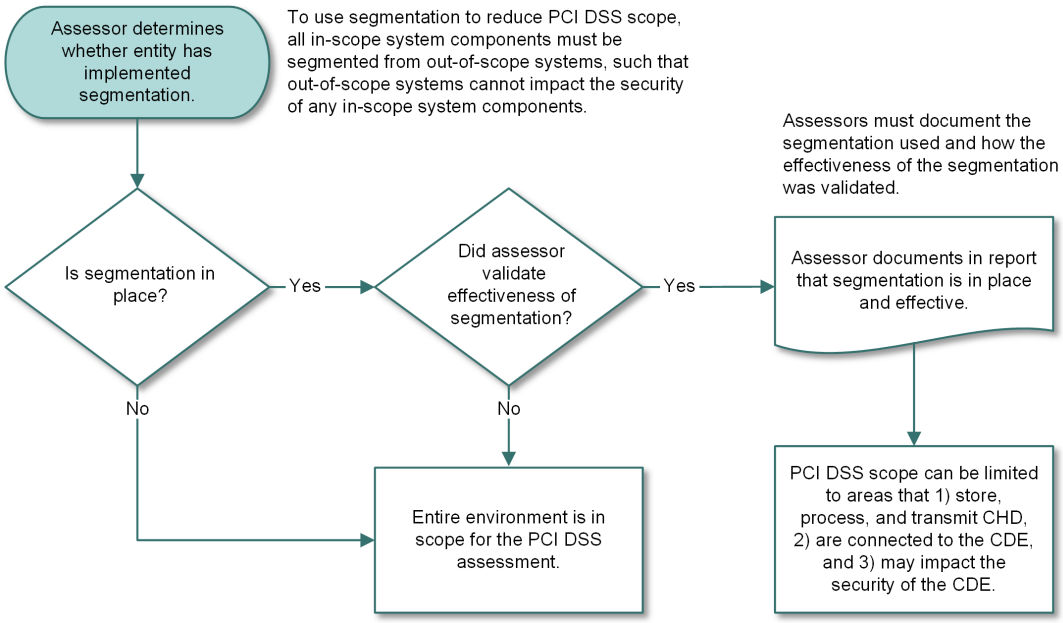

Figure 11: Segmentation and impact to PCI DSS scope (Source: PCI SSC Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation, Page 13)

According to the PCI DSS published Information Supplement: Penetration Testing Guidance, the scope for penetration testing includes the entire CDE perimeter and involves testing critical system components from both inside and outside the network. This applies both to the external perimeter (public-facing attack surfaces) and the internal perimeter of the CDE (LAN attack surfaces). You should also include remote access vectors in the testing, such as VPN connections to the CDE. You can use the Amazon VPC feature Reachability Analyzer prior to or as part of your segmentation testing to verify and validate network configurations and connectivity intents.

Whether you use infrastructure, managed, or abstracted services, you must perform some type of testing and validation of segmentation boundaries. Traditional segmentation methods, which are composed of logical connectivity checks between two points inside and outside your CDE, are needed for environments with infrastructure and managed services. For infrastructure services such as Amazon EC2 instances, and managed services such as Amazon RDS, security groups form the core segmentation boundaries.

For managed services, security groups provide similar segmentation boundaries because of the network interfaces involved. The testing aims to validate isolation of in-scope resources from out-of-scope resources. For container clusters with EC2 launch types, you should perform segmentation testing to validate segmentation between in-scope and out-of-scope clusters, and isolation of underlying EC2 instances between the in-scope and out-of-scope clusters. Amazon ECS and Amazon EKS clusters with the Fargate launch type are abstracted services because

AWS manages the underlying data plane and is responsible for the security and isolation of underlying compute resources.

For in-scope abstracted services, the only interface with the service is through an AWS service endpoint like `dynamodb.us-east-2.amazonaws.com`. The scanning and penetration testing of these endpoints is covered by the PCI DSS assessment for AWS as a service provider. However, you need to validate that the endpoints are restricted to your specific resources through IAM policies. When you use abstracted services, the account data flow and segmentation boundaries are controlled by the application and the application code. Therefore, you should focus on application testing to validate the segmentation boundaries as designed within the application. This focus helps ensure that the account data flow and the PCI DSS scope are maintained by the application as depicted in the account data flow diagram.

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

Validation of network traffic

In this section, we describe an OU in AWS Organizations and an account-segmented structure that can help you address the PCI DSS segmentation and compliance requirements effectively. Figure 12 maps the recommended organization design. The rest of this section describes how network traffic flows in this design and how various AWS services are organized to support PCI DSS compliance and segmentation from a network packet's perspective. We'll also cover how the various AWS services that are used for PCI segmentation control treat network packets and apply security inspections in order to protect the CDE and in-scope workloads.

