AWS における PCI DSS スコーピングとセグメンテーションのためのアーキテクチャ

適切なセグメンテーションコントロールによる
PCI DSS スコープの特定と最小化

First published April 2019

Last updated May 2023

お知らせ

お客様は、この文書に記載された情報を独自に評価する責任を負います。本書は、(a)情報提供のみを目的とし、(b)現在の Amazon Web Services（AWS）の製品提供および実務を表しており、予告なく変更される場合があること、(c) AWS およびその関連会社、サプライヤー、ライセンサーからいかなる約束や保証を行うものではありません。AWS の製品またはサービスは、明示または黙示を問わず、いかなる種類の保証、表明、または条件もなく、「現状のまま」提供されます。AWS のお客様に対する責任と義務は、AWS の契約によって管理され、本書は AWS とそのお客様との間の契約の一部ではなく、またそれを修正するものではありません。

AWS Security Assurance Services, LLC (AWS SAS) は、Amazon Web Services (AWS) の完全子会社です。AWS SAS は独立した PCI QSA 企業（QSAC）であり、AWS のお客様とパートナーに PCI DSS 準拠に関する具体的かつ規定的な情報を提供します。PCI QSAC として、AWS SAS は PCI の機密保持と契約の枠組みのもと、PCI Security Standards Council (SSC)または他の PCI QSAC と対話できます。

目次

要旨

本書では、AWS クラウド上で稼働する PCI DSS（Payment Card Industry Data Security Standard）ワークロードのスコープを適切に定義する方法、およびクラウドネイティブの AWS サービスを使用してスコープ内リソースとスコープ外リソースの間のセグメンテーション境界を定義する方法についてガイダンスを示します。本書はエンジニアやソリューションビルダーを対象としていますが、AWS で利用可能なセグメンテーションコントロールと関連するスコープの考慮事項をよりよく理解するための、有資格セキュリティ評価者（QSA）および内部セキュリティ評価者（ISA）のガイドとしても機能する内容となっています。

イントロダクション

AWS 上の Software-Defined Networking は、アプリケーションのスコーププロセスをオンプレミス環境から変化させます。AWS で利用できる追加のセグメンテーションコントロールは、単なるネットワークのセグメンテーションにとどまりません。アプリケーションを設計する際に、セキュリティに影響を与えるサービスや必要なコントロールを慎重に選択することで、カード会員データ環境（CDE）のシステムやサービスの数を大幅に削減できます。これにより、コンプライアンスのコストと労力を大幅に削減できます。安全な AWS ワークロードの設計、配信、および保守のためのセキュリティベストプラクティスと推奨事項を適用する方法の詳細とガイダンスについては、AWS Well-Architected Framework セキュリティの柱を参照してください。

AWS における PCI DSS スコーピングプロセス

プロシージャーやアプリケーションコードとの相互作用を含め、アプリケーションおよび環境内のアカウントデータの完全なフローを理解することは非常に重要です。環境内のデータフロー、および接続されサポートするすべてのシステムコンポーネントの評価は、PCI DSS 要件の

適用可能性を決定し、CDE の境界とコンポーネント、および PCI DSS 評価のスコープを定義するものです。

スコーピングの理解

PCI DSS の要件は、アカウントデータを保存、処理、または送信する組織に対して義務付けられています。アカウントデータは、カード会員データ（CHD）と機密認証データ（SAD）で構成されています。PCI Security Standards Council が維持するこのデータセキュリティ基準には、機密性の高いクレジットカード情報を保護するためのコントロールとして知られる IT セキュリティ要件の規定セットが含まれています。CHD は、プライマリアカウント番号（PAN）で構成され、PAN に含まれる場合は、カード所有者の名前、カード有効期限、サービスコードも含まれます。SAD は、磁気ストライプまたはトラックデータ、カード検証コード、および PIN または PIN ブロックを含みます。PAN は、カード会員データの定義要素であり、関連する CDE を決定するための基礎です。サービスプロバイダにとって、PCI DSS への準拠はお客様要件でもある場合があります。組織は、PCI DSS 評価のスコープと呼ばれる CDE を正しく特定し、定義する責任を負います。CDE は、アカウントデータのセキュリティと相互作用する、または影響を与える人、プロセス、およびテクノロジで構成され、PCI DSS のスコープになります。本書では、CDE のスコープに含まれるテクノロジーの側面に焦点を当て、人やプロセスには目を向けません。PCI DSS のスコープとアカウントデータに関する追加情報は、PCI DSS v4.0 の 8 ページに記載されています。

PCI DSS では、システムコンポーネントという用語を使用して、CDE のスコープ内および一部とみなされるリソー スを特定しています。システムコンポーネントとは、アカウントデータを保存、処理、または送信し、前述の PCI リソースに無制限に接続できる、またはその他の方法で環境のセキュリティに影響を与える可能性がある技術リソースの総称です。AWS では、AWS アカウントデータを扱う AWS リソースおよびサービス、または PCI の対象であるアカウントおよびリソースをサポートするサービスが含まれます。これには、Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Relational Database Service (Amazon RDS)、または AWS Lambda など、カード会員データの保存または処理に積極的に関与するサービス、 , AWS Config, AWS Identity and Access Management (IAM), および AWS Organizations など、サポートサービスを含めることができます。

セグメンテーション

セグメンテーションは、俗に PCI DSS 要件 0 と呼ばれ、正式な PCI DSS 要件ではありません。セグメンテーションは、他の PCI DSS 要件に対処する前に、PCI DSS のスコープを決済フローに関与するシステムに限定するために使用されます。従来、組織はネットワークセグメンテーションを主要な制御として使用し、PCI DSS の適用スコープ内の環境を制限して、その環境を IT インフラの残りの部分から保護しています。このアプローチは、組織がスコープ外のインフラストラクチャを保護しないことを意味するものではなく、むしろ、セキュリティ制御の選択に柔軟性があり、スコープ外のインフラストラクチャの検証要件が異なることを意味します。

PCI SSC による Guidance for PCI DSS Scoping and Network Segmentation では、IT インフラおよびリソースを PCI DSS のスコープに関して以下のセグメントに分類しています：

- **CDE システム：** CDE システム：これらのシステムコンポーネントは、アカウントデータ（CHD および SAD）を保存、処理、または送信します。CDE システム：これらのシステムコンポーネントは、アカウントデータ（CHD および SAD）を保存、処理、または送信します。また、無制限のネットワークアクセスを持つ同じネットワークセグメント上のシステムコンポーネントも含まれる場合があります。これらのコンポーネントは PCI DSS の適用スコープ内であり、他のリソースをスコープにする可能性があります。
- **システムに接続されセキュリティに影響を与えるコンポーネント：** これらのシステムコンポーネントは、CDE システムに直接または間接的に制限された接続を持つか、CDE システムに何らかの管理サービスまたはセキュリティサービスを提供しています。これらのコンポーネントは、1 つまたは複数の PCI DSS 要件、およびセグメンテーションの境界を確立するのに役立ったりする場合があります。これらはスコープ内のシステムですが、PCI DSS のスコープを他のリソースに拡大することはありません。

- **対象外システム：** これらのシステムコンポーネントは、前述の基準を満たさず、CDE システムのセキュリティまたは構成に影響を与えません。システムがスコープ外とみなされるには、スコープ外のシステムを使用してスコープ内のシステムコンポーネントを侵害することができないことを保証するための制御が実施されている必要があります。これらのシステムは、スコープ内とはみなされません。

PCI DSS のスコープを最小化するための戦略は、以下の 3 つのカテゴリを対象としています：CDE、接続またはセキュリティに影響を与えるコンポーネント、およびスコープ外のものです。本書では、ハイブリッド環境におけるスコープとセグメンテーションの検討についても取り上げます。この環境では、スコープ内のワークロードがオンプレミスのデータセンターと AWS クラウドにまたがる場合があります。

オンプレミスのスコーププロセスと同様に、AWS PCI DSS 環境のスコープは、アカウントデータフローから始まります。AWS クラウドとの大きな違いは、CHD および SAD フローの多くのネットワークセグメントに、Amazon API Gateway や AWS WAF などの AWS 専用のサービスが含まれる可能性があることです。これらのサービスには明確に定義された接続構成があり、AWS PCI DSS Level 1 Service Provider assessment の一部として評価されます。これらのサービスの AWS エンドポイントは、AWS PCI DSS スコープの一部としてファイアウォール機能で保護された RESTful Web サービスインタフェースであり、アカウントデータを受信しないサービスのセグメンテーション境界として機能します。正確な評価範囲を決定するために、文書化されたデータフローに対して、AWS サービスとネットワーク機能によって提供されるセグメンテーション機能を慎重に検討する必要があります。本書では、これらのセグメンテーション機能について学びます。

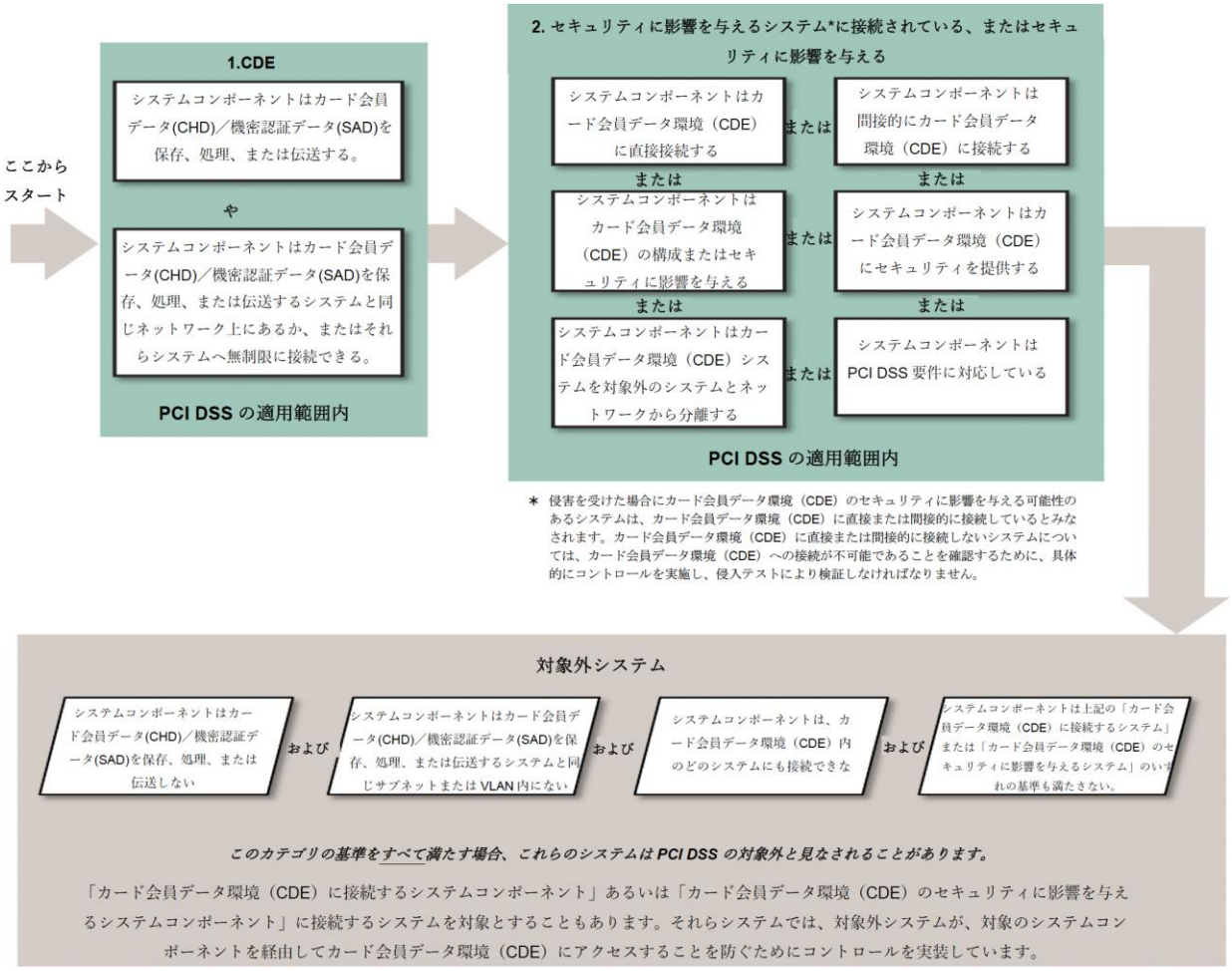

図 1: PCI DSS スコーピングの概要

インフラストラクチャサービス

データレイヤーとネットワークレイヤーの両方を制御するインフラストラクチャサービスの場合、追加のスコーピング手順を踏む必要があります。まず、アカウントデータを保存、処理、または送信するインスタンスを特定する必要があります。次に、AWS リソースに接続され、セキュリティに影響を与えるコンポーネントを特定する必要があります。例えば、アカウントデータを扱うウェブサービスを実行している Amazon EC2 インスタンスは、スコープ内に入ります。しかし、この EC2 インスタンスは、ネットワーク接続の定義と管理を担当するため、他の

