

口バイダーを訴えることも可能ですが、それでもあなたの会社は検査対象になるでしょう。さらに、クラウドプロバイダーのサーバーが他国に設置されている場合、そのクラウドプロバイダーを使用することについての法的影響も考慮する必要があります。

そしてインサイダーの脅威のリスクについても考慮しなければなりません。この場合のインサイダーとは、サービスプロバイダーで勤務している管理者のことです。職務の分離やM of Nコントロールなどの効果的なセキュリティメカニズムがない場合、そうしたインサイダーがあなたのデータへの特権アクセス入手できる可能性が非常に高くなります。結果としてサービスプロバイダーは、インサイダーによるそうした事態を防いでいることを、あなたの満足ゆくまで立証できなければなりません。また、あなたのデータがその他の未知の仮想サーバーの近くにある、または別の仮想サーバーからあなたのデータに何らかの攻撃が仕掛けられるという、先ほど説明したリスクも存在します。



ピットコイン詐欺のために有名アカウントをハイジャックするというTwitterのハッキングは、インサイダーの脅威に由来するリスクの好例です(scmagazine.com/home/security-news/insider-threats/twitter-hack-is-a-reminder-of-the-dangers-of-unfettered-employee-access)。

あらゆる契約型サービスと同じく、どのようなaaSソリューションであっても、サービスプロバイダーに大きな信頼を置くことになります。そのサービスがあなたの業務にとって重要であればあるほど、その信頼関係に投資するリスクは大きくなります。

クラウドセキュリティ制御

クラウドはオンプレミスネットワークと同じタイプのセキュリティ管理を使用しており、そこにはアイデンティティとアクセス管理(IAM)、(仮想インスタンス向け) エンドポイント保護、データとサービスへのアクセスを統制するリソースポリシー、ホスト間のトラフィックをフィルターするファイアウォール、そして監査機能を提供するログ記録が含まれます。

大半のCSPはこれらのセキュリティ管理をクラウドプラットフォームのネイティブ機能として提供することになります。Googleのファイアウォールサービスはこの種のクラウドネイティブ管理の一例です(cloud.google.com/firewalls)。これらの管理のデブロイと構成は、CSPのWebコンソールを使用するか、またはCLI (Command Line Interface)もしくはAPI (Application Programming Interface)を介したプログラミングによって実行できます。通常、サードパーティのソリューションはクラウド内に仮想インスタンスとしてインストールされます。たとえば、サードパーティの次世代ファイアウォールを実行したいとします。これはアプライアンスとして構成し、クラウドにデプロイすることができます。そのインスタンス経由でトラフィックをルーティングすることで、または何らかのブリッジやミラーリングを用いることで、このアプライアンスインスタンスがトラフィックを検査し、そのトラフィックにポリシーを適用することができるよう、仮想ネットワークアーキテクチャを定義することができます。一例として、Barracuda次世代ファイアウォール(campus.barracuda.com/product/cloudgenfirewall/doc/79462645/overview)の構成ガイドを検討しましょう。

同じ検討は他のタイプのセキュリティ管理についても行うことができ、その代表例としてデータ損失防止やコンプライアンス管理が挙げられます。そうした使用例向けのクラウドネイティブ管理が存在していない場合もあり、サードパーティのソリューションが満たしている機能上の要件を満たしておらず、また変更管理やスキル開発との関連で、移行があまりに困難である可能性もあります。

アプリケーションセキュリティとIAM

クラウドにおけるアプリケーションセキュリティとは、ソフトウェアの開発プロセスと、アプリケーションの認可された使用を保証することを目的としたアイデンティティとアクセス管理(IAM)機能の両方を指します。

オンプレミスソリューションと同じく、クラウドベースのIAMにより、ユーザーおよびユーザーセキュリティグループの作成と、ロールベースの特権管理が可能になります。

秘密管理

クラウドサービスはリモートアクセスに対して非常に脆弱です。認証管理に失敗すると、悪意のあるアカウントに悪用される可能性が生じます。次に挙げる強力な認証ポリシーを実行してリスクを軽減しなければなりません。

- 日々のログオンアクティビティについて、CSPアカウント用にルートユーザーを使用しない。
- 対話型ログオンを行う際に強力な多要素認証を求める。条件付き認証を使用することで、リスクのあるアカウントのアクティビティを拒否または警告する。
- プリンシパル（ユーザー アカウント、セキュリティ グループ、ロール、サービス）は、CLIやAPIを介してクラウドサービスと対話することができます。こうしたプログラムによるアクセスは、アカウントに秘密鍵を割り当てることで可能になります。プログラムによるアクセスには、秘密鍵しか使用できません（通常のアカウント認証情報では不可能です）。あるアカウント用に秘密鍵が生成された場合、それをただちにホストへ転送し、そのホスト上で安全に保存する必要があります。

クラウドコンピューティングのセキュリティ

クラウドは物理ハードウェアから抽象化されたリソースを、1つまたは複数の仮想化レイヤーを介して提供します。コンピューティングコンポーネントは、特定のワーカーロードの必要に応じてプロセスとシステムメモリ(RAM)のリソースを提供します。そのワーカーロードは、4つのCPUと16GBのメモリで構成された仮想マシンインスタンスの場合もあれば、ある関数を実行して一定時間内に結果を返すようスピンドアップされた、コンテナインスタンスの場合もあります。仮想化レイヤーにより、このタスクに必要なリソースがオンデマンドで利用できるようになります。これは動的リソース割り当てと呼ばれることもあります。この能力がSLAで合意した標準を満たすようにするのは、CSPの責任となります。

コンピューティングコンポーネントには、次に挙げるセキュリティ面の重要な検討事項があります。

コンテナーセキュリティ

コンテナーはプラットフォーム上の多数の共有コンポーネントを使用しますが、そのことはデータ漏洩のリスクを減らすべく慎重に構成しなければならないことを意味します。Dockerなどのコンテナーエンジンにおいて、各コンテナーは個別のネームスペースや制御グループを通じて他のコンテナーから隔離されます(docs.docker.com/engine/security/security/)。ネームスペースは、あるコンテナーが別のコンテナーのプロセスを読み取ったり、そこに書き込んだりするのを防止します。一方制御グループは、DoS型の攻撃においてあるコンテナーが別のコンテナーを機能停止に追い込まないようにするものです。

APIの検査と統合

APIは、コンシューマーがクラウドインフラストラクチャ、プラットフォーム、またはアプリケーションとやり取りする手段です。コンシューマーはダイレクトAPIコールを使用するか、またはCSPが提供するWebコンソールをそのAPIのグラフィカルインターフェイスとして使用します。APIの使用状況を監視することにより、システムが過負荷になりつつある場合に警告が発せられ（可用性の保証）、不正な使用やその試みを検知できます。

