Bonnes pratiques AWS Key Management Service

Avril 2017

Mentions légales

Ce document est fourni à titre informatif uniquement. Il présente l'offre de produits et les pratiques actuelles d'AWS à la date de publication de ce document, des informations qui sont susceptibles d'être modifiées sans préavis. Il incombe aux clients de procéder à leur propre évaluation indépendante des informations contenues dans ce document et chaque client est responsable de son utilisation des produits ou services AWS, chacun étant fourni « en l'état », sans garantie d'aucune sorte, qu'elle soit explicite ou implicite. Ce document n'offre pas de garantie, représentation, engagement contractuel, condition ou assurance de la part d'AWS, de ses sociétés apparentées, fournisseurs ou concédants de licence. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par les contrats AWS. Le présent document ne fait partie d'aucun contrat et ne modifie aucun contrat entre AWS et ses clients.

Table des matières

Résumé

AWS Key Management Service (AWS KMS) est un service géré qui vous permet de vous concentrer sur les besoins en chiffrement de vos applications pendant qu'Amazon Web Services (AWS) gère la disponibilité, la sécurité physique, le contrôle des accès, la logique et la maintenance de l'infrastructure sous-jacente. De plus, AWS KMS vous permet de réaliser un audit de l'utilisation qui est faite de vos clés en fournissant les journaux de tous les appels d'API effectués sur celles-ci afin de vous aider à répondre à vos besoins en matière de réglementation et de conformité.

Les clients veulent savoir comment mettre en oeuvre AWS KMS dans leur environnement. Ce livre blanc explique comment utiliser AWS KMS pour chaque fonctionnalité décrite dans le livre blanc AWS Cloud Adoption Framework (CAF) Security Perspective, y compris les différences entre les différents types de clés principales client, à l'aide des stratégies de clé AWS KMS pour garantir le principe du moindre privilège, auditer l'utilisation des clés et une présentation de cas d'utilisation qui fonctionnent pour protéger les informations sensibles dans AWS.

Introduction

AWS Key Management Service (KMS) est un service géré qui facilite la création et le contrôle de clés de chiffrement utilisées pour chiffrer vos données. AWS KMS utilise des modules de sécurité matériels (HSM) pour assurer la sécurité de vos clés.[1] Vous pouvez utiliser AWS KMS pour protéger vos données dans des services AWS et dans vos applications. Le livre blanc AWS Key Management Service Cryptographic Details décrit la conception et la mise en œuvre des contrôles au sein du service afin de garantir la sécurité et la confidentialité de vos données.[2]

Le livre blanc AWS Cloud Adoption Framework (CAF) facilite la coordination entre les différentes parties des organisations qui migrent vers le cloud computing.[3] Le CAF AWS est divisé en plusieurs domaines d'intérêt, applicables à l'implémentation de systèmes informatiques basés sur le cloud, que nous appelons *perspectives*. Le livre blanc CAF Security perspective organise les principes qui guide la transformation de la sécurité de votre organisation selon cinq capacités de base : Gestion de l'identité et des accès (IAM), contrôle de détection, sécurité des infrastructures, protection des données et réponses aux incidents.[4]

Pour chaque fonctionnalité de la perspective sécurité de l'infrastructure CAF, ce livre blanc fournit des détails sur la façon dont votre entreprise doit utiliser AWS KMS pour protéger les informations sensibles dans un certain nombre de différents cas d'utilisation et sur la façon dont vous pouvez évaluer leur progression :

- **Identity and Access Management (IAM) :** vous permet de créer plusieurs mécanismes de contrôle d'accès et de gérer les autorisations pour chacun d'entre eux.
- **Contrôles de détection :** fournit la capacité de journalisation native et de visibilité dans le service.
- **Sécurité de l'infrastructure :** fournit la capacité d'adapter vos contrôles de sécurité à vos besoins.
- **Protection des données :** fournit la capacité de conserver la visibilité et le contrôle sur les données.
- **Réponse aux incidents :** fournit la capacité de répondre aux incidents, de réduire les dommages causés, et de reprendre les opérations pendant et après ces incidents.