多くの接続先リソースをスコープに入れることができます。AWS のセキュリティグループは、EC2 インスタンスの仮想ファイアウォールとして機能し、送受信トラフィックを制御します。EC2 インスタンスに接続されたセキュリティグループが適切に構成され制限されていない場合、同じサブネットまたは仮想プライベートクラウド（VPC）内でこれらのインスタンスとの間でネットワーク接続を確立できる他のインスタンスは、潜在的にシステムに接続されており、PCI DSS のスコープと見なされます。これらの接続されたシステムには、レポート用に非 CHD データをポーリングする監視サーバなどのシステムが含まれる場合があります。また、EC2 インスタンスやその他の AWS リソースが接続されていると考えられる場合も、セキュリティに影響を与える可能性があります。セキュリティに影響を与えるシステムには、EC2 インスタンスのウイルス対策などのセキュリティツールをホストする EC2 インスタンスや、認証と認可を提供する AWS Directory Service for Microsoft Active Directory t などの他の AWS サービスが含まれ得る。この同じスコープガイダンスは、EC2 起動タイプで Amazon Elastic Kubernetes Service (Amazon EKS)や Amazon Elastic Container Service (Amazon ECS)など、基盤となる EC2 インスタンスを管理する他の AWS サービスにも適用されます。

マネージド型サービス

マネージド型サービスでは、スコープやネットワーク接続に関して、インフラストラクチャサービスと同様の検討が必要です。ただし、AWS のマネージド型サービスでは、AWS は基本となるリソースインフラストラクチャの多くを管理します。これにより、オペレーティングシステムの保守の責任が抽象化され、リソースレベルでの PCI DSS スコープが縮小されます。ただし、サービスの抽象化されていない部分はユーザーの責任であり、ネットワークセキュリティも含まれます。たとえば、Amazon RDS をセットアップする組織は、VPC の Elastic Network Interface に適切なセキュリティグループを適用し、IAM を使用してサービスへのユーザーアクセスを制限することによって、ネットワークのセグメント化を確認する必要があります。マネージド型サービスインスタンスは、設計上隔離されており、あなたが管理する設定によって明示的に許可されない限り、サービスの異なるインスタンス間でデータが共有されないようにします。

抽象化されたサービス

設計上、抽象化されたサービスは、明示的に許可されていない限り、サービスの異なるインスタンス間でデータが共有されないことを検証できるように保護されています。アカウントデータを保存、処理、または送信する抽象化されたサービスの場合、対象となるコンポーネントは、アカウントデータを処理する AWS サービスの特定のインスタンスです。また、抽象化されたサービスインスタンスに接続する、またはそこから接続される特定のリソースも対象です。例えば、ある組織では、多くの Amazon DynamoDB テーブルがプロビジョニングされているが、アカウントデータを保存または処理するためにそれらのテーブルのサブセットのみを使用しているかもしれない。この場合、アカウントデータの保存に使用されるテーブルは、その組織の PCI DSS スコープの一部となります。アカウントデータを処理する Lambda インスタンスについても同じことが言えます。関数を呼び出す当事者も、Lambda 関数がデータフローの一部として接続するリソースも、インスコープに含まれます。インスコープのリソースに接続されるのは、アカウントデータを当該抽象化されたサービスリソースに渡すシステムやサービスも、データフローにアカウントデータが含まれていなくても何らかの形で CDE に接続するものも含まれます。例えば、EC2 インスタンス上でホストされている決済アプリケーションサーバーから操作ログを受け取る Amazon Simple Storage Service (Amazon S3) バケットである。S3 バケットは、これらのログにアカウントデータが存在しなくても、接続されているとみなされます。この例では、機械学習とパターンマッチングを使用して AWS 上の機密データを発見し保護する、フルマネージドのデータセキュリティおよびデータプライバシーサービスである Amazon Macie を利用できます。PCI DSS のスコープが S3 ログバケットを越えて広がっていないことを確認するために、Macie を構成して、これらのバケット内のアカウントデータのオブジェクトを監視し、機密データが特定された場合にアラートを提供できます。

Elastic Load Balancing (ELB)などの一部の抽象化されたサービスは、ターゲットグループ内のリソースへの永続的なネットワーク接続を維持するので、スコープ作成プロセスの一環としてこれらのサービスの評価が必要です。

サブネット内の従来の仮想サーバーが、互いに利用可能なデフォルトのネットワーク接続を持つのとは異なり、抽象化されたサービスインスタンスは、デフォルトでセグメント化されています。これらのサービス間の接続は、永続的なネットワーク接続ではなく、オンデマンドのデ

ータ接続です。スコープについては、接続を横断するデータの種類と、データの出入りを許可するための抽象化されたサービスの構成と権限に焦点が置かれます。例えば、Lambda 関数は、あなたが用意したコードに基づいてのみ通信を行います。その Lambda 関数は、明示的に許可された IAM の権限と構成を使用してのみ、通信したり、接続したりできます。

抽象化されたサービスがアカウントデータを扱わず、CDE のセキュリティに影響を与えず、アカウントデータを扱うリソースと通信しない場合、アカウントデータがこれらの抽象化されたサービスを通じて CDE を越えて移動できないことを十分に示すために、コンテンツを意識した制御を行っていることを証明できます。

ハイブリッド環境のためのスコーピングガイダンス

お客様の組織がワークロードを AWS クラウドに移行する際、ハイブリッドなインフラストラクチャを運用することがあります。オンプレミスのデータセンターと AWS の両方で実行されているワークロードがあるかもしれません。セグメンテーションコントロールは、このハイブリッドインフラストラクチャとオンプレミス環境と AWS クラウド環境間の通信を考慮する必要があります。

以下のシナリオを考えてみましょう：

1. シナリオ 1:オンプレミスのデータセンター内のリソースにネットワーク接続され AWS でホストされている PCI DSS リソース

2. シナリオ 2: AWS 上のリソースにネットワーク接続されているオンプレミスのデータセンターでホストされている PCI DSS リソース

3. シナリオ 3：オンプレミスのデータセンターと AWS の両方でホストされている PCI DSS リソース

シナリオ 1 および 2 では、非 CDE リソース（ホストされている場所に関係なく）のスコープは、これらの非 CDE リ ソースがスコープ内のシステムと持つ接続のタイプによって決定されます。

非 CDE リソースが CDE リソースに直接接続している場合、これらのオンプレミスまたは AWS リソースは PCI DSS のスコープに含まれ、CDE の一部となります。

- 非 CDE リソースが接続されているとみなされるシステムへの接続を持つ場合、それらのオンプレミスまたは AWS リソースは、セキュリティまたは管理サービスを提供しない限り、PCI DSS の対象外であるとみなされることがあります。
- さらに、それらのスコープ外のシステムが侵害された場合、それらのシステムが CDE をさらに侵害するために使用される方法はないはずです。例えば、AWS Systems Manager は、AWS クラウドとオンプレミスの両方のサーバーとスコープ内の EC2 インスタンスを管理できる。このシナリオでは、オンプレミスサーバーは、アカウントデータとのやり取りや CDE への他の接続がなければ、スコープの外になる可能性があります。

スコープを制限するには、次のガイドラインを考慮します：

- オンプレミスネットワークからのネットワーク接続を、CDE 以外のリソースのみに制限します。
- CDE リソースへの接続が必要な場合は、適切なセキュリティコントロールと設計を実施し、不要なネットワークおよびシステムコンポーネントをスコープ内に取り込む可能性のある推移的ネットワーク接続を防止します。
- スコープの境界が誤って変更されないようにするため、複数のセグメンテーションコントロールを実装します。これらの制御は、オンプレミスのステートフルファイアウォール技術、セキュリティグループ、ネットワークアクセス制御リスト（ネットワーク ACL）を使用して定義および実装されたアクセス制御ルールと、深層防御を実現する IAM 権限によるバックアップを組み合わせて実施できます。

シナリオ 3 では、スコープの決定を簡単にするために、CDE リソースをオンプレミス環境または AWS クラウド環境のいずれかにグループ化する必要があります。このグループ分けがうまくいかない場合は、CDE、接続先、およびセキュリティに影響を与えるシステムがオンプレミスおよび AWS の両方で特定され、これらのシステムへの接続が正しく特定されていることを慎重に検証します。

意思決定フロー – PCI DSS スコープの特定

意思決定フローは、組織における PCI DSS スコープを正しく特定するのに役立ちます。このプロセスは、組織の環境におけるアカウントデータのフローを正しく特定することから始まります。

アカウントデータの流れを正しく特定したら、次のステップとして、環境内のスコープ内のリソースを特定します。

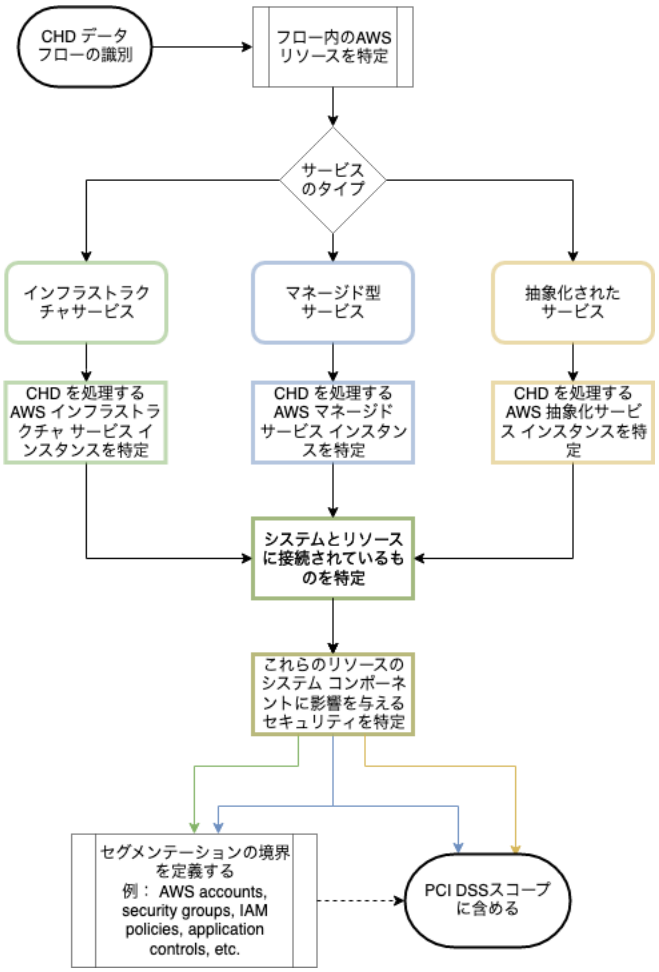

図 2: PCI DSS のスコープと関連するセグメンテーションの境界を正しく特定するための決定フロー

上図は、PCI DSS のスコープを正しく特定するための意思決定フローを示しています。また、フローのさまざまな段階でセグメンテーション境界を正しく定義し、他の AWS リソースをスコープ内のリソースから分離して PCI DSS スコープを縮小するのに役立ちます。