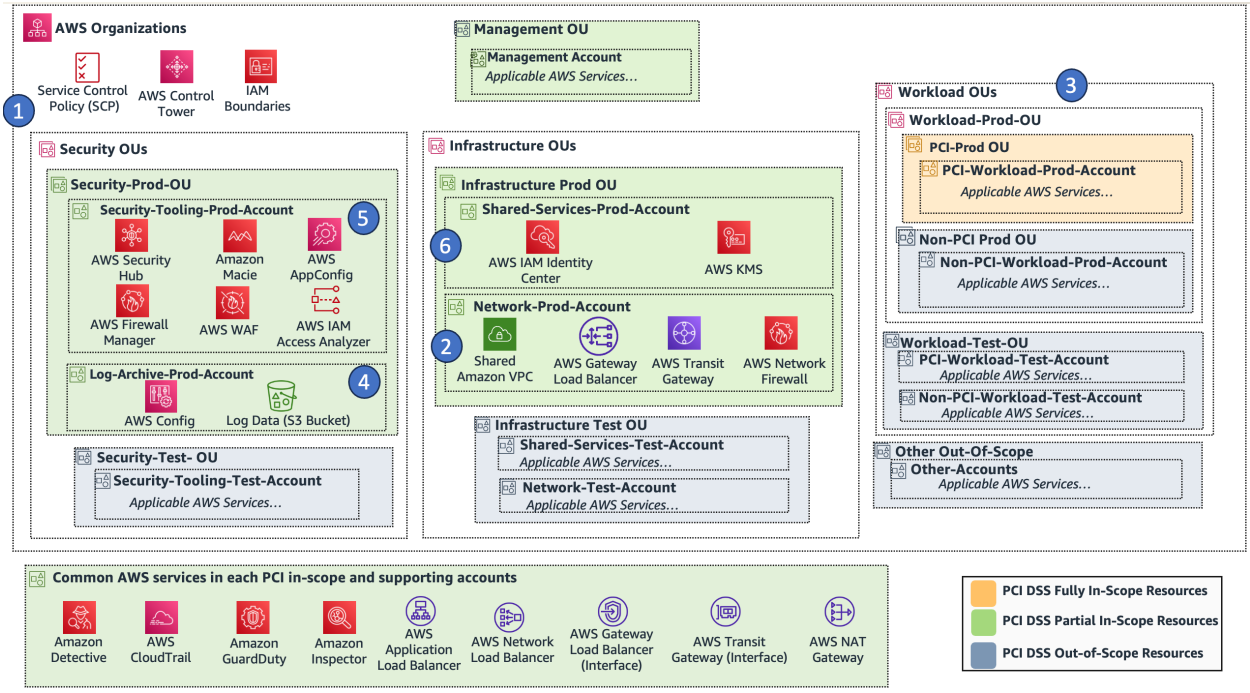

Figure 12: Network traffic journey through AWS account-level segmentation controls

The reference account structure illustrated in Figure 12 works as follows:

1. PCI DSS resources are logically separated and secured within AWS accounts by using the highest level of segmentation offered by AWS Organizations OUs and accounts. OUs logically group accounts into PCI segments, such as the PCI in-scope workload environment and PCI out-of-scope workload environment.
2. A Gateway Load Balancer (GWLB) routes network traffic to multiple VPCs based on its configuration, while AWS Transit Gateway serves as a central hub for inter-VPC traffic across multiple AWS accounts. As a network packet flows through the VPC, it is routed to the appropriate account based on subnet route tables and VPC security group rules. Network traffic must pass through security controls (security groups, network ACLs, network firewall rules), helping to ensure separation between in-scope and out-of-scope environments.
3. AWS accounts isolate in-scope and out-of-scope workloads, helping to ensure that network traffic enters only the appropriate accounts. An AWS SCP limit controls access to services and resources across OUs and accounts, while the network firewall restricts unauthorized access to traffic, providing complementary security measures.

4. The log archive account is exclusively designated to store immutable log data for security, operations, audit, and compliance. Services like AWS Config, AWS CloudTrail, residing in the log archive account, generate logs that capture actions performed by AWS control services. For instance, VPC Flow Logs sends logs to an S3 bucket as network packets traverse VPCs, while AWS WAF and AWS Network Firewall record security enforcement actions on network packets in an S3 bucket.
5. Security services and tools in the security tooling account enhance PCI environment security. For example, Security Hub helps to mitigate threats, Amazon Macie helps to safeguard sensitive data, Amazon GuardDuty detects and responds to security threats, AWS WAF and AWS Firewall Manager enforce policies to help protect CHD, SAD, and PII data, and Amazon Inspector analyzes, while IAM Access Analyzer assesses IAM policies for security risks.
6. The shared services account hosts centralized services for other accounts in the organization. AWS manages identity and access permissions, enabling the network packet to assume a role with authorized access to PCI resources. Additionally, AWS KMS handles encryption keys, enhancing the protection of stored data.

North-south network packet analysis

Continuing with the "network packet" persona, let's delve into its ingress and egress network journey, navigating through the segmentation boundaries of the Open Systems Interconnection (OSI) Model's Network (layer 3), Transport (layer 4), and Application (layer 7) layers. This journey involves intricate routing processes that guide the packet through its pathway.

As the packet progresses, it encounters an array of security controls and inspection mechanisms. These measures review and inspect the packet's content, contributing to the overall security posture of the network. We'll look at these measure in the next few sections.