Identity and Access Management (IAM)

La fonctionnalité d'Identity and Access Management fournit des recommandations pour déterminer les contrôles pour la gestion des accès dans AWS KMS afin de sécuriser votre infrastructure en fonction des meilleures pratiques et des stratégies internes.

AWS KMS et stratégies IAM

Vous pouvez utiliser les stratégies AWS Identity and Access Management (IAM) conjointement avec des stratégies de clé pour contrôler l'accès à vos clés principales client (clés CMK) dans AWS KMS. Cette section décrit l'utilisation d'IAM dans le contexte d'AWS KMS. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour accéder à la documentation complète d'IAM, consultez le Guide de l'utilisateur AWS IAM.[5]

Les stratégies attachées à des identités IAM (autrement dit, des utilisateurs, des groupes et des rôles) sont appelées *stratégies basées sur une identité* (ou stratégies *IAM*). Les stratégies attachées à des ressources en dehors d'IAM sont appelées *stratégies basées sur une ressource*. Dans AWS KMS, vous devez attacher des stratégies basées sur les ressources à vos clés principales client (clés CMK). Il s'agit là de *stratégies de clé*. Toutes les clés CMK KMS ont une stratégie de clé, et vous devez l'utiliser pour contrôler l'accès à une clé CMK. Les stratégies IAM par elles-mêmes ne sont pas suffisantes pour autoriser l'accès à une clé CMK, même si vous pouvez les utiliser conjointement avec une stratégie de clé CMK. Pour cela, assurez-vous que la stratégie de clé CMK comprend la déclaration de stratégie qui permet aux politiques IAM.[6]

En utilisant une stratégie IAM basée sur une identité, vous pouvez appliquer le principe du moindre privilège en octroyant un accès granulaire à des appels API KMS au sein d'un compte AWS. N'oubliez pas que les stratégies IAM sont basées sur une politique de refus par défaut, sauf si vous avez explicitement accordé une autorisation à un mandataire pour effectuer une action.

Stratégies de clé

Les stratégies de clé constituent le principal moyen de contrôler l'accès aux clés principales dans AWS KMS. Chaque clé CMK est associée à une stratégie de clé qui définit des autorisations d'utilisation et de gestion de la clé. La stratégie par défaut autorise tous les mandataires que vous définissez, ainsi que l'utilisateur racine du compte, à ajouter des stratégies IAM qui référencent la clé. Nous vous recommandons de modifier la stratégie CMK par défaut en fonction des bonnes pratiques de votre entreprise concernant le principe du moindre privilège. Pour accéder à une ressource chiffrée, le mandataire doit disposer d'autorisations à utiliser la ressource, et à utiliser la clé de chiffrement qui protège la ressource. Si le mandataire ne dispose pas des autorisations nécessaires pour l'une ou l'autre de ces actions, la demande pour utiliser la ressource chiffrée est refusée.

Il est également possible de limiter une clé CMK afin qu'elle ne puisse être utilisée que par des services AWS spécifiques avec une instruction conditionnelle `kms : ViaService instruction` dans le cadre de la stratégie de clé CMK. Pour plus d'informations, consultez le Guide du développeur AWS KMS.[7]

Pour créer et utiliser un volume Amazon Elastic Block Store (EBS), vous avez besoin de l'autorisation d'utiliser Amazon EBS. La stratégie de clé associée à la clé CMK doit comprendre des éléments semblables à ce qui suit :

```
{
   "Sid": "Allow for use of this Key",
   "Effect": "Allow",
   "Principal": {
     "AWS": "arn:aws:iam::111122223333:role/UserRole"
   },
   "Action": [
     "kms:GenerateDataKeyWithoutPlaintext",
     "kms:Decrypt"
   ],
   "Ressource": "*"
 },
 {
   "Sid": "Allow for EC2 Use",
   "Effect": "Allow",
   "Principal": {
     "AWS": "arn:aws:iam::111122223333:role/UserRole"
   },
   "Action": [
     "kms:CreateGrant",
     "kms:ListGrants",
     "kms:RevokeGrant"
   ],
   "Ressource": "*",
   "Condition": {
     "StringEquals": {
       "kms:ViaService": "ec2.us-west-2.amazonaws.com"
     }
   }
 }
```

Dans cette stratégie CMK, la première instruction donne à un mandataire IAM spécifique la possibilité de générer une clé de données et de déchiffrer cette clé de données à partir de CMK lorsque cela est nécessaire. Ces deux API sont nécessaires pour chiffrer le volume EBS lorsque celui-ci est attaché à une instance Amazon Elastic Compute Cloud (EC2).