Step 1: CHD および SAD フローの特定

PCI DSS CDE を定義し、セグメンテーションの境界を設計する前に、組織内の CHD および SAD フローを正確に理解する必要があります。アカウントデータのフローを正しく特定するには、組織内のアカウントデータのライフサイクル全体を特定および定義する必要があります。これには、アカウントデータの環境への入力、その後のアカウントデータの送信、処理、及び保存、並びに最終的なアカウントデータの安全な破棄、評価、又は環境からの退出が含まれる。

Step 2: 環境におけるスコープ内のリソースの特定

アカウントデータフローを構成する様々な AWS リソースを特定します。これらのリソースは、アカウントデータを受信、処理、保存、または送信するものです。ISA、外部 QSA、またはその両方が、監査を実施する際に評価をどのサービスに限定すべきかを明確に理解できるように、分析の一部として何が対象で何が対象でないかを定義することが重要です。

Step 3: システムやサービスの分類

このステップでは、システムやリソースをインフラストラクチャ、マネージド、抽象化されたサービスに分類する。これらのリソースのスコープ特定と区分けは、異なる接続に基づいて行われます。インフラストラクチャとマネージド型サービスは主にネットワーク（OSI レイヤー 3〜4）接続を介して互いに通信し、抽象化されたサービスへの通信はサービスの API（OSI レイヤー 7）を介して行われることが多いです。異なる AWS リソースを特定したら、PCI DSS スコープを特定できます。

Step 4: 接続されたリソースの特定

アカウントデータの流れに関係するシステムとサービスを定義した後、それらと接続する、または接続するリソースを特定します。これらの接続は、運用監視、ロギング、構成管理などの目的である可能性があります。EC2 インスタンスのようなインフラストラクチャ・サービスから、S3 バケットのような抽象化されたサービスへの接続など、永続的な接続とオンデマンドの接続の両方を確認する必要があります。これらのリソースへの接続は、CDE の一部です。

Step 5: リソースに影響を与えるセキュリティの特定

システムコンポーネントへの接続を定義した後、セキュリティサービスを提供する、または CDE のセキュリティや構成に影響を与える可能性があるリソースと AWS サービスを特定する必要があります。これには、ログのような特定の PCI DSS 要件を満たすために使用されるシステムコンポーネントや、環境をサポートするために使用される DNS のようなサービスなど、幅広いシステムカテゴリが含まれる可能性があります。多くの種類のシステム コンポーネントがこのカテゴリに分類される可能性があります。詳細については、PCI SSC Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation の 12 ページを参照してください。

Step 6: セグメンテーション境界の設計

スコープ内の AWS リソースを特定したら、他の AWS リソースが適切にセグメント化されて PCI DSS スコープから除外されるように、セグメンテーション境界を設計する必要があります。 AWS の抽象化されたサービスの場合、このセグメント化は主に、アカウントデータのフローを制御するアプリケーションおよび関連するアプリケーションコードによって制御されます。 Amazon EC2 のようなインフラストラクチャ サービス リソースや、Amazon RDS のようなマネージド サービス リソースの場合は、アプリケーション レベルのセグメンテーションとともにネットワーク レベルのセグメンテーションも設計する必要があります。

ソリューションに関する **AWS** クラウドの考慮事項

スコープ内の AWS リソースを特定したら、他の AWS リソースを PCI DSS のスコープから除外するために適切にセグメンテーションされるように、セグメンテーション境界を設計する必要があります。AWS の抽象化されたサービスの場合、このセグメンテーションは主に、アカウントデータのフローを制御するアプリケーショ ンおよび関連するアプリケーションコードによって制御されます。Amazon EC2 のようなインフラストラクチャサービスリソースや、Amazon RDS のようなマネージド型サービスリソースについては、アプリケーションレベルのセグメンテーションとともにネットワークレベルのセグメンテーションも設計する必要があります。

AWS クラウドには、スケーラビリティ、使い捨てのリソース、トレーサビリティ、セキュリティ制御のオートメーション、継続的な検証・テストなどのメリットがあります。また、AWS は、固定費モデルから変動費モデルへの移行が可能であり、必要な個々のサービスに対して、使用する期間だけ支払うことができ、長期間のビジネス契約が不要であるなどの利点があり、様々なビジネス上の利点を提供しています。このような独自の特性により、ほとんどの組織でクラウド導入が有利になります。

AWS の経済的メリットの詳細については、Cloud Economics Center をご覧ください。これらのクラウド特有の特性を利用して決済アプリケーションのインフラを設計することで、これらのメリットを実感できます。AWS のその他の利点については、AWS Well-Architected Framework および Overview of Amazon Web Services.を参照してください。

責任共有モデル

セキュリティとコンプライアンスは、AWS とお客様の間で責任を共有するものです。この共有モデルは、お客様の運用負担を軽減できます： AWS は、ハイパーバイザーのホスト OS や仮想化レイヤーから、サービスが稼働する施設の物理的なセキュリティに至るまで、コンポーネントを運用、管理、制御しています。主に、AWS がクラウドのセキュリティに責任を持ち、お客様はクラウド上のデータの機密性、完全性、可用性を保護する責任を負います。お客様の責任には、情報保護に関するお客様の特定のビジネス要件を満たすことも含まれます。

コンプライアンスの旅に出る前に、責任共有モデルを理解していることを確認してください。共有される責任は、特定の AWS サービスが提供する抽象化レベルに応じて変化します。サービスの抽象度が上がれば上がるほど、お客様の責任は軽減されます。環境で使用されているすべてのサービスを評価し、そのサービスの使用が PCI DSS の全体的なスコープに与える影響を理解します。このアプローチは、コンプライアンス義務を果たすために何をしなければならないかを理解するのに役立ちます。

従来のネットワーク制御の仮想化

AWS が提供する Software-Defined Network（SDN）は、従来のネットワーク構成を模倣していますが、ネットワークの新しい見方や管理方法が必要です。例えば、仮想ローカルエリアネットワーク（VLAN）のような従来のオンプレミスのネットワーク制御は、レイヤー2 ネットワーキングが透過的であるため、AWS では異なる方法で実装されます。

AWS では、Amazon Virtual Private Cloud (Amazon VPC)は AWS クラウドの論理的に分離された区切りを表し、仮想ネットワークでリソースを起動できます。同様の機能または同様のスコープを提供するリソースは、冗長性を提供するために、異なるアベイラビリティゾーン（AZ）をまたぐ複数のサブネットにまたがるのが一般的です。したがって、Amazon VPC とサブネットは、グループ化の構成要素として使用されるべきであり、セグメンテーションのコントロールではありません。

弾力性

アプリケーションに割り当てられたコンピュートやストレージなどの AWS リソースは、需要に応じて水平に拡張できます。AWS Auto Scaling は、お客様のアプリケーションを監視し、可能な限り低いコストで安定した予測可能なパフォーマンスを維持するためにコンピューティング容量を自動的に調整します。このような AWS リソースは短命で、オンプレミスの物理的なリソースのように数ヶ月や数年ではなく、数分や数時間で測定できる可能性があります。Lambda や ELB などの他の AWS マネージド型サービスや機能は、リソース要件に対応するために垂直方向に拡張します。設計するセグメンテーションコントロールは、クラウド環境の伸縮性と一時的な性質に対応できるものでなければなりません。インフラストラクチャが変化し

ても、これらのコントロールが適用され続けるように設計します。そうでない場合、誤ったスコープ定義につながる可能性があります。AWS Auto Scaling グループの一部である EC2 インスタンスなど、リソースが水平方向にスケールする場合、PCI DSS のスコープ内リソースの母集団はそれに合わせてスケールすることに注意が重要です。

抽象化されたサービスと **API** ベースのインフラストラクチャ

AWS の多くのサービスは、マネージド型サービスまたは抽象化されたサービスとして提供されています。これは、AWS がお客様のために基礎となるインフラの多くを管理することを意味し、お客様はそれを制御することができません。多くの抽象化された AWS サービスは、HTTPS 上のサービスの API エンドポイントを通じてのみ通信可能です。AWS サービス API には、認証、認可、データ整合性などの他の制御に加えて、固有のネットワークセグメンテーション制御が含まれています。これは、許可されたエンティティからのデータのみが、呼び出し側システムとサービスの間で交換されることを検証します。このように構成されている場合、これらのサービスは、アクセス制御された API を介して自分自身と他の AWS サービス間で通信します。この構成は、サービスの一部として提供され、ファイアウォールによって提供されるレイヤー3-4 のネットワークセキュリティ制御を満たしています。AWS の抽象化されたサービスを使用して、環境において、システムに接続され、セキュリティに影響を与えるコンポーネントを減らす必要があります。例えば、オープンな TCP/IP 接続を維持するエージェントではなく、暗号化された API コールを行ってログデータを転送する Amazon CloudWatch のようなログ統合サービスを選択します。

アカウントデータフローのために Lambda や ELB のような抽象化されたサービスを構成する場合、TLS バージョン 1.2 以降を使用する HTTPS のような安全な接続も構成する必要があります。AWS の抽象化されたサービスインスタンスがアカウントデータを保存、処理、または送信する場合、それが接続するように構成されているリソースも PCI DSS の対象となります。共有責任モデルの責任の一部として、アプリケーションレイヤーベースのセグメンテーションおよびトラフィックフィルタリング制御を設計する必要があります。

オートメーション

オートメーションにより、インフラストラクチャやアプリケーションの変更のほとんどを、手動で介入することなく実施できます。これにより、俊敏性が得られ、変更管理プロセスが分散化され、導入プロセスが迅速化されます。また、セグメンテーション制御を可能な限りオートメーションし、インフラストラクチャとアプリケーションの変更と同時にセグメンテーション制御が適用されるようにする必要があります。このアプローチにより、定義されたスコープの境界が維持されます。オートメーションは、セグメンテーション制御が変更されたときに、制御を再確立したり、変更の原因と結果を分析するためにほぼリアルタイムで誰かに警告したりするなど、適切な修復ステップを実施できるように検出するのにも役立ちます。

クラウド向けセグメンテーションデザイン

このセクションでは、クラウド機能を使用してクラウド サービスを保護し、多層防御を実現するという原則に基づいて設計できるさまざまなセグメンテーション境界について説明します。これらの境界を AWS のさまざまなレイヤーで実現し、それらを相互に組み合わせて、PCI DSS の対象となるシステムを安全で機能的な CDE に必要な最小限に減らすことができます。

AWS アカウントレイヤー

個々の AWS アカウントは、AWS で達成できる最も高いレベルのセグメンテーションを提供します。設計上、アカウント内でプロビジョニングされたリソースは、他のアカウントでプロビジョニングされたリソースから、AWS Organizations の自分の組織内でさえ論理的に分離されています。AWS 環境を設計する場合、PCI DSS ワークロードに分離されたアカウントを使用することがベストプラクティスとなります。スコープ内とスコープ外のリソースをそれぞれのアカウントにセグメント化することで、PCI DSS のスコープを縮小できます。

AWS アカウントは、ID およびアクセス管理の分離境界として機能します。論理的なアカウントレベルの分離は、別々のアカウントにあるリソース間の明示的な通信チャネルを確立することでしか変更できないため、これは偶発的なスコープクリープを防ぐのに役立ちます。また、こ