- リクエスト数 – この基本的な負荷指標では、1秒または1分あたりのリクエスト数がカウントされます。サービスのタイプに応じて、典型的な使用状況のベースラインを確立し、異常な使用を警告するしきい値を設定することができます。たとえばAPIコールにおける説明できないスパイクが、DDoS攻撃のインジケーターとなり得ます。
- レイテンシー – これは、サービスがAPIコールに応答するのにかかる時間をミリ秒(ms)で表したものです。レイテンシーは特定のサービスについて、またはすべてのサービスをまとめる集計値として計測できます。通常はレイテンシーが高いと、コンピューティングリソースが十分でないことを意味します。とは言え、この原因が本物の負荷であることもあれば、DDoS攻撃であることもあります。

- エラーレート – これはコールの総数に対するエラー数の割合として計測され、通常はカテゴリー項目ごとにエラーのタイプが分類されます。APIが応答しない場合、エラーはシステムの過負荷を表し、また許可/アクセスを拒否するタイプのエラーであれば、セキュリティ上の問題を表します。
- 不正なエンドポイントと疑わしいエンドポイント – APIへの接続は、リモートアクセスと同様の方法で管理できます。接続を行うクライアントエンドポイントはACLを用いて制限することができます、またエンドポイントのIPアドレスをモニターして地理的位置を特定することができます。

インスタンス認識

オンプレミスの仮想化と同じく、インスタンス（仮想マシンとコンテナー）を管理することで、文書化されていないインスタンスが起動して管理されないままになるスプロールを避けることが重要です。インスタンスを起動させる権限を制限するだけでなく、ログインと監視を構成して使用状況をトラッキングしなければなりません。

クラウドストレージのセキュリティ

コンピューティングコンポーネントがCPUとシステムメモリのリソースを指す一方、ストレージコンポーネントは永続ストレージ容量のプロビジョニングを意味します。コンピューティングコンポーネントと同じく、クラウドの仮想化レイヤーはそれを支えるハードウェアから抽象化を行い、VMインスタンス用の仮想ハードディスク、Webアプリケーションの静的ファイルを対象とするオブジェクト型ストレージ、またはデータベースサーバー用のブロックストレージなどの必要とされるタイプのストレージを確保します。ストレージのプロファイルには、データベース向けの高速SSDストレージや、それより遅いアーカイブ用のHDDメディアなどのように、さまざまなアプリケーションに応じて異なるパフォーマンス特性があります。主要なパフォーマンス指標として、サポートしている1秒あたりの入出力オペレーション(IOPS)数があります。

アクセス許可とリソースのポリシー

オンプレミスのシステムと同じく、クラウドストレージのリソースは許可されたエンドポイントからのみ読み取りや書き込みが行えるよう構成しなければなりません。クラウドにおいて、リソースポリシーはオブジェクト向けのACLとして機能します。またリソースポリシーの中で、アクセス許可のステートメントは通常、JavaScript Object Notation (JSON)文字列として記述されます。こうしたリソースポリシーの構成を誤ると、幅広く悪用される攻撃ベクトルとなります。一例を挙げると、次のポリシーは「あらゆる」を意味するワイルドカード(*)を使用して、アクション（読み取りと書き込み）とプリンシパル（アカウント）の両方をストレージオブジェクトに割り当てています。この種のポリシーは、最小限の特権という原則を破っており、極めて危険です。

```
"Statement": [ {
    "Action": [
        "*"
    ],
    "Effect": "Allow",
    "Principal": "*",
    "Resource": "arn:aws:s3:::515support-courses-data/*"
} ]
```

暗号化

クラウドストレージの暗号化は、フルディスク暗号化(FDE)というオンプレミスの概念と等しいものです。その目的は、CSPのストレージシステムに対するインサイダーまたは侵入者による攻撃

を通じた、データ喪失のリスクを最小限にすることです。各ストレージユニットはAES鍵を用いて暗号化されます。脅威アクターがデータセンターへ物理的にアクセスし、ディスクをコピーするか取り外しても、そのディスクのデータを読み取ることはできません。

データの読み取りや書き込みを行うには、ストレージオブジェクトを使用するVMやコンテナでAESキーが利用可能である必要があります。CSPが管理する鍵を用いる場合、クラウドプロバイダーは、ストレージリソース上で構成されたアクセス制御権限を使用することでこのプロセスを処理し、アクセスが許可されたものかどうかを判断した上で、許可されたものである場合はそのVMないしコンテナーが鍵を使用できるようにします。この鍵はクラウド内部のハードウェアセキュリティモジュール(HSM)に保存されます。HSMと、権限の分離ポリシーによって、インサイダーの脅威から鍵を保護することができます。それとは別に、カスタマー自身が鍵を管理し、安全な配布と保管に全責任を持つこともできます。

また、暗号化は異なるレベルで適用できます。たとえば、アプリケーションはファイルシステムオブジェクトを選択的に暗号化したり、データベースレベルの暗号化を用いてフィールドやレコードを暗号化することができます。カスタマーからクラウドの間であろうと、クラウド内のVM/コンテナ間であろうと、全てのネットワークにおいてHTTPSやIPSecなどの暗号化されたプロトコルを使用する必要があります。

高可用性

クラウドの利点の1つに、コンポーネント、サーバー、ローカルネットワーク、サイト、データセンター、ワイドエリアネットワークなどさまざまなレベルで、障害に対する耐性が高いサービスを提供し得ることが挙げられます。CSPは仮想化レイヤーを用いることで、コンピューティング、ストレージ、ネットワークのプロビジョニングが、SLAで定められた可用性の基準を満たしていることを保証します。ストレージのパフォーマンスティアという観点から言えば、高可用性(HA)とは99.99%以上の稼働時間を保証する形でストレージがプロビジョニングされることを指します。オンプレミスアーキテクチャと同じく、CSPは冗長性を利用することで、ストレージリソースのプールが複数のディスクコントローラーとストレージデバイスを使用できるようにしています。データはプール間またはグループ間でレプリケーション（複製）され、各プールは別々のハードウェアリソースによってサポートされています。

レプリケーション

データレプリケーションにより、企業は最も効果的に使用できる場所へデータをコピーすることができます。クラウドは中央ストレージエリアとして使用できるため、すべての事業部門でデータが使用可能となります。データレプリケーションには、低レイテンシーのネットワーク接続、セキュリティ、データの完全性が必要です。CSPはいくつかのレベルのデータストレージパフォーマンスを提供しています(cloud.google.com/storage/docs/storage-classes)。ホットストレージとコールドストレージは、データがどのくらい迅速に読み出されるかを指す用語です。ホットストレージの方がデータをより迅速に読み出せますが、データの読み出しが速ければ速いほどコストも高くなります。各アプリケーションにはそれぞれ異なるレプリケーションの要件があります。通常、データベースは低レイテンシーの同期型レプリケーションを必要とします。多くの場合、トランザクションがすべての複製に対して実行されるまで、そのトランザクションは完了したと見なせないからです。データの重要度によっては、データファイルをバックアップストレージに複製するメカニズムに、ここまで高い要件がないこともあります。

ゾーン間の高可用性

CSPは全世界をリージョンに分割しており、各リージョンは互いに独立しています。リージョンはさらにアベイラビリティゾーンに分割されます。アベイラビリティゾーンには、それ自体の電源、冷却装置、ネットワーク接続を有する独立したデータセンターがあります。低レイテンシーのサービスをカスタマーに提供できるよう、データ、サービス、VMインスタンスをどのリージョンでホストするかを選択できます。複数のゾーンとリージョンでリソースをプロビジョニングすることにより、パフォーマンスを改善するとともに、冗長性を向上させることもできますが、十分な水準のレプリケーション能力が必要になります。