La deuxième instruction de cette stratégie donne au mandataire IAM spécifié la possibilité de créer, répertorier et révoquer des droits pour Amazon EC2. Les droits sont utilisés pour

déléguer un sous-ensemble d'autorisations à des services AWS ou autres mandataires, afin qu'ils puissent utiliser vos clés en votre nom. Dans le cas présent, la stratégie de condition vérifie explicitement que seul Amazon EC2 peut utiliser les droits. Amazon EC2 les utilise pour rattacher un volume EBS chiffré à une instance si le volume est détaché en raison de coupure prévue ou imprévue. Ces événements sont enregistrés dans AWS CloudTrail lorsque ces coupures sont liées à votre audit.

Lorsque vous développez une stratégie CMK, vous devez garder à l'esprit la façon dont les les déclarations de stratégie sont évaluées au sein d'AWS. Cela signifie que si vous avez activé l'IAM afin de contrôler l'accès à une clé CMK, lorsqu'AWS évalue si une action autorisée doit être acceptée ou refusée, la stratégie CMK est associée à la stratégie IAM. De plus, vous devez vous assurer que l'utilisation et la gestion d'une clé sont limitées aux parties nécessaires.

Moindre privilège/Séparation des responsabilités

Les stratégies de clé spécifient une ressource, action, un effet, un mandataire et des conditions pour accorder l'accès aux clés CMK. Les stratégies de clé vous permettent de transmettre plusieurs autorisations granulaires à des clés CMK pour appliquer le principe du moindre privilège. Par exemple, une application peut effectuer un appel d'API KMS pour chiffrer des données, mais il n'existe pas de cas d'utilisation pour cette même application pour déchiffrer les données. Dans ce cas d'utilisation, une stratégie peut accorder l'accès à l'action *kms: Encrypt* mais pas à *kms: Decrypt* et réduire les possibilités d'exposition. De plus, AWS vous permet de séparer les autorisations d'utilisation à partir des autorisations d'administration associées à la clé. Cela signifie qu'une personne peut avoir la possibilité de manipuler la stratégie de clé, sans avoir toutefois les autorisations nécessaires pour utiliser la clé pour des fonctions cryptographiques.

Étant donné que vos clés CMK sont utilisées pour protéger vos informations sensibles, vous devez vous assurer que les stratégies clés correspondantes suivent un principe de moindre privilège. Cela comprend la garantie que vous n'intégrez **PAS** les autorisations kms:* dans une stratégie IAM. Cette stratégie accorde au mandataire des autorisations administratives et d'utilisation sur toutes les clés CMK auxquelles le mandataire a accès. De la même façon, si vous intégrez des autorisations kms:* pour les mandataires dans votre stratégie de clé, vous lui donnez des autorisations à la fois administratives et d'autorisation sur la clé CMK.

Il est important de garder à l'esprit que les stratégies de refus explicite ont la priorité sur le refus implicite. Lorsque vous utilisez NotPrincipal dans la même déclaration de stratégie que « Effect: Deny », les autorisations spécifiées dans la déclaration de stratégie sont explicitement refusées à tous les mandataires, *à l'exception* de ceux spécifiés. Une stratégie KMS de niveau supérieur peut refuser explicitement l'accès à pratiquement toutes les opérations KMS, à l'exception des rôles qui en ont réellement besoin. Cette technique vous permet d'empêcher les utilisateurs non autorisés de s'accorder l'accès pour KMS.