のアプローチは、スコープ外のアカウントにおけるアーキテクチャとコントロールの変更が、他のアカウントにおけるスコープ内のリソースのセキュリティに悪影響を与えることを許さないため、影響を軽減するのに役立ちます。複数の AWS アカウントを使用することで、さらにいくつかの運用上の利点があります。これには、AWS service quotas（リミット値）やリクエストレート制限（スロットリング）がアカウントごとに割り当てられるため、これらを分散させることができます。詳細は、Benefits of using multiple AWS accounts を参照してください。

次の図は、PCI DSS のスコープとセグメンテーションに関連したマルチアカウントアーキテクチャの案を示しています：

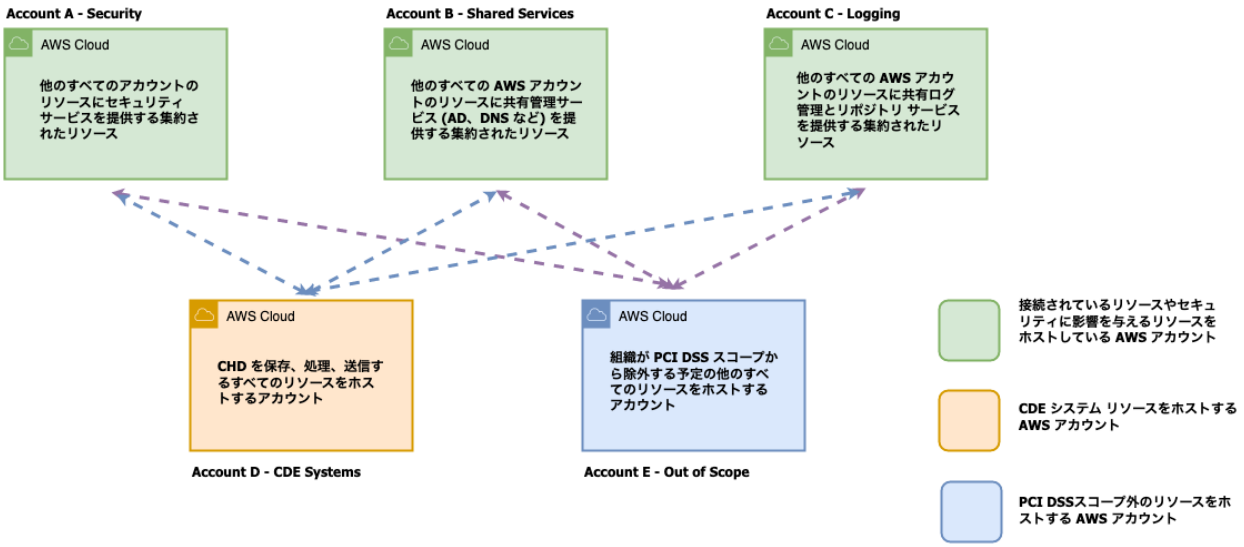

図 3: PCI DSS スコープを制限するマルチアカウント アーキテクチャ

マルチアカウントアーキテクチャのコンポーネント

図 3 は、ホワイトペーパーOrganizing Your AWS Environment Using Multiple Accounts whitepaper で公開されている、AWS のベストプラクティスに従って設計されたマルチアカウントアーキテクチャを示します。同様の機能を提供するリソースは、アカウントにグループ化されます。アカウントはさらに、ルート内の組織単位（OU）の下にグループ化されます。このグループ化により、PCI DSS のスコープが制限されていることを確認しながら、スコープ内お

よびスコープ外の AWS リソース間でセキュリティおよび管理機能を提供するリソースを共有できます。この配置はまた、職務の分離をサポートし、これらのアカウントで操作する必要があるチームに対して最小特権を強制するのに役立ちます。OU は、アカウントを整理する方法を提供し、同様のニーズを持つアカウントに共通の包括的なポリシーを適用することをより簡単にします。組織ポリシーの例として、AWS Organizations のサービスコントロールポリシー (SCP)があります。SCP は、IAM ポリシーが IAM ユーザーやロールなどのアカウント内のエンティティに付与できる権限を制限することで、アカウントがアクセス制御ガイドライン内に留まることを保証するのに役立ちます。たとえば、SCP を使用して、PCI DSS の適用スコープ内のアカウントで構成される OU 内で、PCI DSS 非適合 AWS サービスのプロビジョニングを禁止できます。

現在のシナリオでは、アカウントは、リソースの PCI DSS スコープ、CDE システム、接続されセキュリティに影響を与えるシステム、スコープ外のシステムに基づいて、異なる OU でグループ化されています。

推奨される OU とアカウント

このセクションでは、AWS multi-account strategy guidance に基づき、推奨される OU とアカウントの詳細を説明し、PCI DSS のスコープに合わせるためにグループ名を調整します。組織の命名規則に従って OU に名前を付ける必要があります。

図 4: PCI DSS スコープに関する推奨 AWS Organizations OU 論理グループ化

推奨される OU は一般的なユースケースに沿ったものですが、ニーズに最も適した独自の OU 構造を定義できます。このガイダンスは、ほとんどのお客様のニーズに合致するはずですが、万能ではありません。お客様の要件によっては、推奨される OU をすべて設定する必要がない場合もあります。

PCI DSS のスコープに関する推奨 OU カテゴリは次のとおりです：

- PCI - これは、CDE をホストするアカウント、ワークロード、およびアカウントデータを直接保存、処理、または送信するリソース、およびシステムに接続されていると分類されるリソースをグループ化する OU カテゴリです。この OU は通常、最上位のワークロード OU の下にある子 OU である。
- セキュリティに影響を与える - これは、CDE を含む AWS 環境全体のセキュリティとサポートを助けるために、共通のセキュリティと共有サービス機能を提供するアカウントをグループ化した OU のカテゴリである。
- Out-of-scope - PCI DSS の対象外とみなされるリソースやワークロードをホストするアカウントをグループ化した、その他の OU のカテゴリです。

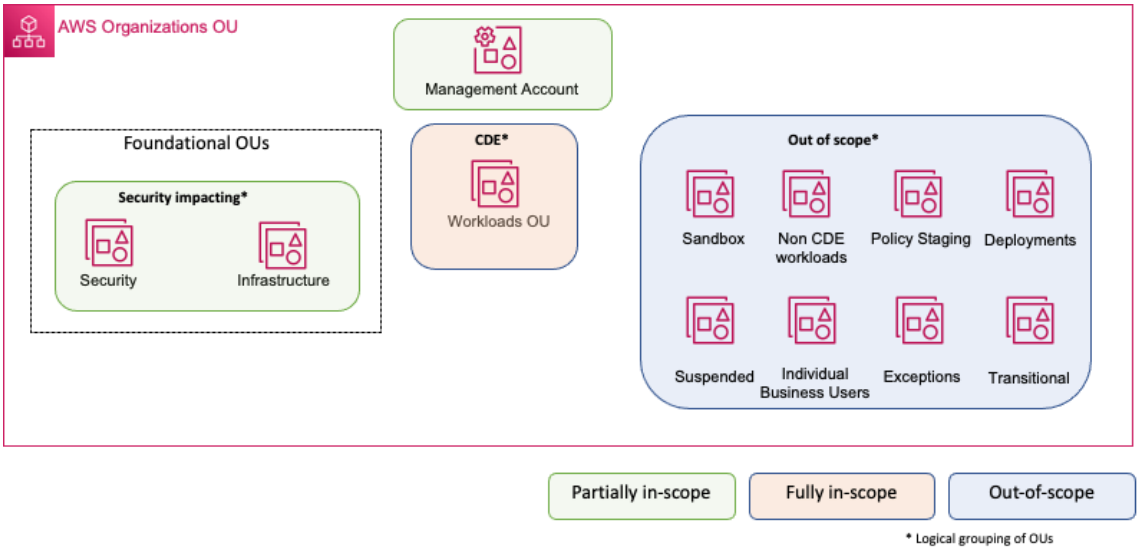

図 4: PCI DSS スコープに関する推奨される AWS Organizations OU 論理グループ

管理アカウント

AWS Organizations は、複数の AWS アカウントを、お客様が作成し一元管理する組織に統合するために利用できるアカウント管理サービスです。AWS Organizations のアカウント管理とマージされた請求機能を使用することで、ビジネスの予算、セキュリティ、およびコンプライアンスニーズをよりよく満たすことができます。

管理アカウントは、組織と CDE のセキュリティに影響するため、PCI DSS のスコープ内と考える必要があります。管理アカウントは、そのアカウントによってのみ実行できるタスクにのみ使用することをお勧めします。AWS リソースを組織内の他のアカウントに保存し、管理アカウントには入れないようにします。

リソースを他のアカウントに保管する重要な理由の 1 つは、AWS 組織 SCP が管理アカウントのユーザーまたはロールを制限するように動作しないためです。詳細については、Best practices for the management account を参照してください。

ワークロード OU

ワークロード OU は、本番環境と非本番環境の両方を含む、ビジネス固有のワークロードの大部分を収容することを目的としています。この OU は、ビジネスユニットやチーム、ソフトウェア開発ライフサイクル（SDLC）環境（本番または非本番）ごとにグループ化された複数の子 OU で構成されています。CDE をホストするアカウントをグループ化するために、そのような子 OU を 1 つ専用にする必要があります。この子 OU には、接続されたワークロードとリソースを含めることができます。この子 OU には、接続されたワークロードやリソースを含めることができます。この専用 OU は、PCI DSS スコープの焦点となり、SCP を使用して実装される組織ポリシーを定義する必要があります。このレイヤー構造により、この OU の下でホストされるアカウントのセグメンテーション制御を含む PCI DSS 要件を満たすために必要なセキュリティ制御が高度に制御および制限されるようにできます。本書では、簡単のため、OU を PCI OU、PCI ワークロードをホストするアカウントを PCI アプリケーションアカウントと名付けました。しかし、組織の命名規則に従ってこれらを命名できます。

CDE と接続されたリソースを別々の PCI アプリケーションアカウントに分離したり、同じアカウントにまとめたりできます。この設計は、ワークロード要件に基づいて行う必要があります。これらのタイプのリソースを混在させる場合、追加のネットワークおよびアプリケーションレベルのコントロールを使用して、これらのリソースを互いに分離できます。このガイダンスでは、1 つのアカウントのみを使用します。ワークロード設計で必要な場合を除き、PCI DSS のスコープにあるリソースをプロビジョニングするために PCI アプリケーションアカウントのみを使用する必要があります。

これらのアカウントには、以下のリソースタイプのホスティングが含まれます：

- アカウントデータを外部エンティティから、または外部エンティティに受信または送信するコンピュートおよび ネットワークリソース
- アカウントデータを処理するコンピューティングリソース
- アカウントデータを静止状態で保存するリソース
- 前述のリソースへのネットワークまたはアプリケーションレベルの接続を確立するリソース

PCI アプリケーションアカウントは、PCI DSS インフラストラクチャの中でも最も機密性の高い部分です。これらのアカウントのシステム・コンポーネントは、他のリソースをスコープに入れることができます。つまり、これらのアカウントが接続するシステムコンポーネントは、通信に関係する機能またはデータに関係なく、スコープ内と見なされる可能性があります。これらのアカウントとの間の通信は、他のアカウントの必要なリソースのみに、システム管理またはビジネス要件の目的に基づいて、高度に制限（論理およびネットワークアクセスの両方）されるべきです。また、これらのアカウントの変更が、組織のセグメンテーション境界および全体的な PCI DSS スコープに悪影響を与えないようにするための変更管理プロセスも実装する必要があります。AWS Config は、特定の AWS リソースへの変更を追跡することができ、それによって Amazon CloudWatch イベントがアラートを生成し、その後 Lambda 関数がセキュリティコントロールの逸脱を自動的に修復したり、変更管理プロセスを実装するための他のさまざまなステップをオーケストレートできます。