結果として、CSPはレプリケーションについていくつかレベルを提供しており、それぞれのレベルは異なる高可用性サービスのレベルを表わしています。

- ローカルレプリケーション – ストレージアカウントを作成したリージョン内の単一のデータセンター内で、データが複製されます。多くの場合、複製は別々の障害ドメインと更新ドメインの中に入ります。
- リージョナルレプリケーション（またはゾーン冗長ストレージ） – 1つまたは2つのリージョン内にある複数のデータセンターにまたがる形でデータが複製されます。これにより、単一のデータセンターが破壊されたりオフラインになったりした場合でも、データとアクセスが保護されます。
- Geo冗長ストレージ(GRS) – 1次リージョンから離れたところにある2次リージョンにデータを複製します。これにより、地域全体に及ぶ稼働停止や災害が発生した場合でも、データが保護されます。

クラウドネットワーキングのセキュリティ

CSPはクラウドの内部で、その基礎にある物理ネットワークを抽象化する仮想化レイヤーを確立します。これにより、CSPは、それぞれのカスタマーアカウントで実行されるネットワークが他のアカウントから分離されたパブリッククラウドを運用することができます。カスタマーが設定するクラウドネットワーキングには、さまざまな環境があります。

- クラウドコンシューマーがクラウドシステムの運用と管理を行うネットワーク。
- クラウド内のVMとコンテナー間で確立される仮想ネットワーク。
- クラウドサービスをインターネット上のゲストまたはカスタマーに公開する仮想ネットワーク。

仮想プライベートクラウド(VPC)

1人1人のカスタマーは、自分のアカウントに紐付ける形で1つまたは複数の**仮想プライベートクラウド(VPC)**を生成することができます。デフォルトでは、VPCは他のCSPアカウントや、同じアカウントで動作しているその他のVPCから切り離されています。このことは、カスタマーBのVPCを通過するトラフィックを、カスタマーAが閲覧できないことを意味します。各VPCのワークロードは他のVPCから切り離されています。VPCの内部において、クラウドコンシューマーはIPv4 CIDRブロックを割り当て、そのブロックの中で1つまたは複数のサブネットを構成できます。その他にも、IPv6 CIDRブロックを割り当てることもできます。



以下の注記はAWSのネットワーク機能に焦点を当てるものです。他のベンダーも同様の機能をサポートしていますが、異なる用語が用いられることがあります。たとえばMicrosoft Azureにおいて、VPCは仮想ネットワークと呼ばれています。

パブリックサブネットとプライベートサブネット

VPC内の各サブネットは、プライベートかパブリックのいずれかです。パブリックサブネットを構成するには、まずインターネットゲートウェイ（仮想ルーター）をVPC構成に紐付ける必要があります。次に、インターネットゲートウェイを各パブリックサブネットのデフォルトルートとして構成しなければなりません。デフォルトルートが構成されていないと、たとえインターネットゲートウェイがVPCに紐付いていても、サブネットはプライベートのままでです。またサブネット内の各インスタンスは、そのクラウドプロファイルのパブリックIPで構成する必要があります。インターネットゲートウェイは1対1のネットワークアドレス変換(NAT)を実行し、インスタンスを出入りするインターネット通信のルーティングを行います。



インスタンスネットワークアダプターは、このパブリックIPアドレスで構成されません。インスタンスのNICはサブネットのIPアドレスで構成されます。パブリックアドレスを用いるのは仮想化管理レイヤーだけです。パブリックIPアドレスはあなた自身のプールから、またはAmazonのElastic IP (docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html)など、CSPが管理するサービスから割り当てることができます。

パブリックにするのが適切でない場合は、サブネットの外部接続をプロビジョニングするその他の方法があります。

- NATゲートウェイ – この機能は、インスタンスをインターネットや他のAWSサービスに接続するのを可能にしますが、インターネットから要求される接続は許可しません。
- VPN – ソフトウェア層で仮想プライベートネットワーク(VPN)を使用して、またはCSPが管理する機能を使用して、VPC間の接続を確立するためのさまざまなオプションが用意されています。

VPCとトランジットゲートウェイ

VPC内のサブネット間でルーティングを構成することができます。このトライフィックは、ホストIPとポートを基にトライフィックを許可またはブロックする、クラウドネイティブACLの対象にすることができます。それとは別に、トライフィックが仮想ファイアウォールインスタンスや、その他のセキュリティアプライアンス経由でルーティングされることもあります。

また、同一アカウント内のVPC間の接続や、異なるアカウントに属するVPCとの接続、VPCとオンプレミスネットワークとの接続も構成することができます。VPCの中でサブネットでなく追加のVPCを構成することにより、インスタンス間のさらなるセグメント化が可能になります。複雑なネットワークでは、パフォーマンス上またはコンプライアンス上の理由により、異なるクラウドアカウントをまたがるさまざまなVPC間でセグメントが分割されていることもあります。

伝統的に、VPNはピアリング関係を用いて相互接続し、VPNゲートウェイを使用してオンプレミスネットワークと接続することができます。こうした1対1のVPCピアリング関係は、特に各VPCがメッシュ構造の中で相互接続しなければならない場合に、管理がすぐさま難しくなることもあります。トランジットゲートウェイは、そうした相互接続を管理するより簡素な方法です。基本的に、トランジットゲートウェイとは、接続された各VPCのサブネット、および接続されたVPNゲートウェイの間のルーティングを処理する仮想ルーターです(aws.amazon.com/transit-gateway)。



Amazonのホワイトペーパーには、マルチVPCインフラストラクチャを構成する際のオプションがさらに詳しく記されています(d1.awsstatic.com/whitepapers/building-a-scalable-and-secure-multi-vpc-aws-network-infrastructure.pdf)。

VPCエンドポイント

VPCエンドポイントとは、他のVPCのインスタンスがAWS内部ネットワークとプライベートIPアドレスだけを使ってアクセスできるよう、サービスを公開する手段です(d1.awsstatic.com/whitepapers/aws-privatelink.pdf)。このことは、トライフィックが決してインターネットに晒されないことを意味します。VPCエンドポイントにはゲートウェイとインターフェイスの2種類があります。

ゲートウェイエンドポイント

ゲートウェイエンドポイントは、VPC内のインスタンスをAWS S3（ストレージ）とDynamoDB（データベース）サービスに接続するために使用します。ゲートウェイエンドポイントは、VPCのルートテーブルにあるサービスへのルートとして構成されます。

インターフェイスエンドポイント

インターフェイスエンドポイントはAWSのPrivateLink機能を利用し、次のようにカスタムサービスへのプライベートアクセスを可能にします。

- カスタムサービスプロバイダーのVPCが、DNSホスト名によりサービスを公開することで構成される。あるいはサービスプロバイダーが、CloudWatch Events/LogsなどのVPCインターフェイスエンドポイントとして有効化される、Amazonのデフォルトサービスである場合もあります。