Your organization may opt for a specific network architecture pattern that follows your organization's guiding principles, requirements, and policies. However, the core networking and security services governing segmentation, traffic routing, and security inspections will largely remain unaltered. For information about how to implement distributed inspection architecture, see Scaling network traffic inspection using AWS Gateway Load Balancer. See the publications Deployment models for AWS Network Firewall, Inspection Deployment Models with AWS Network Firewall, and Distributed Inspection Architectures with Gateway Load Balancer using third-party security appliances for centralized network traffic inspections that use Network Firewall and third-party security appliances.

Ingress traffic inspection for PCI DSS

Let's assume that the network packet request originated from the internet and is destined for the PCI-Workload-Prod account. For example, this could be a request from a payment processing or PCI-compliant service or similar service to access PCI-workloads in your CDE environment.

Figure 13 demonstrates how you can use AWS services and security controls to make sure that the network packet adheres to PCI DSS network segmentation requirements as it flows through the network layers.

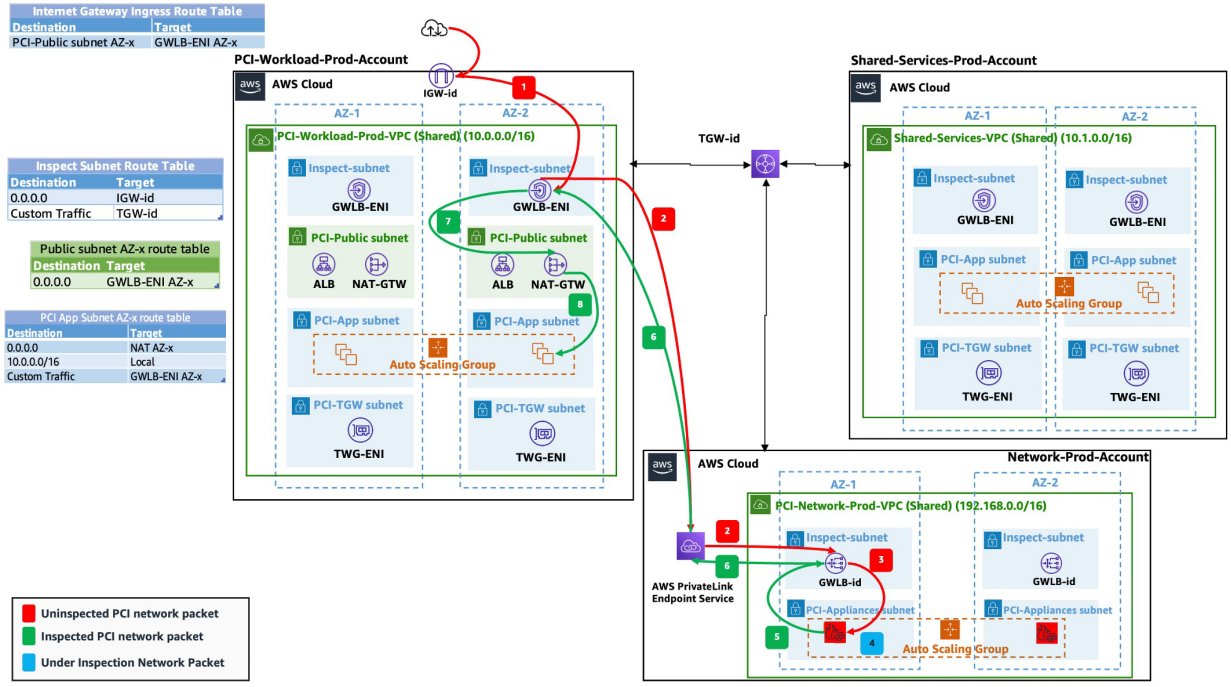

Figure 13: Ingress network packet traffic inspection journey

The process shown in Figure 13 is as follows:

1. As the network packet enters into the architecture, the uninspected ingress network packet destined for a PCI DSS-workload is routed to a Gateway Load Balancer endpoint (GWLB-ENI) for inspection using the Internet Gateway Route (IGW-id) route table.

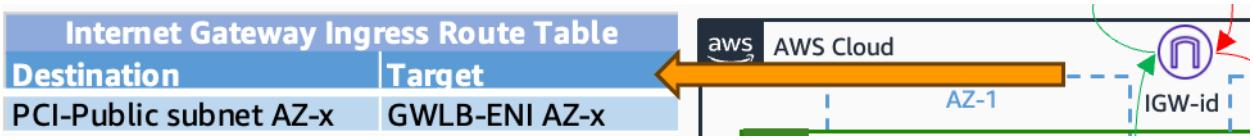

Figure 14: Internet Gateway ingress route table

2. AWS PrivateLink enables the private delivery of the network packet to the Gateway Load Balancer (GWLB-id) in the Inspect subnet of Network-Prod-Account, where it undergoes security inspection.
3. The Inspect-subnet route table directs the network packet to the firewall appliances located in PCI-Appliances-subnet.

4. The route table associated with the Inspect subnet within the VPC named PCI-Network-Prod-VPC in Network-Prod-Account directs network packets to the firewall appliances located in PCI-Appliances-subnet.