非 PCI ワークロードをホストする他の AWS アカウントは、以下の両方の基準を満たしている場合に限って、PCI の範囲内に含めずに、ワークロード アカウントの下の 1 つ以上の個別の子 OU にグループ化できます。

- CDE システムアカウントのセキュリティに影響を与える
- CDE システムアカウント内のリソースとの接続 (論理アクセスまたはネットワーク) がある

前述したように、AWS アカウントは設計上、論理的な分離を提供するため、これらのアカウントは、非 PCI ワークロードに関連するリソースを PCI ワークロードに関連するリソースから分離します (クロスアカウント ネットワークまたはアプリケーション接続を明示的に導入しない限り)。本書では、これらの OU を非 PCI OU と呼びます。およびアカウントは非 PCI アプリケーション アカウントとして使用されます。 アプリケーション チームが所有し、ワークロードの開発、テスト、ステージングに使用するテスト アカウントをホストする追加の子 OU を持つことができます。 PCI DSS v3.2.1 要件 6.4.5.3 および PCI DSS v4.0 要件 6.5.2 では、運用環境でのシステム コンポーネントに対するすべての変更をテストして、システム セキュリティに悪影響を与えないことを確認することが求められています。ただし、アカウント内のリソースが CDE リソースに接続していないこと、または CDE リソースのセキュリティに影響を与えていないことを確認できる場合は、これらの非 PCI 運用リソースを PCI DSS スコープから安全に除外できます。 以下は、関連付けられたアカウントとその PCI DSS スコープを含むワークロード OU の構造例です。

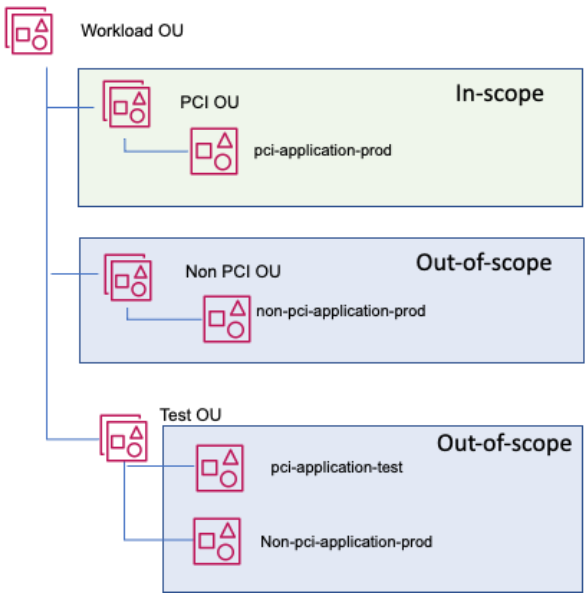

図 5: ワークロード OU に推奨される AWS Organizations メンバー アカウント構造

基礎的な **OU**

これは、CDE リソースにセキュリティと管理サービスを提供するアカウントをグループ化するための 1 つまたは複数の OU とサブ OU の論理的な集合体です。セキュリティ、インフラ、運

用の各チームから構成される中央クラウド環境またはクラウドエンジニアリングチームは、通常、基礎 OU に存在するアカウント、ワークロード、およびデータを所有しています。AWS multi-account recommendation は、セキュリティとインフラストラクチャという 2 つの異なる OU を概説しています。

セキュリティ OU

CDE アカウントにセキュリティサービスを提供するアカウントは、セキュリティ OU にまとめておく必要があります。セキュリティ組織は、子 OU および関連するアカウントとともに、この OU を所有および管理する必要があります。PCI 環境では、セキュリティ OU を使用して、CDE システムにセキュリティサービスを提供するリソースをホストします。テスト用の非本番アカウントをホストするためにサブ OU を持つことをお勧めします。

以下は、セキュリティ OU で推奨されるアカウントです：

- **ログアーカイブ** - このアカウントは、組織内のアカウントから収集されたログデータの監査証跡統合ポイントとして機能します。これには、CDE システムアカウントからのログが含まれます。このアカウントは、PCI DSS v3.2.1 および PCI DSS v4.0 の要件 10 で概説されているいくつかの管理を満たすのに役立つため、ログアーカイブアカウントはスコープ内であると考える必要があります。
- **セキュリティツール** - このアカウントは、広範に適用可能なセキュリティ指向のワークロードをグループ化します。これには、アンチウイルスまたはアンチマルウェアサービス、脆弱性スキャンサービス、侵入検知システム（IDS）および侵入防止システム（IPS）などが含まれる場合があります。このアカウントは、PCI DSS v3.2.1 および PCI DSS v4.0 の要件 2、3、5、6、7、8、および 11 に概説されている多くの管理を満たすのに役立つため、セキュリティツーリングアカウントはスコープ内と考える必要があります。

ワークロード OU で説明したように、アカウント内のリソースが CDE リソースに接続せず、PCI スコープ内リソースのセキュリティに影響を与えないことを確認する限り、テストサブ OU お

よび関連アカウントは PCI DSS スコープ外であるとみなすことができます。次の図は、セキュリティ OU の構造例で、関連するアカウントとその PCI DSS スコープを示しています：

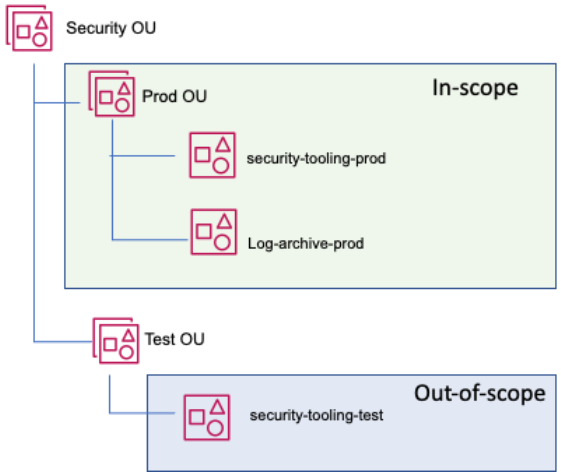

図 6: セキュリティ OU の推奨される AWS Organizations メンバーアカウント構造

インフラストラクチャ **OU**

インフラストラクチャ OU は、もう一つの基礎となる OU であり、共有インフラストラクチャをサポートするためのサービスを含む必要があります。インフラストラクチャチームは、この OU と子 OU、および関連するアカウントを所有および管理する必要があります。PCI 環境では、この OU を使用して、CDE システムの管理機能を提供するリソースをホストします。この OU の一般的な使用例としては、ハイブリッド DNS インフラストラクチャやディレクトリサービスなどの共有サービスがあります。セキュリティ OU と同様に、推奨されるベストプラクティスは、本番用およびテスト用子 OU を通じて、本番用とテスト用リソースを分離することです。

以下は、インフラストラクチャ OU の推奨アカウントです：

- **共有サービス** - このアカウントは、複数のアプリケーションやチームが成果を出すために使用するサービスをサポートします。例えば、AWS Directory Service for Microsoft Active Directory のようなディレクトリサービス、メッセージングサービス、メタデータサービスなどがこのカテゴリーに入ります。このアカウントは、PCI DSS v3.2.1 と PCI DSS v4.0 の両方の要件 2、3、5、6、7、8、11 で規定されている管理を満たすのに役立つため、スコープ内と考える必要があります。

> AWS のマルチアカウントガイダンスホワイトペーパーでは、インフラストラクチャ OU の下に集中型 networking account を推奨しています。これは、アカウント間のトラフィックをルーティングし、インターネットへのトラフィックをイグレスまたはイングレスするネットワークリソースをホストする AWS 上のネットワークの中央ハブになることをお勧めします。PCI DSS のスコープを縮小するには、CDE システムとの間のネットワークトラフィックを集中管理せず、CDE アカウントの一部としてローカルにホストする必要があります。アカウントデータトラフィックを処理するネットワークリソースを一元化すると、PCI DSS のスコープが一元化されたネットワークアカウント内およびそれを経由して拡大する可能性があります。

ワークロード OU について説明したように、アカウント内のリソースが CDE リソースに接続せず、PCI スコープ内リソースのセキュリティに影響を与えないことを確認する限り、テストサブ OU および関連アカウントは PCI DSS のスコープ外であるとみなすことができます。

次の図は、インフラストラクチャ OU の構造例で、関連するアカウントとその PCI DSS スコープを示しています。

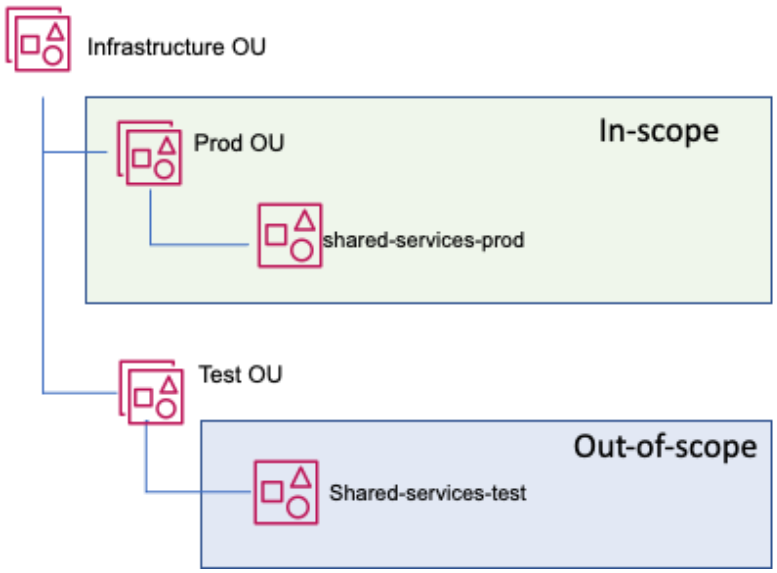

図 7: インフラストラクチャ OU の推奨 AWS 組織メンバーアカウント構造

スコープ外の OU

CDE システムへの接続を必要としない、または CDE システムのサービスを提供しない IT インフラストラクチャをプロビジョニングするには、別の AWS アカウントを使用します。この分離により、これらのアカウントのリソースが、固有の AWS アカウントの分離によって、設計上 PCI スコープ対象システムから適切に分離されることが保証されます。ビジネス機能やニーズに応じて、これらのアカウントを個別の OU にグループ化できます。他のタイプの OU に関する推奨事項については、Organizing Your AWS Environment Using Multiple Accounts を参照してください。

AWS Control Tower

AWS Control Tower はランディングゾーンと呼ばれる安全なマルチアカウント AWS 環境のセットアップとガバナンスを支援するために設計されています。AWS Control Tower は、AWS Organizations を使用してランディングゾーンを作成し、継続的なアカウント管理とガバナンスで実装のベストプラクティスをもたらします。AWS Control Tower を使用すると、アカウントが組織のポリシーに適合していることを確認しながら、数回のクリックで新しいアカウントを

プロビジョニングできます。既存のアカウントを新しい AWS Control Tower 環境に追加することも可能です。AWS Control Tower のオーケストレーションは、AWS 組織の機能を拡張します。組織とアカウントをドリフト、つまりベストプラクティスからの乖離から保護するために、AWS Control Tower はガードレールと呼ばれる予防および検出コントロールを適用します。たとえば、ガードレールを使用して、セキュリティログと必要なクロスアカウントのアクセス許可が作成され、変更されないことを確認するのに役立ちます。

AWS Control Tower を使用する場合、CDE が存在するマルチアカウント環境をオーケストレーションおよび管理するため、PCI DSS のスコープに入ると考える必要があります。

> AWS Control Tower は、デフォルトでセキュリティ OU の下にあるアカウントを Audit Account と命名します。AWS Control Tower のセットアップ中にこのアカウントの名前を変更できます。