- VPCエンドポイントインターフェイスがサービスコンシューマーの各VPCサブネット内で構成される。VPCエンドポイントインターフェイスが、サブネット内のプライベートIPアドレスに加え、サービスプロバイダーのDNSホスト名で構成される。
- VPCサブネット内の各インスタンスが、サービスプロバイダーにコンタクトするエンドポイントアドレスを使用するように構成される。

クラウドファイアウォールのセキュリティ

オンプレミスネットワークの場合と同じく、ファイアウォールは出入りするトラフィックを受け入れるか拒否する（または破棄する）かを判断します。ファイアウォールは複数のアカウント、VPC、VPC内のサブネット、そしてサブネット内のインスタンスと連動し、アーキテクチャの設計が求めるセグメント化を実行します。セグメント化が必要とされる理由は多数あり、パフォーマンスやロードバランシングのためにワークロードを分離する、法令や規制を遵守するために、隔離されたセグメント内にデータ処理を留める、さまざまな部門や機能面での要件のために、データのアクセスと処理を区分する、などがその一例です。

フィルタリングの判断は、パケットヘッダーやペイロードのコンテンツを基に、次に挙げるOSIモデルで規定されている、さまざまなレイヤーにおいてなされます。

- ネットワークレイヤー（第3層） – ファイアウォールはIPアドレスまたはアドレス範囲、そしてTCP/UDPポート番号（後者は実際には第4層のヘッダーに含まれていますが、それでもこの機能は基本的な第3層パケットフィルタリングとして常に説明されています）を基に接続を受け入れるか拒否するかを判断します。
- トランスポートレイヤー（第4層） – ファイアウォールは接続ステートを保存し、ルールを用いて確立済みのトラフィックや関連するトラフィックを許可することができます。ファイアウォールは既存の接続のステートテーブルを常に保持しなければならないので、より多くの処理能力（CPUとメモリ）が必要となります。
- アプリケーションレイヤー（第7層） – ファイアウォールはアプリケーションプロトコルのヘッダーとペイロード（HTTPパケットなど）を解析し、その内容を基にフィルタリングの判断を行います。これはさらに多くの処理能力（またはロードバランシング）を必要とし、ファイアウォールがボトルネックとなってネットワークのレイテンシーを増大させることもあります。

クラウド型ファイアウォールを用いてオンプレミスネットワークのセキュリティを実装することができますが、ここでは主に、クラウド内部のトラフィックやクラウドを出入りするトラフィックをフィルターするためのファイアウォールの使用を取り扱います。そうしたファイアウォールは、それぞれの目的に適した、次のようないくつかの方法で実装することができます。

- インスタンス上で動作するソフトウェアとして実装する。この種のホスト型ファイアウォールは、オンプレミスホスト向けに構成するものと同一です。これはステートフルのパケットフィルタリングファイアウォールの場合もあれば、悪意のある攻撃を防止するよう調整されたルールセットを有する、Webアプリケーションファイアウォール(WAF)の場合もあります。欠点として、ソフトウェアがインスタンスリソースを消費するので、さほど効率的でないことが挙げられます。また、多数のインスタンスにまたがるルールセットを管理するのが困難な場合もあります。
- VPCサブネットとインスタンスの間でトラフィックをフィルタリングする、仮想化レイヤーのサービスとして実装する。これは、オンプレミスネットワークのファイアウォールの概念と同一のものです。

ネイティブなクラウドアプリケーション対応のファイアウォールはトランザクションコストを生じさせ、それは通常、デプロイにかかる時間とトラフィック量によって算定されます。これらのコストが理由となって、ネイティブコントロールでなくサードパーティのソリューションが選ばれることもあります。

セキュリティグループ

AWSにおいて、各インスタンスが受け入れるトラフィックを管理する基本的なパケットフィルタリングルールは、セキュリティグループを通じて管理することができます(docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)。セキュリティグループは、第4層でインバウンドとアウトバウンドのステートフルフィルタリングを行います。ステートフルフィルタリングプロトコルとは、新しい接続が受け入れられた場合に、確立済みのトラフィックと関連するトラフィックを許可するということです。

デフォルトのセキュリティグループはすべてのアウトバウンドトラフィックと、そのデフォルトセキュリティグループに紐付いているインスタンスからのすべてのインバウンドトラフィックを許可します。カスタムセキュリティグループでは、インバウンドとアウトバウンドトラフィックを許可するポートとエンドポイントを規定します。セキュリティグループに拒否のルールではなく、許可ルールにマッチしないトラフィックはすべてドロップされます。結果として、ルールを持たないカスタムグループはすべてのネットワークトラフィックをドロップすることになります。同じセキュリティグループに複数のインスタンスを割り当てることもでき、同じサブネット内のインスタンスを異なるセキュリティグループに割り当てることも可能です。また複数のセキュリティグループを同じインスタンスに割り当てることもできます。さらに、セキュリティグループをVPCエンドポイントインターフェイスに割り当てることも可能です。

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTPS	TCP	443	0.0.0.0/0	HTTPS
HTTPS	TCP	443	::/0	HTTPS

AWS EC2で新規インスタンスを起動させる際に、カスタムセキュリティグループを追加。このポリシーにより、単一のIPアドレス（編集済み）からのSSHアクセスと、あらゆるIPアドレスからのHTTPSへのアクセスが可能になります。

大半のクラウドプロバイダーは同様のフィルタリング機能をサポートしていますが、実装の仕方が異なる場合があります。一例を挙げると、Azureにおいて、ネットワークセキュリティグループはネットワークインターフェイスまたはサブネットに適用されることがあります(docs.microsoft.com/en-us/azure/virtual-network/security-overview)。

クラウドアクセスセキュリティプローラー

クラウドアクセスセキュリティプローラー (CASB)は、すべてのタイプのデバイスにおいてユーザーによるクラウドサービスへのアクセスを仲介するよう設計された、エンタープライズ管理ソフ

トウェアです。CASBのベンダーには、現在Symantecの傘下にあるBlue Coat (broadcom.com/products/cyber-security/information-protection/cloud-application-security-cloudsoc)、Forcepoint (forcepoint.com/product/casb-cloud-access-security-broker)、Microsoft Cloud App Security (microsoft.com/en-us/microsoft-365/enterprise-mobility-security/cloud-app-security)、Cisco Cloudlock (cisco.com/c/en/us/products/security/cloudlock/index.html)などがあります。

CASBは、クライアントやその他のネットワークノードによるクラウドサービスの使用状況に関し、優れた可視性をもたらします。CASBの機能として以下のものが挙げられます。

- シングルサインオン認証を可能にし、企業ネットワークからクラウドプロバイダーへのアクセス制御と認可を実行する。
- マルウェア、不正なデバイスによるアクセス、および不適合のデバイスによるアクセスをスキヤンする。
- ユーザーとリソースのアクティビティを監視し監査する。
- 管理されているデバイスから不正なクラウドサービスへのアクセスを防止することで、データの流出を軽減する。