   This is where security controls, such as Network Firewall or AWS marketplace security appliances, are implemented to inspect the traffic. The security appliances' firewall rules and policies enforce traffic restrictions between PCI and non-PCI environments. They allow only necessary traffic, and specifically deny all other traffic.

   Based on the network firewall guidance provided in the previous section, the following firewall rules dictate whether network traffic is allowed or denied from proceeding.

   Note: This is not a comprehensive list of firewall rules. The information is for guidance purposes only.

Table 1 - Example firewall rules for ingress PCI network traffic control

Name	Source	Destination	Action	Notes
Allow traffic only from known CDE systems, *connected to*, and *security impacting* systems	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	Allow	Service requests, upon inspection by firewall rules and traffic policies that enforce PCI compliance and segmentation, shall be allowed.
	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Allow	
Allow traffic from known sources	[Known IPs/applications]	[PCI Workloads]	Allow	The traffic could be from external or internal known service or request sources which are allowlisted to allow traffic to PCI workloads.
Deny non-PCI workload traffic accessing PCI-	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	Deny	Blocks non-PCI traffic to PCI workloads.

Name	Source	Destination	Action	Notes
Production-Workload traffic	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Deny	
Block malicious IPs in	Malicious IPs sourced from IDP/IPS systems	Any	Deny	Log traffic data for further inspections.
Block unapproved applications	[Blocked Apps]	Any	Deny	
Block specific regional traffic	[Specific Regional Traffic]	Any	Deny	Log traffic data for further inspections.

Let's assume that the network packet is allowed to move forward within the Network layers 3 and 4 in the OSI model.

5. The network packet, having undergone inspection, now returns to the GWLB instance located in the Inspect subnet of Network-Prod-Account.
6. Using AWS PrivateLink, the inspected network packet is sent from Network-Prod-Account to the GWLB endpoint in PCI-Workload-Prod-Account, based on the configuration of the GWLB endpoint. The GWLB endpoint then forwards the inspected traffic to the Application Load Balancer (ALB).
7. The network packet has now reached the Application layer (layer 7). The network packet is inspected again at the Application layer by using ALB and AWS WAF (not shown in the diagram), which are attached to the ALB listener.
    - Refer to the **Application firewall rules** topic of the general firewall guidance section of this document. The application layer firewall rules help safeguard sensitive cardholder data by filtering out malicious traffic, guarding against unauthorized access, and enforcing gradual access controls. AWS WAF offers a compressive set of firewall rules that addresses these requirements.
    - The AWS network ACL is used to inspect and determine whether to allow or deny access on ports.
8. Once the packet is successfully inspected at the Application layer, it is routed to its destination.

Egress traffic inspection for PCI DSS

Let's assume that the network packet as a response originated from your CDE, for example PCI-workloads, and that it is destined for the internet. As shown in Figure 15, the packet could be a payment process confirmation or data shared with another PCI-compliant service or similar service's response, destined to go outside of your CDE.

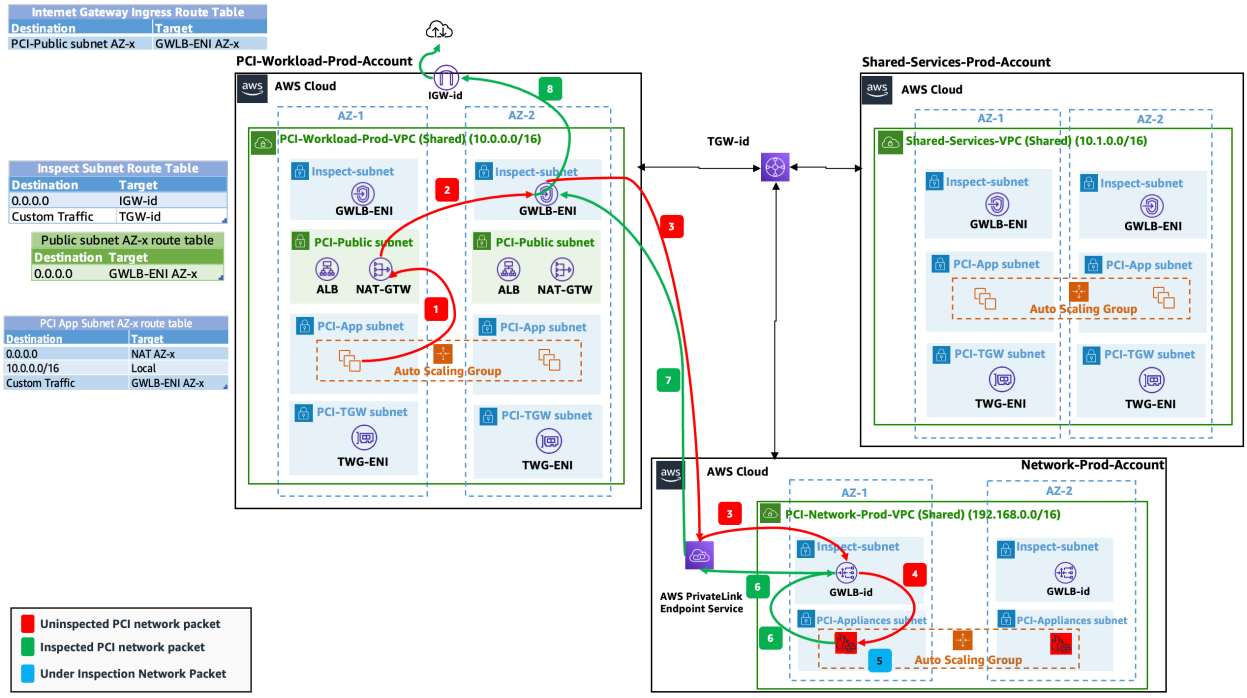

Figure 15: Egress network packet traffic inspection journey

The process shown in Figure 15 is as follows:

1. The egress network packet, originating from the PCI workload in the PCI-App subnet, is destined outbound to the internet. Based on the PCI-App-Subnet route table, the network packet gets routed to AWS NAT Gateway (NAT-GTW).

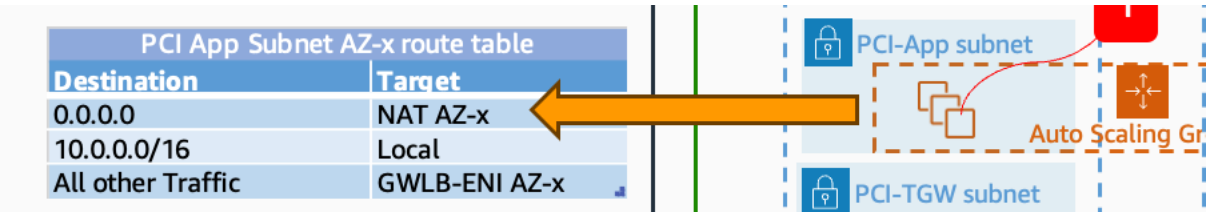

Figure 16: PCI App Subnet route table

2. Upon address translation by the NAT Gateway (NAT-GTW), the PCI-Public subnet routes the network packet to the Gateway Load Balancer (GWLB-ENI) endpoint.

Figure 17: Application Load Balancer network packet route

3. The PCI-Public subnet route table forwards the translated packets to the Gateway Load Balancer (GWLB-ENI) endpoint. AWS PrivateLink provides private delivery of the network packet to the Gateway Load Balancer (GWLB-id) in the Inspect subnet of Network-Prod-Account for the security inspection.

4. The network packet is routed to the security appliances residing in the PCI-Appliance subnet.

5. The firewall rules and policies of the security appliances (for example, AWS Network Firewall or third-party security appliances) enforce traffic restrictions. Outbound traffic restrictions are then enforced to align with PCI DSS requirement 1.3.2. Based on the network firewall guidance provided in the preceding section, the following firewall rules guide the network traffic and allow or deny it to move forward.

   (**Note:** This is not a comprehensive list of firewall rules. The information is for guidance purposes only).

Table 2 - Example firewall rules for egress PCI network traffic control

Name	Source	Destination	Action	Notes
Allow traffic only from known CDE systems, connected to, and *security impacting* systems	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	Allow	Service requests, upon inspection by firewall rules and traffic policies that enforce PCI compliance and segmentation, shall be allowed.
	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Allow	
Deny non-PCI workload traffic accessing PCI-	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	Deny	Block non-PCI traffic to PCI workloads.

Name	Source	Destination	Action	Notes
Production-Workload traffic	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Deny	
Allow traffic out to known destinations	[PCI Workloads]	[Known IPs/Applications]	Allow	The traffic could be destined to external or internal known targets which are allowlisted to allow traffic from PCI workloads.
Block malicious IPs out	Any	Malicious IPs or site addresses as a destination listed in the IDP/IPS systems	Deny	Log traffic data for further inspections.
Block unapproved applications	Any	[Blocked Apps]	Deny	Log traffic data for further inspections.
Block private network address Destination to Internet	Any	[Private network IP ranges]	Deny	Log traffic data for further inspections.
Block specific regional traffic	Any	[Specific Regional Traffic]	Deny	Log traffic data for further inspections.

Let's assume that the network packet is validated and allowed to move forward within the Network 3 and 4 layers in the OSI model.

6. The network packet, having undergone inspection, now returns to the GWLB instance located in the Inspect subnet of Network-Prod-Account.
7. Using AWS PrivateLink, the inspected network packet is sent from Network-Prod-Account to the AWS GWLB endpoint in PCI-Workload-Prod-Account, based on the configuration of the GWLB.

8. The scrutinized traffic is directed back to the Gateway Load Balancer (GWLB) endpoint in the Inspect subnet of the PCI-Workload-Prod-VPC account. The GWLB endpoint uses the Inspect subnet route table to forward the outbound network packet to the internet gateway (IGW-id), destined for the internet.

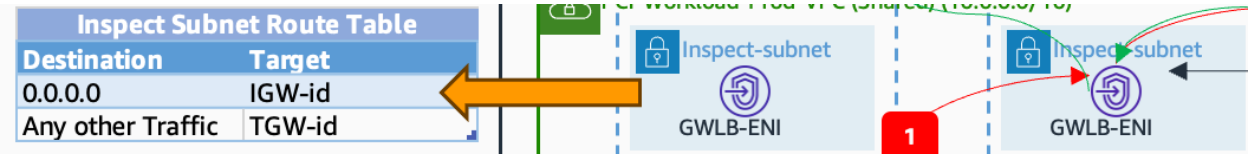

Figure 18: Inspect subnet route table

East-west network traffic analysis

This section and Figure 19 describe the network packet inspection journey, which originates internally in Shared-Services-Account and is destined to go to PCI-Workload-Account. Read about the Transit Gateway appliance mode in the Inspection VPC TGW attachment to learn more about how to maintain flow symmetry.