詳細については、Best practices for AWS Control Tower administrators.を参照してください。

ネットワークレイヤー(OSI レイヤー 3-4)

セキュリティグループは、暗黙の拒否の原理で動作するステートフルなネットワーク層のトラフィックフィルタリングを提供する Amazon VPC の機能です。セキュリティグループで定義されたトラフィックのみを通過させることができます。セキュリティグループは、ホストベースのファイアウォールに似ており、Amazon EC2 インスタンスのネットワークインターフェイスに関連付けられます。セキュリティグループを使用して、CDE システムを他の接続されたシステムからセグメント化するために必要なポート、およびソースと宛先アドレスに基づいて、ネットワーク通信を制限できます。セキュリティグループは、Amazon VPC の中核となるセグメンテーション境界であり、これらのセキュリティグループを中心にネットワーク層の PCI DSS スコープ戦略を設計する必要があります。AWS の Software-Defined Networking 機能を使用すると、これらのステートフルファイアウォールを EC2 インスタンスに「近づける」ことができ、ネットワークインターフェースに直接アタッチできます。ネットワーク上でリソースを「より近くに」移動させることは、2 つのリソース間のネットワーク経路の長さを短くすることを意味します。これにより、接続されたスコープに影響を与える可能性のあるネットワーク

相互接続の可能性が減り、ネットワークのレイテンシーが減少する可能性があります。つまり、セキュリティグループでトラフィックを制限するように設定している場合、隣接する EC2 インスタンスが PCI DSS に接続され、スコープ内にあると見なされない可能性がある。従来のオンプレミス環境では、ネットワークスタックのさらに上にネットワークの境界が存在するため、アカウントデータを含む単一のホストがそのサブネット全体をスコープに取り込むことになります。また、セキュリティグループを使用してインスタンス間のトラフィックフローを制限することで、PCI DSS v3.2.1 要件 1.2 および 1.3 と PCI DSS v4.0 要件 1.3.1, 1.3.2, 1.4.1 に対応できます。

セキュリティグループを Auto Scaling グループにアタッチすることで、グループのスケールアウトおよびスケールイン時にグループ内のインスタンスに適用され、そこから削除されるようにできます。ピアリングされた Amazon VPC 間で、セキュリティグループを連結して、ハードコードされた IP アドレスやそのレンジを使用する代わりに、あるセキュリティグループがソースまたは宛先として他のセキュリティグループを参照できるようにできます。この設計は、セキュリティグループアーキテクチャのオートメーションに役立ち、クラスレスドメイン間ルーティング（CIDR）レンジの代わりにセキュリティグループメンバーシップを通じてピアリングトラフィックを制御することでスケーラビリティを提供します。

新しく作成されたセキュリティグループのデフォルトの送信構成は、すべての送信トラフィックを許可するため、PCI DSS v3.2.1 要件 1.2.1 または PCI DSS v4.0 要件 1.3.2 に適合しません。セキュリティグループのデフォルトの設定を変更して、必要なトラフィックのみを許可するようにアウトバウンドトラフィックを制限する必要があります。

PCI DSS の全体的なスコープに影響を与えることなく、スコープ外の AWS アカウントにある Amazon EC2 インスタンスと接続された AWS アカウントとの接続を有効にできます。Amazon VPC ピアリング接続は非遷移的であるため、CDE アカウントと、他の接続されたシステムのアカウント間のピアリング接続は、CDE アカウントとアウトオブスコープのアカウント間の明示的なピアリング接続がない限り、それらのアウトオブスコープのアカウントに接続性とスコープを拡張することはありません。

次の図は、セグメンテーションを実現するためにセキュリティグループを使用する方法の例を示しています。このように、セキュリティグループは、VPC やネットワークセグメントなど、他のネットワーク構成に関係なく、主にネットワークのセグメンテーションを定義します。

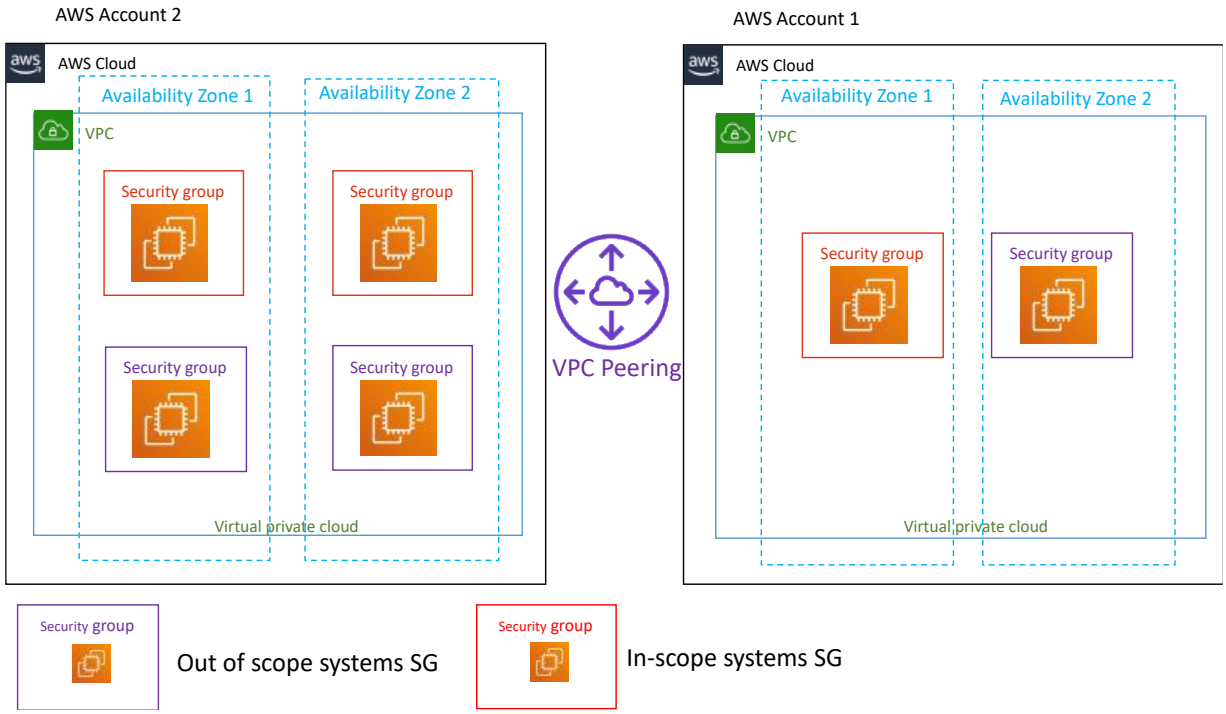

図 8：AWS VPC リファレンスアーキテクチャ

PCI DSS の適用スコープ内リソースのプロビジョニングには、1 つの AWS アカウントを使用する必要があります。ただし、変更が PCI DSS スコープに悪影響を与えないようにするため、CDE のセキュリティグループルールを一元的に制御する必要があります。これを実現するには、複数のアカウントにまたがるセキュリティグループを一元的に管理する機能を提供する AWS Firewall Manager を使用します。AWS Firewall Manager を使用すると、アカウント、リソースタイプ、およびタグによってリソースをグループ化し、スコープ内のセキュリティグループをグループ化して、それら全体でポリシーを適用するのに役立ちます。AWS Firewall Manager は、ポリシーの階層的な適用を支援するため、特定のルールを集中的に適用する機能を保持しながら、アプリケーション固有のルールの作成を委任できます。AWS Firewall Manager は、一元的に適用されたルールが誤って削除されたり、誤操作されたりしないか常に

監視し、ルールが一貫して適用されていることを確認します。AWS Firewall Manager を使用すると、VPC 全体でどのセキュリティグループを許可または不許可にするかを定義するガードレールを設定するポリシーを作成できます。コンプライアンス違反のアカウントやリソースの通知を受け取ったり、AWS Firewall Manager が自動修復によって直接行動するように設定したりできます。AWS Firewall Manager の機能を利用して、未使用、冗長、または過度に許可されたセキュリティグループを半年に一度のファイアウォール規則のレビュープロセスに組み込むことで、PCI DSS v3.2.1 要件 1.1.7 および PCI DSS v4.0 要件 1.2.7 を満たすことができる。

組織のセキュリティ管理で、より高度なファイアウォール機能や、VPC 内でのイングレスとイグレスのトラフィック検査の集中化が必要な場合は、AWS Network Firewall の利用をご検討ください。AWS Network Firewall は、VPC 向けのステートフルでマネージドなネットワークファイアウォールと侵入検知・防止サービスです。AWS Network Firewall を使用すると、VPC の境界でトラフィックをフィルタリングできます。これには、インターネットゲートウェイ、NAT ゲートウェイ、または VPN や AWS Direct Connect 経由のトラフィックのフィルタリングが含まれます。AWS Network Firewall は、ステートフルインスペクションにオープンソースの侵入防御システム（IPS）エンジン Suricata を使用しています。ネットワークセグメンテーションに関する PCI DSS の要件を満たすために、AWS Network Firewall の使用は必須ではありません。しかし、以下のような方法で VPC のトラフィックを監視・保護することで、さらなるセキュリティ効果を発揮します：

- Amazon S3 などの既知の AWS サービスドメインまたは IP アドレスエンドポイントからのトラフィックのみを許可する。
- 既知の不良ドメインのカスタムリストを使用して、お客様のアプリケーションやリソースがアクセスできるドメイン名を制限します。
- VPC に出入りするトラフィックに対してディープパケットインスペクションを実行します。

アプリケーションレイヤー **(OSI** レイヤー **7)**

このレイヤーでは、アカウントデータを扱うアプリケーションは、データの流れを管理し、セグメンテーションの境界を定義する必要があります。AWS は、Lambda、Amazon S3、DynamoDB など、多くの API ベースの抽象化されたサービスを提供しており、組織は、サーバーや Amazon EC2 インスタンスを管理する必要なくビジネス機能を有効にするために利用できます。ほとんどの抽象化されたサービスには、サービスの暗号化された API を介して、HTTPS でのみ通信できます。通信を行うためには、IAM ポリシーを使用してサービスへのアクセスを明示的に許可する必要もあります。AWS の抽象化されたサービスを利用する場合、IAM ポリシーは IAM ポリシーと権限による暗黙の拒否に基づくアプリケーション層の論理セグメンテーション境界となる。抽象化された AWS サービスを使用する場合、PCI DSS のスコープを縮小するためにアプリケーション層のセグメンテーションを設計するのはお客様の責任となります。

抽象化されたサービスのセグメンテーション

抽象化されたサービスは、デザインによってネットワークの分離を実装しています。サブネット内の従来の仮想サーバーが互いに利用可能なデフォルトのネットワーク接続を持つのとは異なり、抽象化されたサービスインスタンスはデフォルトでセグメント化されており、許可されたイベントに基づいてのみ接続を確立します。スコープについては、接続を横断するデータの種類と、データの出入りを許可するための抽象化されたサービスの構成と権限に重点が置かれます。例えば、Lambda 関数は、プロビジョニングしたコードに基づいて通信し、明示的に設定したリソースにのみ接続できます。また、同じ関数は、明示的に許可された IAM の権限と設定を使用してのみ、通信または接続できます。

Amazon API Gateway を使用したセグメンテーション

サーバーレスアプリケーションなどのお客様定義の Web API ベースのサービスでは、Amazon API Gateway を使用して、CDE リソースとサービス、および他の Web ベースのサービス間の接続を仲介できます。API Gateway は、バックエンドのサービスからデータ、ビジネスロジック、または機能にアクセスするためのアプリケーションの「フロントドア」として機能できます。これには、Amazon EC2 上で動作するワークロード、AWS Lambda 上で動作するコード、その他のサポートされる AWS サービス、または Web やモバイルアプリケーションが含まれる可能性があります。CDE システムと AWS でホストされているサービスからの通信を仲介する