CASBは、原則として以下の3つの方法のいずれかで実装されます。

- フォワードプロキシー クライアントネットワークエッジに位置し、ユーザートラフィックのコンテンツがポリシーに準拠している場合に、そのトラフィックをクラウドネットワークに転送するセキュリティアプライアンスまたはホストです。ただし、ユーザーのデバイスにおいて構成を行うか、エージェントをインストールすることが必要です。このモードでは、プロキシがすべてのトラフィック（承認されたクラウドアプリケーションに向けられていないトラフィックも含む）をリアルタイムで検査します。このモードの問題点として、ユーザーがプロキシを回避して直接接続できることが挙げられます。また、プロキシは負荷分散ソリューションがなければボトルネックとなり、単一障害点となる可能性があるため、パフォーマンスの低下と関連しています。
- リバースプロキシー これはクラウドネットワークエッジに位置し、トラフィックのコンテンツがポリシーに準拠している場合にそのトラフィックをクラウドサービスに送信します。この場合、ユーザーのデバイスを構成する必要はありません。このアプローチは、クラウドアプリケーションがプロキシをサポートしている場合にのみ可能です。
- API (Application Programming Interface) — CASBアプライアンスやホストをクラウドコンシューマーとクラウドサービスの間に設置するのではなく、APIベースのCASBは、クラウドサービスとクラウドコンシューマーの間のブローカー接続を利用します。たとえば、ローカルネットワーク上でユーザー アカウントが無効にされたか、認可が取り消された場合、CASBはこれをクラウドサービスに伝え、APIを用いてそこへのアクセスを無効にします。これは、CASBが要求する機能と、アクセスおよび認可ポリシーが要求する機能をサポートしているAPIに依存します。CASBソリューションでは、セキュリティ管理の目的の違いに合わせて、プロキシモードとAPIモードの両方を使用する可能性が極めて高くなります。

次世代セキュアWebゲートウェイ

多くの場合、企業ネットワークはセキュアWebゲートウェイ(SWG)を活用しています。オングレミスSWGは、プロキシ型のファイアウォール、コンテンツフィルター、かつ侵入検知/防止のシステムであり、インターネットのサイトやサービスへのユーザー アクセスを仲介します。Netskope (netskope.com/products/next-gen-swg)が市場展開している次世代SWGは、SWGの機能にデータ損失防止(DLP)とCASBの機能を組み合わせたものであり、クライアントによるWebサイトやクラウドアプリへのアクセスのために、完全にクラウドホスト型のプラットフォームを提供しています。これは、Gartnerがセキュアアクセスサービスエッジ(SASE)として定義しているアーキテクチャをサポートしています(scmagazine.com/home/opinion/secure-access-service-edge-sase-key-points-for-early-adopters)。

レビュー アク ティビティ： クラウドセキュリティソリューション

次の質問にお答えください。

1. クラウドソリューションプロバイダーを介してデータやシステムをホストする際に行うべき、主要な検討事項を説明してください。
2. 次の記述は正しいですか、誤りですか？ CSPサービス用に登録したアカウントは、ルート特権を持つアカウントではない。
3. APIのレイテンシーを監視し、どのような問題でもただちに修正することを保証するセキュリティ属性はどれですか？
4. クラウドリソースポリシーのアクセス許可ステートメントを記述するのによく用いられるのは、どのフォーマットですか？
5. 次の記述は正しいですか、誤りですか？ カスタマーは1つのアカウントにつき1つのVPCしか作成できない。
6. どのような機能によって、インスタンスに到着するトラフィックのフィルタリングを行えるようになりますか？
7. クラウドアクセスセキュリティブローカー (CASB)とは何ですか？

トピック15C

インフラストラクチャとしての コードの概念として要約する



対象試験範囲

2.2仮想化とクラウドコンピューティングのコンセプトを要約することができる。

仮想化やクラウドの活用と相まって、自動化とサービス統合のための継続的デリバリーモデルという考え方があります。これらのテクノロジーは、アプリケーションサービスをサポートするためにネットワークとホストをプロビジョニングする、「インフラストラクチャとしてのコード」モデルを提供するために一緒に使用することができます。

サービスインテグレーションとマイクロサービス

コンピューターネットワークの初期の時代、アーキテクチャはサーバーマシンと仲介ネットワークシステム（スイッチとルーター）のプロビジョニングに焦点を当てていました。ルーティング、セキュリティ、アドレス割り当て、名前解決、ファイル共有、電子メールなど、モノリシックネットワークアプリケーションを運用するための「箱」をどこに置くかを中心に、アーキテクチャの選択が行われていたのです。仮想化の出現と共に、こうしたアプリケーションのプロビジョニングにおいては、箱を置く場所とその箱を動作させるOSへの依存度が少なくなっています。仮想化は、ビジネスワークフローをプラットフォームの要件に合わせるのでなく、設計アーキテクチャをビジネス上の要件に適合させるのに貢献しています。

サービス指向アーキテクチャ (SOA)

サービス指向アーキテクチャ (SOA)は、ビジネスワークフローへ密接にマッピングされた微細化サービスを生み出すものです。それぞれのサービスは定義済みの入力を受け取り、定義済みの出力を生成します。サービス自体がサブサービスから構成されている場合もあります。サービス機能の主な特徴として、自己完結型であり、その他のサービスの状態に依存せず、明確な入力/出力(I/O)インターフェイスを顕在化させるということが挙げられます。各サービスにはシンプルなインターフェイスがあるので、複雑なモノリシックアプリケーションの場合よりも相互運用がはるかに簡単に行えます。サービスの実装は、異なるプラットフォームや開発言語を使用することができるクライアントサービスの互換性の選択肢を制限するものではありません。サービスと、そのサービスをリクエストするクライアントがこのように独立していることは、疎結合と呼ばれます。

マイクロサービス

マイクロサービス型の開発は、アジャイルのソフトウェアプロジェクト管理や、継続的なデリバリーとデプロイのプロセスと多くの点で類似しています。また、各プログラムやツールは1つのことだけをうまくやるべきだというUnixの哲学もルーツとしています。SOAとマイクロサービスの主な違いは、SOAでは他のサービスからサービスを構築できるという点です。対照的に、各マイクロサービスは独立して開発、テスト、およびデプロイされる必要があります。マイクロサービスは単にゆるく分離しているのではなく、高度に分離していると言われます。

サービスインテグレーションとオーケストレーション

サービスインテグレーションとは、そうした分離したサービスやマイクロサービスの各コンポーネントを協働させ、1つのワークフローを実行する方法を指します。SOAがエンタープライズサービスバスという概念を用いているのに対し、マイクロサービスインテグレーションとクラウドサービス/仮想化/自動化インテグレーションは一般的に、オーケストレーションツールを用いて実装されることが非常に多いです。自動化が單一かつ個別のタスクを容易に繰り返せるようにすることに焦点を当てているのに対し、オーケストレーションは一連の自動化されたタスクを実行します。例えば、ロードバランシングされたクラスターに新しいVMを追加することをオーケストレーションするします。このエンドツーエンドプロセスには、VMをプロビジョニングし構成する、新しいVMをロードバランシングされたクラスターへ追加する、そして新たなクラスター構成においてロードバランシングの荷重配分を再構成する、といったことが含まれるでしょう。それを行うにあたり、オーケストレーションされた各ステップでは、自動化された多数のスクリプト、またはAPIサービスコールが実行されるはずです。