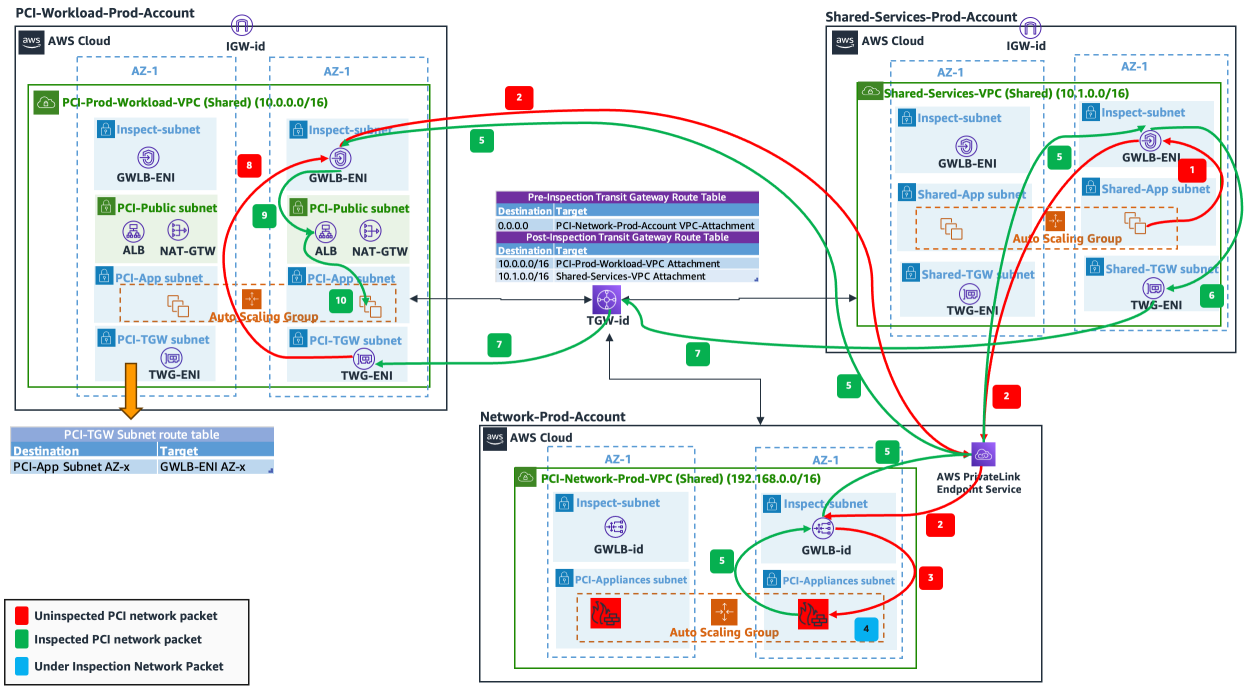

Figure 19: East-west network packet traffic inspection journey

The process shown in Figure 19 is as follows:

1. The network packet, originating from the Shared-Services-Prod-Account workload and destined for PCI Workload resources in the PCI-Workload-Prod-Account, undergoes

security inspection based on PCI compliance criteria. The PCI-App subnet routes the network packet to the GWLB endpoint situated in the Inspect subnet.

2. The GWLB endpoint routes the network packet through AWS PrivateLink to the GWLB instance that resides in Network-Prod-Account for inspection.
3. The Inspect subnet routes the network packet to the PCI-Appliance subnet using the route table for PCI compliance inspection.
4. The security appliances, such as AWS Network Firewall or third-party security appliances, enforce firewall rules and policies. These rules restrict traffic between PCI and non-PCI environments, permitting only necessary traffic while explicitly denying all other traffic.

   Based on the network firewall guidance provided in the previous section, the following firewall rules guide the network traffic and allow or deny it to move forward.

   (**Note:** This is not a comprehensive list of firewall rules. The information is for guidance purposes only).

Table 3 - Example firewall rules for east-west PCI network traffic control

Name	Source	Destination	Action	Notes
Allow traffic only from known CDE systems, *connected to*, and *security impacting* systems	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	Allow	Service requests, upon inspected by firewall rules and traffic policies that enforce PCI compliance and segmentation, shall be allowed.
	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Allow	
	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Shared-Services-Prod-Account: Shared-Services-VPC (10.1.0.0/16)]	Allow	
	[Example: Shared-Services-Prod-Account: Shared-	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Allow	

Name	Source	Destination	Action	Notes
	Services-VPC (10.1.0.0/16)]			
	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	[Example: Shared-Services-Prod-Account: Shared-Services-VPC (10.1.0.0/16)]	Allow	
	[Example: Shared-Services-Prod-Account: Shared-Services-VPC (10.1.0.0/16)]	[Example: Network-Prod-Account: PCI-Network-Prod-VPC (192.168.0.0/16)]	Allow	
Deny non-PCI workload traffic accessing PCI-Production-Workload traffic	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	Deny	Block non-PCI traffic to PCI workloads
	[Example: Non-PCI-Workload-Prod-Account: VPC (xx.xx.xx.xx/xx)]	[Example: PCI-Workload-Prod-Account: VPC (10.0.0.0/16)]	Deny	

Let's assume that the network packet is allowed to move forward within the Network 3 and 4 layers in the OSI model.