CDE 境界インターフェースとして API Gateway を利用できます。ここでは、CDE システムやサービスからの接続はカスタム API インスタンスによって終了され、API インスタンスから非 CDE 先のシステムやサービスへ新しい接続が確立されます。次に、API Gateway を介して CDE システムと通信する AWS CDE 外のリソースは、そのリソースがアカウントデータを扱ったり CDE システムにセキュリティサービスを提供したりしない限り、PCI DSS スコープから除外できます。接続されたシステムとしてスコープに入るのは API Gateway インスタンスのみです。これは PCI DSS 検証済みサービスであるため、サービス自体および PCI DSS 準拠のセグメンテーション境界として機能する能力を維持および検証する必要はありません。お客様は、責任共有モデルの一環として、実装の他の側面について責任を負うことになります。これには、IAM によるアクセス管理や AWS CloudTrail によるロギングの要件が含まれますが、これらに限定されるものではありません。API Gateway は、安全でアクセス制御された Web サービス API メカニズムを提供しますが、アプリケーションは、CDE からの予期せぬアカウントデータの監視を含む、受信データの検証を行います。

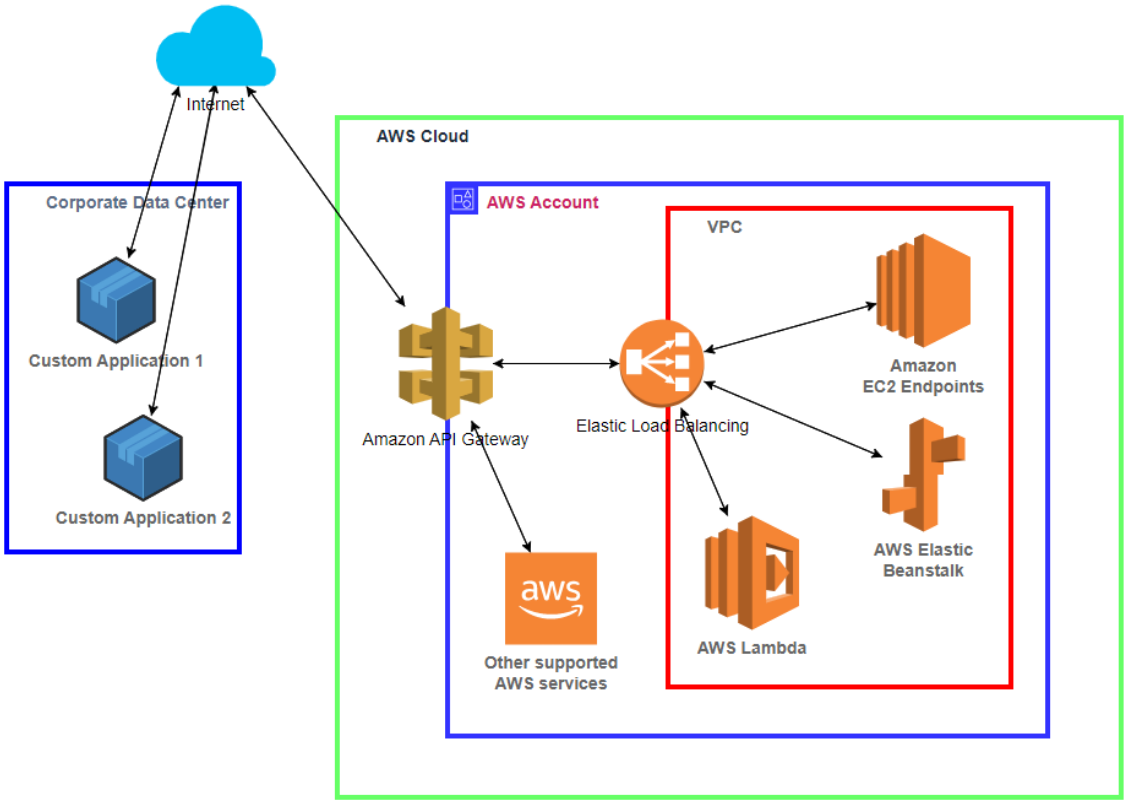

図 9: API Gateway を使用した抽象化されたサービスのセグメンテーション

コンテナワークロードのスコーピングとセグメンテーション

組織はコンテナを使用して、基盤となるインフラストラクチャから独立したアプリケーションを迅速、確実、かつ一貫して展開します。

コンテナを使用すると、アプリケーションのコードと関連する設定や依存関係を含む、自己完結型の標準化されたソフトウェア開発単位を作成できます。コンテナを大規模なビジネスアプリケーションのスケーラブルなビルディングブロックとして使用することで、環境の一貫性を実現し、堅牢なバージョン管理をサポートし、開発者と運用者の効率を高めることができます。Amazon ECS と Amazon EKS は、AWS 上でコンテナ化されたアプリケーションの実行と拡張を可能にする、拡張性と性能の高いコンテナオーケストレーションサービスです。AWS

Fargate は、Amazon ECS と Amazon EKS の両方に対応するサーバーレスコンピュートエンジンで、EC2 インスタンスよりも高い抽象度、スコープ削減、セグメンテーションを提供します。コンテナ化された PCI DSS インスコープ・アプリケーションを実行または設計している場合、それらが適切にスコープおよびセグメント化されていることを確認する必要があります。具体的なガイダンスについては、以下の AWS ホワイトペーパーを参照してください。

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

コンテナのスコープ設定

Amazon ECS tasks と Kubernetes pods on Amazon EKS は、それぞれのサービスのための実行コンテナの論理的なグループです。Amazon ECS と Amazon EKS の両方が、2 つの起動タイプをサポートしています：

- EC2 起動タイプ: このタイプは、あなたが管理する EC2 インスタンスのクラスタ上でコンテナ化されたアプリケーションを実行できます。
- Fargate 起動タイプ: このタイプでは、バックエンドのインフラストラクチャをプロビジョニングおよび管理する必要なく、コンテナ化されたアプリケーションを実行できます。

前述のように、アカウントデータを扱うコンテナ型ワークロードをデプロイするには、PCI OU の下にある PCI アプリケーションアカウントを使用する必要があります。その後、スコープ外の OU の下で別のアカウントを使用して、スコープ内のコンテナから分離されたコンテナ化ワークロードをホストします。スコープ内のワークロードに接続する必要があるが、その通信がアカウントデータに関与しないコンテナ化ワークロードがある場合、PCI OU の下のアカウントまたは他の接続先またはセキュリティに影響を与えるリソースを含む共有サービスアカウントでそれらをホストすることを検討する必要があります。

AWS Cloud Map を使用して、リソースを動的に発見し、そのインベントリを維持し、コンテナ化環境のスコープを定義するのを支援できます。また、AWS App Mesh は、トラフィックフ

ローを維持し、データフロー図を支援することで、スコープを定義するのを助けることができます。App Mesh を使用して、CDE サービス間のトラフィックルーティングを動的に更新できます。

技術的またはビジネス上の制約により、コンテナ型ワークロードのアカウントレベルの分離を使用できない場合は、別の ECS および EKS クラスタと VPC を作成して、スコープ内およびスコープ外のコンテナをグループ化する必要があります。お客様は、同じ AWS アカウントにスコープ内とスコープ外のリソースを持つことで、その PCI アカウントで目的のリソースをうまくスコープ外に保つために、追加の管理および論理的制約が課されることを認識する必要があります。これらのスコープ外のリソースをサポートするプロセスおよびリソース自体は、CDE のセキュリティに影響を与える可能性があるため、スコープ内に入る可能性があります。これには、アカウント内の IAM パーミッションに対するアクセスプロビジョニング手順、スコープ内リソースに隣接して存在するスコープ外リソースをプロビジョニングする展開プロセス、スコープ内アカウント内のスコープ外リソースへの変更に対する変更管理手順、およびこれらのサポートリソースの変更検出が含まれるが、これらに限定されない。スコープ内クラスタを含む AWS アカウントは CDE の一部であるため、スコープ外リソースを維持するサポートプロセスが侵害されたとしても、CDE のセキュリティに影響を与えないことを証明する必要があります。これは、QSA と難しい話になるかもしれません。QSA は、あなたの主張を裏付ける証拠を見た上で、あなたのセグメンテーションと分離の努力によって、目的のリソースが十分に対象外として除外されることに同意しなければなりません。

お客様は、EC2 起動タイプを使用するクラスタを別々の VPC に配置し、ネットワーク層でのセグメント化能力を強化する必要があります。これらのクラスタは、PCI DSS のスコープを設定するための最も低い構成となります。 次のセクションで説明するように、さらなるネットワーク制御を使用してクラスタ間の通信を制限し、同じアカウント内のスコープ外のクラスタをセグメント化します。

Amazon ECS クラスターの分割

Amazon ECS クラスタは、タスクまたはサービスの論理的なグループ化です。タスクやサービスは、クラスタに登録されているインフラストラクチャ上で実行されます。Amazon ECS クラスタにセキュリティグループを割り当て、セキュリティグループのルールを設計して、ネット

ワーク通信をスコープ内のシステムのみに制限するか、スコープ外のシステムコンポーネントからクラスタを分離するか、またはその両方を行うことができます。セキュリティグループは、関連するコンテナインスタンスのファイアウォールとして機能し、コンテナインスタンスレベルでインバウンドとアウトバウンドの両方のトラフィックを制御します。Amazon ECS クラスタは、PCI DSS のスコープ境界を定義するために使用できる最も低い構成です。

AWS Fargate タスク起動タイプでは、タスクは最も低いスコープ構成であるため、クラスタの割り当てとスコープを気にする必要はありません。各 Fargate タスクは独自の分離境界を持ち、基礎となるカーネル、CPU リソース、メモリリソース、またはネットワークインターフェースを他のタスクと共有することはありません。スコープ内コンテナを実行しているタスクをグループ化し、awsvpc ネットワークモードとセキュリティグループルールを併用して、スコープ内タスクと接続されているタスクの間の通信を制限する必要があります。

Amazon EKS クラスターの分割

EC2 起動タイプの EKS クラスタは、以下の主要コンポーネントを備えています：

- EKS コントロールプレーン
- コントロールプレーンに登録されている EKS ノード

コントロールプレーンは AWS が管理するアカウントで実行され、Kubernetes API はクラスタに関連付けられた EKS エンドポイントを通じて公開されます。各 EKS クラスタのコントロールプレーンはシングルテナントで一意であり、独自の Amazon EC2 インスタンスセットで実行されます。EKS ノードはお客様のアカウントで実行され、API サーバーエンドポイントとお客様のクラスタ用に作成された証明書ファイルを通じて、お客様のクラスタのコントロールプレーンに接続します。EKS クラスタは VPC 内に作成されます。Amazon VPC Container Network Interface (CNI)プラグインは、ポッドネットワーキングを提供します。

クラスターを作成すると、Amazon EKS は eks-cluster-sg-my-cluster-uniqueID というセキュリティグループを作成します。デフォルトのルールでは、クラスタとノード間ですべてのトラフィックが自由に流れることができ、任意の宛先へのすべてのアウトバウンドトラフィックを許可します。クラスタとノード間のオープンポートを制限する必要がある場合は、デフォルトのルールを削除し、最小限のルールを追加できます。このクラスタセキュリティグループを使用

して、他の EKS クラスタからのネットワークトラフィックをセグメント化します。EKS クラスターは、PCI DSS スコープの境界を定義するための最も低い構成です。

AWS Fargate タスク起動タイプでは、Kubernetes ポッドが最も低いスコープ構成であるため、ポッドクラスタ間でクラスタ割り当てと設計セグメンテーションを実行する必要はありません。Fargate で実行される各ポッドは、独自の分離境界を持ちます。基礎となるカーネル、CPU リソース、メモリリソース、ネットワークインターフェイスを別のポッドと共有することはない。スコープ内コンテナを実行しているポッドをグループ化し、セキュリティグループルールと組み合わせてポッドネットワーキングを使用し、スコープ内と接続されているポッド間の通信を制限する必要があります