オーケストレーションが正しく機能するには、依存性を考慮に入れつつ、自動化された各ステップが正しい順番で実行されなければなりません。また、途中の各ステップで正しいセキュリティ認証情報を提供し、定義されたタスクを実行する権限とパーミッションがなければなりません。手作業による数十または数百のステップを必要とする複雑なプロセスでも、オーケストレーションによって自動化できます。

クラウドオーケストレーションプラットフォームは、多数の人気あるクラウドプラットフォームやサービスに接続し、それらの運営、管理、オーケストレーションを可能にします。サードパーティによるオーケストレーションプラットフォームを用いることの利点の1つに、ベンダーによる囲い込みから身を守るということが挙げられます。あるクラウドプロバイダーから別のプロバイダーに移りたい場合、またはマルチクラウド環境へ移行したい場合でも、自動化されたワークフローは多くの場合、新しいプラットフォーム用に適合させることができます。この分野における業界のリーダーとして、Chef (chef.io)、Puppet (puppet.com)、Ansible (ansible.com)、およびKubernetes (kubernetes.io)などがあります。

API

SOA型かマイクロサービス型かを問わず、サービスインテグレーション、自動化、オーケストレーションはすべてAPI (Application Programming Interface)に依存しています。サービスAPIは、外部のエンティティがそのサービスとやり取りを行う手段であり、期待されるパラメータでそれを呼び出し、期待される出力を受け取ります。WebアプリケーションAPIを作成するにあたり、以下の2つの一般的な「スタイル」があります。

- SOAP (Simple Object Access Protocol) — XML形式のメッセージングを使用し、認証、トランスポートセキュリティ、非同期メッセージングなどの共通の特性をサポートする多くの拡張機能をWeb Services (WS)標準の形で備えています。
- REST (Representational State Transfer) — SOAPが厳密に規定されたプロトコルであるのに対し、RESTはよりゆるいアーキテクチャのフレームワークであり、RESTful APIとも呼ばれます。SOAPのリクエストが正しくフォーマットされたXMLドキュメントとして送信されなければならない一方、RESTのリクエストはHTTP操作または動詞 (GETやPOSTなど) として送信できます。API内の各リソースないしエンドポイントは名詞として表現されており、単一のURL経由でアクセスしなければなりません。

サーバーレスアーキテクチャ

サーバーレスはサービスデリバリーの現代的な設計様式です。これは現在のWebアプリケーションと強く結びついており、最も有名なものとしてNetflix (aws.amazon.com/solutions/case-studies/netflix-and-aws-lambda)がありますが、企業LANのコンセプトを完全に置き換える製品を用意しているプロバイダーも出現しつつあります。サーバーレスでは、すべてのアーキテクチャはクラウド内でホストされますが、「従来の」仮想プライベートクラウド(VPC)サービスとは異なり、認証、Webアプリケーション、通信などのサービスは、クラウド内にあるVMインスタンス上で動作するアプリケーションとして開発、管理されるわけではありません。その代わ

り、アプリケーションは機能とマイクロサービスとして開発されており、それそれが他の機能と連動してクライアントのリクエストを円滑に遂行します。クライアントが何らかの処理を要求するとき、クラウドはそのコードを実行するためのコンテナを起動し、処理を実行した後、コンテナを破棄します。請求は時間あたりの料金ではなく実行時間に基に行われます。このタイプのサービスプロビジョニングは、Function as a Service (サービスとしての機能：FaaS) とも呼ばれます。FaaS製品にはAWS Lambda (aws.amazon.com/lambda)、Google Cloud Functions (cloud.google.com/functions)、およびMicrosoft Azure Functions (azure.microsoft.com/services/functions)などがあります。

サーバーレスパラダイムによって、物理または仮想サーバーインスタンスを管理する必要がなくなり、ソフトウェアとパッチ、管理特権、またはファイルシステムのセキュリティ監視のために管理が行われることはできません。冗長性やロードバランシングのために複数のサーバーをプロビジョニングする必要もありません。処理のすべてがクラウド内で行われるので、企業ネットワークのプロビジョニングに重点が置かれることもほとんどありません。この基礎的なアーキテクチャは、サービスプロバイダーによって管理されます。主なネットワークセキュリティ業務として、悪意あるアカウントが正当なユーザーになりますまで、サービスにアクセスするクライアントを侵害することが無いことを保証する、というものがあります。これは特に、サービスを支えるアプリケーションコードの更新に用いられる開発者のアカウントとデバイスにおいて、重要な検討事項となります。これらのワークステーションは完全にロックし、開発に必要なアプリケーションやWebコード以外は動作させてはいけません。

またサーバレスにはかなりのリスクがあります。新しいパラダイムということもあります、特にセキュリティとの関連において、使用例やベストプラクティスは成熟したものではありません。さらに、サービスプロバイダーに強く依存せざるを得ないため、サービスのプロビジョニングが失敗した場合の災害復旧オプションも限られています。

サーバーレスアーキテクチャは動作を円滑にするにあたり、イベント駆動型オーケストレーションの概念に大きく依存しています。一例を挙げると、クライアントがアプリケーションに接続する際、複数のサービスが呼び出され、ユーザーとデバイスの認証、デバイスの位置とアドレスのプロパティの識別、セッションの生成、アクションの許可のロード、アプリケーションロジックの使用によるアクションの処理、データベースからの情報の読み取りまたはコミット、そしてトランザクションのログの書き込みを行います。この設計ロジックは、「モノリシック」サーバー型の環境で動作するよう記述されたアプリケーションとは異なります。このことは、既存の企業ソフトウェアを適合させるには、かなりの設計努力が必要であることを意味します。

Infrastructure as Code

クラウドテクノロジーの使用により、手作業で構成を変更したり、パッチをインストールしたりするのではなく、スクリプト化によるプロビジョニングというアプローチを採用するのが合理的です。インフラストラクチャ管理において、自動化とオーケストレーションが手作業による構成を完全に置き換えている場合、そのアプローチを**Infrastructure as Code (インフラストラクチャとしてのコード : IaC)** と呼びます。

IaCの目標の1つに、逸脱(snowflake)システムを除去することが挙げられます。逸脱とは、他のあらゆるものと異なる構成やビルドのことです。プラットフォーム環境における一貫性の欠如（ドリフト）は、パッチがインストールされていないといったセキュリティ上の問題や、細かな構成の違いのためにスクリプトが動作しないといった安定性の問題につながります。手作業による構成を一切拒否することで、IaCは幕等性を保証します。**幕等性**とは、同じパラメータで同じ呼び出しを行なった場合、常に同じ結果を生み出すことを意味します。IaCは単にスクリプトを用いてインスタンスを生成することではない、ということに注意してください。その場で記述されたスクリプトを動作させることも、手作業と同様に環境のドリフトを引き起こすことになります。IaCでは、入念に設計・テストが行われたスクリプトとオーケストレーションのランブックを使用して、一貫したビルドを生み出すことになります。