5. The network packet is routed back to the GWLB instance upon successful completion of the inspection.
6. The GWLB endpoint in the Inspect subnet of Shared-Services-Prod-Account receives the previously inspected network packet.
7. The GWLB endpoint uses the Inspect subnet route table to forward the network packet to the TGW endpoint (TGW-ENI) in the PCI-TGW subnet. The TGW route table forwards the network packet to the PCI-TGW subnet of the PCI-Prod-Workload-VPC in PCI-Workload-Prod-Account.

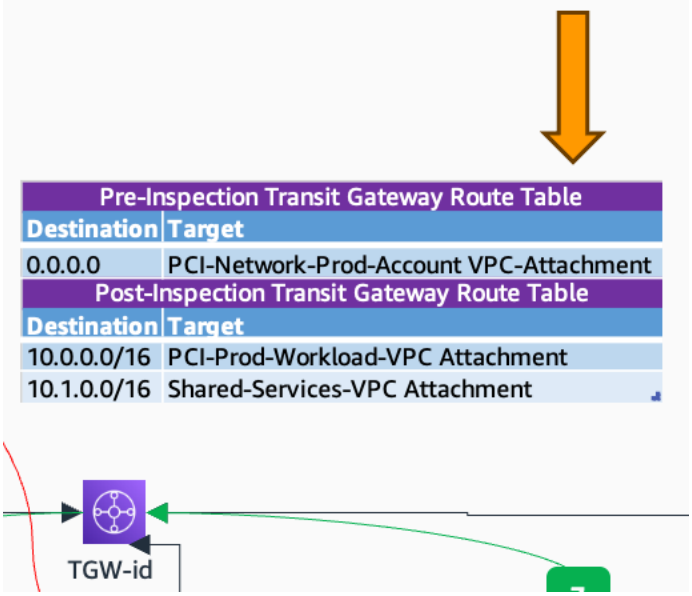

Figure 20: Transit Gateway route table

8. Although the network packet was inspected earlier, entry into the PCI workload OUs or account causes the network packet to be treated as uninspected traffic. The PCI-TGW subnet routes the traffic to the GWLB (GWLB-ENI) that resides in the Inspect subnet of the same account.

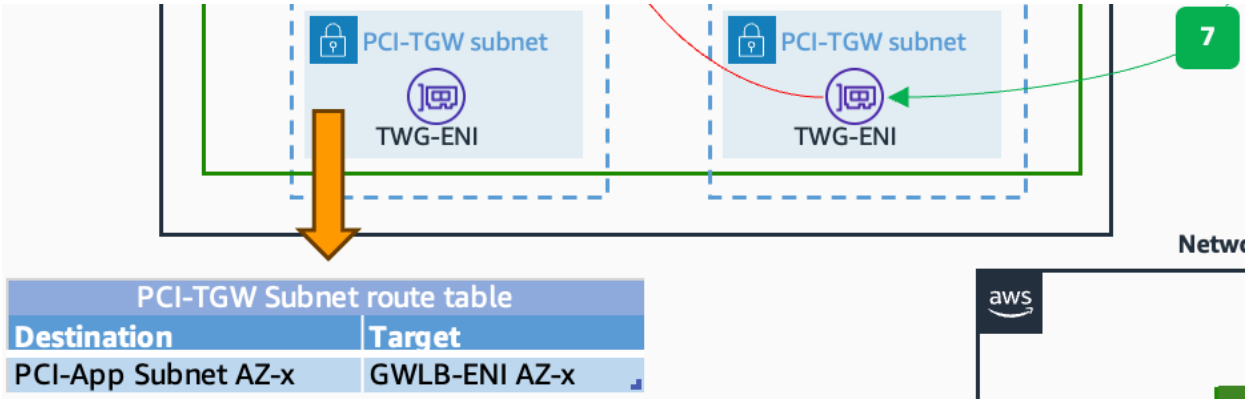

Figure 21: Transit Gateway interface traffic routing

The uninspected network packet is then routed to the Gateway Load Balancer endpoint residing in the Inspect subnet of the PCI-Prod-Workload-VPC. The network packet undergoes re-inspection following the same flow as described in steps 2, 3, 4, 5, and 6 in this section.

9. The network packet, now inspected, reaches the Application layer (layer 7) in the OSI model. The network packet is inspected again at the Application layer by using ALB and AWS WAF (not shown in the diagram) attached to the ALB listener.
    a. Refer to the **Application firewall rules** topic of the general firewall guidance section of this document. The Application layer firewall rules help safeguard sensitive cardholder data by filtering out malicious traffic, helping to deny unauthorized access, and enforcing gradual access controls. AWS WAF offers a comprehensive set of firewall rules that addresses these requirements.
    b. The AWS network ACL is used to inspect and determine whether to allow or deny access on ports.