スコーピングとセグメンテーションの検証

PCI DSS v3.2.1 要件 11.3.4 および PCI DSS v4.0 要件 11.4.5 では、加盟店では少なくとも年 1 回、サービスプロバイダでは 6 ヶ月ごとに侵入およびセグメンテーションテストを実行する必要があるとしています。このステップは、NIST SP 800-37 に記載されているセキュリティとプライバシーのシステムライフサイクルアプローチの「コントロールの評価と認可」フェーズに対応します。この種のセキュリティテストを実施する前に、AWS Customer Service Policy for Penetration Testing を確認する必要があります。

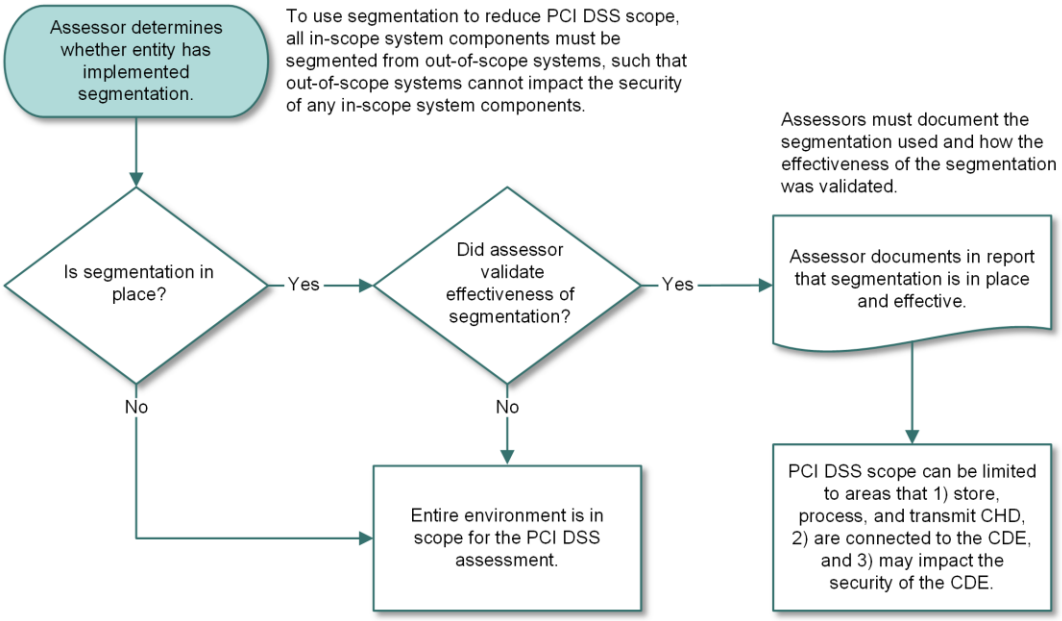

図 10:セグメンテーションと PCI DSS のスコープへの影響

PCI DSS が公表している Information Supplement: Penetration Testing Guidance によると、侵入テストのスコープには、CDE の境界全体と、ネットワークの内外からの重要なシステムコンポーネントのテストが含まれます。これは、外部境界（公衆に向けた攻撃面）と CDE の内部境界（LAN 攻撃面）の両方に適用されます。また、CDE への VPN 接続など、リモートアクセスベクターも含める必要があります。セグメンテーションテストの前または一部として、Amazon VPC 機能の Reachability Analyzer を使用して、ネットワーク構成と接続の意図を検証し、妥当性を確認できます。

インフラストラクチャ、マネージド型、または抽象化されたサービスのいずれを使用する場合でも、セグメンテーション境界のテストと検証を何らかの形で実行する必要があります。従来のセグメンテーション方法は、CDE の内外の 2 点間の論理的な接続性チェックで構成されており、インフラストラクチャやマネージド型サービスを使用する環境では必要です。Amazon EC2 インスタンスなどのインフラサービスや Amazon RDS などのマネージド型サービスでは、セキュリティグループがコアなセグメンテーション境界を形成します。

マネージド型サービスの場合、ネットワークインターフェイスが関係しているため、セキュリティグループも同様のセグメンテーション境界を提供します。テストは、スコープ内のリソー

スとスコープ外のリソースの分離を検証することを目的としています。EC2 の起動タイプを持つコンテナクラスタの場合、セグメンテーションテストを実施して、スコープ内クラスタとスコープ外クラスタ間のセグメンテーション、およびスコープ内クラスタとスコープ外クラスタ間の基盤となる EC2 インスタンスの分離を検証する必要があります。AWS は、基礎となるデータプレーンを管理し、基礎となるコンピュートリソースのセキュリティと分離に責任を持つため、Fargate 起動タイプの Amazon ECS および Amazon EKS クラスタは抽象化サービスです。

スコープ内の抽象化されたサービスでは、サービスとの唯一のインタフェースは dynamodb.us-east-2.amazonaws.com のような AWS サービスエンドポイントを経由します。これらのエンドポイントのスキャンと侵入テストは、サービスプロバイダとしての AWS の PCI DSS 評価の対象となります。ただし、IAM ポリシーによってエンドポイントが特定のリソースに制限されていることを検証する必要があります。抽象化されたサービスを使用する場合、アカウントのデータフローとセグメンテーションの境界は、アプリケーションとアプリケーションコードによって制御されます。したがって、アプリケーションテストに重点を置いて、アプリケーション内で設計されたセグメンテーション境界を検証する必要があります。このように焦点を当てることで、アカウントデータフローと PCI DSS のスコープが、アカウントデータフロー図に描かれているように、アプリケーションによって維持されることが保証されます。

ハイブリッド環境の場合、セグメンテーションテストのソースは、物理的なオンプレミスネットワークのスコープ外のネットワークセグメントであり、ターゲットはスコープ内の EC2 インスタンスであることができます。

プロアクティブセキュリティ制御

PCI DSS で義務付けられている定期的な侵入テストに加え、セグメンテーション制御の不正な変更を防止するためのプロアクティブセキュリティ制御を実施する必要があります。

このセクションでは、実装されたセグメンテーション制御のステータスを監視するための情報を提供します。この監視は、定義された PCI DSS のスコープに意図的または誤って違反しないようにするのに役立ち、PCI DSS v3.2.1 要件 10.8 （サービスプロバイダのみ）および PCI

DSS v4.0 要件 10.7 によって要求されています。違反が発生した場合、それぞれの利害関係者にできるだけ早く通知し、直ちに改善措置を講じることができるよう、予防的および検出的な管理を設計する必要があります。セキュリティ態勢が成熟するにつれて、ほとんどの対応をオートメーションし、ほぼリアルタイムで人手を介さずに逸脱を是正できるようにする必要があります。セグメンテーションの境界を監視する方法としては、次のようなものがあります：

- 予定されている CDE のスコープに対して、定期的にセキュリティグループルールを検証する。
- セキュリティグループとネットワーク構成を変更管理プロセスで管理する
- CDE システムアカウントでセキュリティグループルールの変更を監視する
- CDE システムアカウントへの Amazon VPC ピアリング接続を監視する。
- CDE へのデータの出入りを許可するスコープ内の API または AWS サービスへの構成変更を監視する。
- アカウントデータの漏洩を防ぎ、スコープ境界を検証するために、接続先のシステムでデータ損失防止コントロールを実施する。

ガバナンスやセキュリティに役立つリソースの構成履歴や構成変更通知を提供する AWS Config を使用することで、対応をオートメーションできます。カスタム AWS Config ルールを作成し、セキュリティグループ、Amazon VPC ピアリング接続、Amazon API Gateway API、およびセグメンテーション境界を強制する AWS 上の他のリソースへの変更を監視できます。AWS Config ルールを適切な Lambda レスポンダーにアタッチし、変更が定義された PCI DSS セグメンテーション境界に違反した場合に逸脱を評価し、自動修復を開始します。

AWS CloudTrail を使用して、AWS リソースの構成変更を監視できます。CloudWatch イベントを構成して、Lambda レスポンダーを起動させ、お客様に代わって修復アクションを実行できます。これらは、AWS 上の CDE のセキュリティ制御を設計し、オートメーションするために使用できる様々な AWS サービスのサンプルです。

また、これらのサービスのほとんどと、その他のサードパーティーのツールやアプリケーションを AWS Security Hub と統合できます。Security Hub で PCI DSS standard を有効にして、特

定の PCI DSS 要件を満たすのに役立つ関連するセキュリティコントロールのステータスを監視できます。これには、セグメンテーションコントロールの監視が含まれます。Security Hub を通じて逸脱を検出した場合、Jira などのワークフロー発券システムとの統合や、調査結果と結果を Amazon EventBridge に送信する Security Hub custom actions の定義など、さらなる下流アクションを定義できます。

フィードバックループ

お客様のビジネスは常に変化しており、PCI DSS カード会員データ環境の設計や機能、関連する AWS サービスの選択も変化する可能性があります。ビジネス要件の変更とは別に、AWS インフラストラクチャをより安全、効率的、および管理しやすくするために開発する必要があります。このような絶え間ない変化により、セキュリティとセグメンテーションコントロールの設計プロセスにおいて、フィードバックループを確立することが不可欠となります。前のフェーズや業界の同業者からのフィードバックを収集するチャネルを確立し、このフィードバックを使用して、現在のプロセスをより良く、より安全にする必要があります。フィードバックループとそれに伴う変更により、PCI DSS のスコープを定義するために実装された現在のセグメンテーション制御の再評価が必要になる場合があります。この再評価は、組織の CHD フローの変更に起因することもあります。理由の如何にかかわらず、確立された PCI DSS スコープを検証し、必要に応じてスコープ内のシステムを再分類するプロセスを、最低でも毎年実施する必要があります。

まとめ

この文書では、PCI DSS のスコープを AWS でホストされている CDE リソースの安全な機能に必要なシステム コンポーネントに限定するために、適切なセグメンテーション境界を設計するために採用できるさまざまなアーキテクチャ パターンを紹介しました。すべてクラウドネイティブであるため、弾力的で伸縮性のある性質を持っています。インフラストラクチャをコードとして実装し、組織の既存の継続的インテグレーションと継続的デリバリー（CI/CD）パイ

プラインを使用してオートメーションによってデプロイすることで、コンプライアンスの確保をサポートできます。

PCI DSS で定義された分離と通信制限というセグメンテーションの考え方は、AWS 上のリソースでも変わることはなく、変わるのはそれらのコントロールを実現する方法であり、AWS クラウドに特有のものです。さらに、責任共有モデルの一環として、PCI DSS 検証済みの AWS サービスを使用することは、それらのサービスの使用によって環境の PCI DSS コンプライアンスを達成することを意味するものではありません。お客様は、PCI DSS に準拠した方法でそれらのサービスを使用し、アーキテクトする必要があります。スコープ、設計、および決定にはお客様の組織が責任を負いますが、設計は AWS 上でより俊敏になります。

寄稿者

この文書の寄稿者は次のとおりです。

- Ted Tanner, Principal Assurance Consultant, PCI DSS QSA, AWS Security Assurance Services LLC
- Avik Mukherjee, Sr. Security Consultant, AWS Global Services Security
- Joseph Okonkwo, Sr. Security Architect, AWS Professional Services
- Tomohiro Nakashima, Sr. Security Architect, Amazon Web Services Japan

参考文献

詳細については、次のリソースを参照してください：

- Payment Card Industry Data Security Standard (PCI DSS) 3.2.1 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 Controls using AWS Security Hub on the AWS Cloud
- 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

ドキュメントリビジョン

Date	Description
2019 年 4 月	初版
2023 年 6 月	第 2 版