ソフトウェア定義ネットワーク

IaCは、スクリプトやAPIを介した構成が完全に可能な物理および仮想ネットワークアプライアンスによって部分的に促進されます。数千もの物理・仮想コンピューターやアプライアンスが関わるなど、ネットワークがますます複雑になるのに従い、セキュリティの保証やトラフィックフロー

の管理といったネットワークポリシーの実装も難しくなります。構成すべきデバイスがこれほど多くあるという状況では、まず一歩下がり、ネットワークがどう機能するかに関する抽象化されたモデルを検討する方がいいでしょう。このモデルにおいて、ネットワーク機能は3つの「プレーン」に分けられます。

- 制御プレーン – トラフィックの優先順位付けやセキュリティの確保、どこにスイッチングされるべきかに関する決定を行います。
- データプレーン – トラフィックの実際のスイッチングとルーティングを行い、セキュリティアクセス制御を課します。
- 管理プレーン – トラフィックの状況とネットワークの状態を監視します。

ソフトウェア定義ネットワーク(SDN)アプリケーションを使用することで、制御プレーンにおけるポリシーの決定を定義できます。次にそれらの決定は、APIを用いてネットワークデバイスとやり取りを行うネットワークコントローラーアプリケーションによって、データプレーンに実装されます。SDNアプリケーションとSDNコントローラーの間のインターフェイスは「ノースバウンドAPI」と呼ばれ、コントローラーとアプライアンスの間のインターフェイスは「サウスバウンドAPI」と呼ばれます。互換性のある物理アプライアンスだけでなく、仮想スイッチ、仮想ルーター、仮想ファイアウォールを管理するためにSDNを使用することもできます。汎用のVMとコンテナーを用いた仮想ネットワークの急速なデプロイを支えるのが、**ネットワーク仮想化(NFV)**と呼ばれるアーキテクチャです(redhat.com/en/topics/virtualization/what-is-nfv)。

このアーキテクチャを用いることで、ネットワーク管理者とセキュリティ管理者の仕事が減ると共に、望ましいポリシーを実行するために各アプライアンスを正しい設定で構成する際の複雑さも軽減されます。またネットワーキング、アプライアンス、サーバーのデプロイ（またはプロビジョニング）の完全自動化も可能になります。このために、SDNは最新の自動化とオーケストレーションテクノロジーの重要な一部となっています。

ソフトウェア定義ビジビリティ

SDNがセキュアなネットワーク「ビルト」のソリューションを指すのに対し、**ソフトウェア定義ビジビリティ(SDV)**は評価とインシデント対応の機能をサポートするものです。ビジビリティ(可視化)とは、ネットワークのトラフィックフローに関するデータと、そのネットワークに参加しているすべてのホスト、アプリケーション、ユーザーアカウントの構成と状態に関するデータを、ほぼリアルタイムで収集・集約・レポートすることです。

転送システムから統計情報をを集め、次いで分類スキームをそれらのシステムに適用して、ベースラインレベルから逸脱しているネットワークトラフィックを検知することで、SDVはセキュリティデータの収集プロセスを支援することができます(gigamon.com/content/dam/resource-library/english/white-paper/wp-software-defined-visibility-new-paradigm-for-it.pdf)。これにより、インシデントを示唆し得る異常を検知する能力がさらに安定します。それゆえSDVは、ネットワークフローや、エンドポイントとユーザーアカウントの行動について、従来のアプライアンスでは不可能な高次元の視点をもたらします。SDVはゼロトラストやイースト/ウエストなどの設計に加え(paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture)、セキュリティオーケストレーションと自動化されたレスポンス(SOAR)の実装をサポートしています。

フォグコンピューティングとエッジコンピューティング

ここまで検討してきたクラウドサービスの大半は「ユーザー対面型」のものです。それらはビデオストリーミング、CRMビジネス分析、電子メールと会議、そしてエンドポイント保護分析など、人間のユーザーがやり取りを行うアプリケーションをサポートしています。しかし、非常に多くのクラウドデータ処理が、モノのインターネット(IoT)デバイスとセンサーによって生成されたデータに関して行われており、その数はますます増えています。産業的なプロセスだけでなく家庭の自動化においてさえも、可用性に焦点が当てられています。機密性と完全性が依然として重要な懸念事項である一方、オペレーションナルテクノロジーネットワークにおいてサービスが中断されると、物理的な危険が生じ得ます。その結果、低いレイテンシーでIoTデータを回収し分析する必要性が高まっています。

従来のデータセンターーアーキテクチャは、この必要性を非常によく満たしているとは言えません。センサーの帯域幅は比較的狭く、データネットワークへのWANリンクも高レイテンシーである場合が非常に多いです。センサーが膨大な量のデータを生成しても、優先的に分析する必要があるのは一握りに過ぎません。Ciscoが開発したフォグコンピューティング(cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf)は、IoTセンサーから物理的に近い場所にフォグノード処理リソースを配置することで、これらの必要性に対処するものです。センサーはWi-Fi、ZigBee、または4G/5Gを用いてフォグノードと通信を行い、そのフォグノードはトラフィックの優先順位を決め、注意すべき状況を分析し修復すると共に、残りのデータをデータセンターにバックホールし、そこで保存と優先順位の低い分析が行われます。

エッジコンピューティングはさらに幅広い概念であり、フォグコンピューティングから開発された部分と、それと並行して進化した部分から成っています。現在、フォグコンピューティングはエッジコンピューティングの中で機能するものとして考えられています。またエッジコンピューティングは次の概念を用いています。

- エッジデバイスは、それらの運用に関するデータを収集し、それに依存しているデバイスです。一例を挙げると、HVACシステムの温度計は温度データを収集し、またHVACシステムのコントローラーは電気機械コンポーネントを起動することにより、周辺温度の変化に応じて暖房や空調装置を作動させたり停止させたりします。また自動運転車などのエッジデバイスを考える際には、レイテンシーの問題が浮上します。
- エッジゲートウェイは、エッジデバイスを出入りするデータの事前処理を行うことで、優先順位付けを可能にします。また有線接続もしくは無線接続を行い、ストレージネットワークや処理ネットワークとデータをやり取りします。
- フォグノードはエッジゲートウェイに近接しているデータ処理レイヤーとして組み込むことができ、重要なデータ伝送の優先順位付けを支援します。
- クラウドもしくはデータセンターレイヤーが主たるストレージリソースと処理リソースを提供し、サイト間でデータの配布と集約を行います。

セキュリティの観点から言えば、フォグノードやエッジゲートウェイレイヤーは、サービス拒否攻撃とデータ流出攻撃の両方にとって価値の高い標的となります。



5Gやエッジネットワークの内部でHuaweiの機器を使用することにまつわる論争は、サプライチェーンと信頼できるコンピューティングに関するリスクと懸念を浮き彫りにしています(threatpost.com/huawei-5g-security-implications/152926)。