The network packet then reaches its destination workload residing in the PCI-App subnet.

Proactive security controls

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

This section provides information on monitoring the status of implemented segmentation controls. This monitoring helps ensure that the defined PCI DSS scope is not violated intentionally or erroneously, and is required by PCI DSS Requirement 10.8 (*for service providers only*) and PCI DSS Requirement 10.7. You should design the preventive and detective controls so that in case of a violation, respective stakeholders are notified as early as possible and remediation steps can be taken immediately. As your security posture matures, you should automate most responses so that deviations can be remediated without human intervention on a near real-time basis. The following are some ways that you can monitor your segmentation boundaries:

- Periodically validate security group rules against the planned CDE scope
- Manage security group and network configurations with a change control process
- Monitor security group rule changes in the CDE systems account
- Monitor Amazon VPC peering connections to the CDE systems account
- Monitor configuration changes to in-scope APIs or AWS services that allow data in or out of the CDE
- Implement data loss prevention controls on *connected to* systems to help prevent account data leakage and to validate scope boundary

You can automate responses by using AWS Config, which provides you with a resource configuration history and configuration change notifications to help with governance and security. You can create custom AWS Config rules to monitor changes to security groups, Amazon VPC peering connections, Amazon API Gateway APIs, and other resources on AWS that enforce segmentation boundaries. Attach AWS Config rules to appropriate Lambda responders to evaluate deviations and initiate automatic remediation when a change violates the defined PCI DSS segmentation boundaries.

You can use AWS CloudTrail to monitor configuration changes for your AWS resources. You can configure CloudWatch events to initiate Lambda responders to take remediation action on your behalf. These are a sample of the various AWS services that you can use to design and automate security controls for your CDE on AWS.

You can also integrate most of these services and other third-party tools and applications with AWS Security Hub. You can activate the PCI DSS standard in Security Hub to help you monitor the status of relevant security controls to help you meet particular PCI DSS requirements. This includes monitoring segmentation controls. When you detect a deviation through Security Hub, you can define further downstream actions that might include integrating with your workflow ticketing systems such as Jira, or defining Security Hub custom actions to send findings and results to Amazon EventBridge.

Feedback loop

Your business is constantly changing, and the design and functionality of your PCI DSS cardholder data environment and associated AWS service choices can change. Apart from business requirement changes, you should also develop your AWS infrastructure to make it more secure, efficient, and manageable. This constant change makes it important to establish a feedback loop in the security and segmentation controls design process. You should establish channels to gather feedback from the previous phases and from industry peers and use this feedback to make your current process better and more secure. The feedback loop and the associated changes might require reevaluation of your current segmentation controls that were implemented to define the PCI DSS scope. This reevaluation can also stem from a change in your organization's CHD flow. Regardless of the reasons, there must be at a minimum a yearly process of validating your established PCI DSS scope and recategorizing systems in the scope, if required.

Conclusion

This whitepaper provides cloud-based segmentation strategies for PCI DSS on AWS. It details how you can use AWS accounts, security groups, and services to help establish isolation for your cardholder data (CDE) that adheres to PCI standards. While core PCI DSS principles

remain constant when you move to the cloud, AWS offers unique implementation methods. Remember, validated AWS services alone don't guarantee compliance—proper architecture is essential. Our reference guide provides a blueprint for building a custom, secure AWS environment tailored to your needs. This guide helps you to simplify PCI DSS scoping and segmentation, strengthen governance, and safeguard cardholder data. In addition, as part of the shared responsibility model, using PCI DSS validated AWS services does not imply that use of those services automatically leads to the achievement of PCI DSS compliance for your environment. You must use and architect those services in a PCI DSS compliant manner. Your organization is responsible for the scope, design, and determination of security controls. AWS can help streamline your journey of achieving compliance with confidence and building trust with your customers.

Contributors

Contributors to this document include:

- Ted Tanner, Principal Assurance Consultant, PCI DSS QSA, AWS Security Assurance Services LLC
- Abdul Javid, Sr. Security Assurance Consultant, PCI DSS QSA, AWS Security Assurance Services LLC
- Padmakar Bhosale, Sr. Technical Account Manager, AWS Enterprise Support

Further reading

For more information, see the following resources:

- Payment Card Industry Data Security Standard (PCI DSS) 4.0 on AWS
- Architecting on Amazon ECS for PCI DSS Compliance
- Architecting Amazon EKS for PCI DSS Compliance
- PCI DSS and AWS Foundational Security Best Practices on AWS
- Audit companion for the AWS PCI DSS Quick Start
- AWS Whitepapers and Guides
- 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

- Organizing Your AWS Environment Using Multiple Accounts
- PCI SSC Cloud Computing Guidelines
- PCI DSS Virtualization Guidelines
- Transforming transactions: Streamlining PCI compliance using AWS serverless architecture

Document revisions

Date	Description
April 2019	First publication
May 2023	Second publication
November 2024	Third publication