レビュー アク ティビティ： Infrastructure as Code

次の質問にお答えください。

1. ある企業は専用に開発された顧客管理用のクライアントサーバーアプリケーションを使用しており、VPN経由でリモートサイトからアクセスしています。急速な海外展開のために、システムが何度も機能停止に陥り、ユーザー数の増加や、スマートフォンなどのクライアントデバイスによるアクセス数の増加に対処できないという不満が、従業員から多数寄せられるようになりました。より拡張性の高いソリューションを構築し得るのは、どのタイプのアーキテクチャですか？
2. あなたはChefとPuppetという製品について、よい点と悪い点を要約するよう求められました。これらがサポートするのは、どのタイプの仮想化テクノロジーまたはクラウドコンピューティングテクノロジーですか？
3. 次の記述は正しいですか、誤りですか？サーバーレスとは、コンピューターコードが組み込みシステム上で実行されることである。
4. 特定のシステム上で更新がずっと失敗しているせいで、ある会社のWebサービスがパフォーマンスの問題に見舞われています。この問題に対処し得るのはどのタイプのアーキテクチャですか？
5. SDVとは何ですか？

レッスン15

概要

仮想化とクラウドコンピューティングの概念を要約し、コンピューティング、ストレージ、ネットワーク機能向けにクラウドセキュリティ管理を実装できる必要があります。

セキュアなクラウドソリューションを実装するためのガイドライン

クラウドや仮想化インフラストラクチャをデプロイし、それらの使用を拡大させる際には、次のガイドラインに従ってください。

- 適切なクラウドデプロイモデル（パブリック、ホステッドプライベート、プライベート、コミュニティ、またはハイブリッド）を決定する可用性と機密性の要件を評価する。
- 利用可能な開発リソースと必要とされるカスタム化の度合いに応じて、アプリケーションの要件に最も適したサービスプロビジョニング（ソフトウェア、プラットフォーム、またはインフラストラクチャ）を特定する。
- サービスやビジネスのニーズが、次に挙げる先進的な概念によってよりよくサポートされ得るかどうかを検討します。
 - サーバーの管理でなくワークフローの要件に焦点を当てるマイクロサービス、サーバーレス、オーケストレーション。
 - プラットフォームの自動化プロビジョニングに向けてのIaC、SDN、SDV。
 - 組み込みシステムやIoTネットワークの可用性と低レイテンシーを保証するエッジ/フォグコンピューティング。
- CSPを使用する場合はSLAとセキュリティ責任分担表を作成し、セキュリティ上重要なタスクを誰が実行するかを特定する。クラウドセキュリティデータのレポートと監視が、オンプレミスの監視やインシデント対応と統合されていることを確認する。
- オンプレミスの仮想化またはプライベートデータセンターを使用する場合は、仮想マシンの開発とデプロイ、ハイパー-バイザーセキュリティの保護に向けた堅牢な手順を確保する。
- ネイティブまたはサードパーティのセキュリティ管理を次のように構成して、クラウドサービスを保護します。
 - コンピューティングリソースに関し、ワーカロードの分離と動的リソース割り当てがされていることを保証する。
 - ストレージリソースに関し、ローカルまたはゾーンベースのレプリケーションを通じて高可用性を確保する。
 - ネットワークリソースに関し、仮想化ネットワークを通じてインスタンスを適切なセキュリティゾーンに分離し、リクエストのフィルタリングと認証を行うために、ネイティブまたはベンダーのファイアウォールとセキュリティをプロビジョニングする。
- 多要素認証(MFA)によって保護された開発者アクセス用のセキュアなアカウントをプロビジョニングし、APIおよびSSHキーと、その他の秘密鍵の管理が有効に行われるようとする。

レッスン16

データのプライバシーと保護の概念を説明する

レッスン概要

人が組織にとって最も重要な資産であるとすれば、データはそのすぐ次に重要です。サイバーセキュリティに対する認識と技術の急速な普及は、注目を集めたデータやプライバシーの侵害による莫大な風評被害と経済的コストからもたらされました。それは通常、脅威アクターが欲するデータであり、システム全体が保護するために設定されたデータです。

機密性、完全性、そして可用性という、データの処理とストレージにおけるセキュリティ属性は、管理面、運用面、および技術面の制御を組み合わせることで保証されます。あなたはセキュリティと共に、データの収集と保存を行う際のプライバシー要素を評価し、法令や規制を遵守すべくプロセスをどう形作る必要があるかを特定できなければなりません。

レッスンの目的

このレッスンの内容は、以下のとおりです。

- プライバシーとデータセンシティビティの概念を説明する。
- プライバシーとデータ保護の制御を説明する。

トピック16A

プライバシーとデータセンシティビティの概念を説明する



対象試験範囲

- 2.1 エンタープライズ環境におけるセキュリティコンセプトの重要性を説明することができる。
- 5.3 組織のセキュリティに関するポリシーの重要性について説明することができる。
- 5.5 セキュリティに関するプライバシーおよび機密データの概念を説明することができる。

プライバシーとデータセンシティビティの概念を詳しく理解すれば、データガバナンスチーム全体の中で活動する際に役立つでしょう。データのセキュリティとプライバシーという領域では、コンプライアンスを保証するにあたり、ポリシーと手順が技術的制御と同じくらい重要になります。またこれらのポリシーと手順は、外部のパートナー、サプライヤー、顧客との合意事項の中に明記しなければならない場合もあります。あなたはセキュリティのプロフェッショナルとして、これらのポリシー、手順、合意事項を賢明に選択・適用する必要があります。

プライバシーと要配慮（センシティブ）データの概念

情報資産の価値は、機密性、完全性、可用性（CIAトライアド）という、データのセキュリティ属性が侵害されることで、組織にどの程度の影響が及ぶかという観点から考えることができます。組織内の情報を調査する際は、どの程度秘密に保管する必要があるかだけでなく、そのデータがワークフローの中でどのように使用されているかを判断することも重要です。例えば、公開情報には機密性に関するリスクが存在しません。しかし可用性に関するリスクが、ワークフローに大きな影響を及ぼすこともあります。

データは、CIAの属性が適用される処理および保管システム内に安全に保管されなければなりません。実際には、認証および認可アカウントに読み取りまたは読み取り/書き込みのアクセスを提供し、それ以外のアクセスを（例えば、暗号化によって）拒否するファイルまたはデータベース管理システムを意味することになります。このセキュリティ要件とは別に、データガバナンスを形作る際のプライバシーの影響を考慮する必要があります。

プライバシーとセキュリティ

データのセキュリティは重要ですが、プライバシーもやはり重要な要素です。プライバシーとは、個人データの収集と処理を行う際に発生する、データガバナンス上の要件です。個人データとは、特定可能な個人に関する一切の情報であり、その個人はデータ主体と呼ばれます。データセキュリティ管理が処理システムのCIA属性に焦点を当てているのに対し、プライバシーでは、個人データを特定し、保管、処理、保持が関連規制に準拠していることを保証し、個人データへのアクセスを許可された者のみに制限し、データ対象者が自分に関する情報を確認し、削除する権利を有することを保証する必要があります。

情報ライフサイクル管理

情報ライフサイクルモデルでは、セキュリティポリシーとプライバシーポリシーの策定を支援する個別のステップを特定しています。大半のモデルでは次の一般的な各段階が示されています。