Partage de clés entre comptes

La délégation d'autorisations à une clé CMK dans AWS KMS peut se produire lorsque vous incluez le mandataire racine d'un compte approuvé dans la stratégie de clé CMK. Le compte approuvé a ainsi la possibilité de déléguer ces autorisations aux utilisateurs et rôles IAM au sein de leur propre compte en utilisant des stratégies IAM. Si cette approche peut simplifier la gestion de la stratégie de clé, elle s'appuie toutefois également sur des comptes approuvés pour veiller à ce que les permissions déléguées soient correctement gérées. L'autre approche consiste à gérer les autorisations de manière explicite pour tous les utilisateurs autorisés uniquement à l'aide de la stratégie de clé KMS, ce qui peut rendre la stratégie de clé complexe et moins facilement gérable. Quelle que soit l'approche adoptée, l'approbation spécifique doit être déclarée clé par clé afin de garantir le respect du principe du moindre privilège.

Droits de clé CMK

Les modifications apportées à la stratégie de clé suivent le même modèle d'autorisations que celui utilisé pour la modification de stratégie partout ailleurs dans AWS. Autrement dit, soit les utilisateurs sont autorisés à modifier la stratégie de clé, soit ils ne le sont pas. Les utilisateurs ayant l'autorisation `PutKeyPolicy` pour une clé CMK peuvent remplacer complètement la stratégie de clé pour une clé CMK par une autre stratégie de clé de leur choix. Vous pouvez utiliser des stratégies de clé pour autoriser d'autres mandataires à accéder à une clé CMK, mais les stratégies de clé sont plus adaptées aux affectations d'autorisations relativement statiques. Pour permettre une gestion des autorisations plus précises, vous pouvez utiliser des droits. Les droits sont utiles pour définir des autorisations temporaires et à portée réduire accordées à d'autres mandataires pour l'utilisation de votre clé CMK en votre nom en l'absence d'un appel direct d'API de votre part.

Il est important de connaître les droits par clé et les droits correspondants à un mandataire par limites de clé lors de la conception d'applications pour lesquelles le contrôler d'accès aux clés repose sur des droits. Assurez-vous qu'un mandataire abandonne un droit lorsqu'il n'est plus utilisé pour éviter d'atteindre ces limites.

Contexte de chiffrement

En plus de limiter les autorisations aux API AWS KMS, AWS KMS vous donne également la possibilité d'ajouter une couche d'authentification supplémentaire pour vos appels d'API KMS avec le contexte de chiffrement. Le contexte de chiffrement est une paire clé-valeur de données supplémentaires que vous voulez associer à des informations protégées par AWS KMS. Elle est alors intégrée dans les données authentifiées supplémentaires (données AAD) du chiffrement authentifié dans les textes chiffrés chiffrées par AWS KMS. Si vous soumettez la valeur du contexte de chiffrement dans l'opération de chiffrement, vous devez la transmettre dans l'opération de déchiffrement correspondante. Vous pouvez utiliser le contexte de chiffrement dans vos stratégies pour appliquer les contrôles les plus rigoureux à vos ressources chiffrées. Le contexte de chiffrement étant consigné dans CloudTrail, vous pouvez obtenir davantage d'informations concernant l'utilisation de vos clés dans une perspective d'audit.

Sachez que le contexte de chiffrement n'est pas chiffré et qu'il est visible dans les journaux CloudTrail. Le contexte de chiffrement ne doit pas être considéré comme des informations sensibles et n'exige pas le secret.

Les services AWS qui utilisent AWS KMS se servent du contexte de chiffrement pour limiter la portée des clés. Par exemple, Amazon EBS envoie l'ID du volume en tant que contexte de chiffrement lors du chiffrement/déchiffrement d'un volume, et lorsque vous prenez un instantané l'ID d'instantané, celui-ci est utilisé comme contexte. Si Amazon EBS n'a pas utilisé ce contexte de chiffrement, une instance EC2 est capable de déchiffrer n'importe quel volume EBS dans le cadre de cette clé CMK spécifique.

Un contexte de chiffrement peut également être utilisé pour les applications personnalisées que vous développez et agit comme un niveau de contrôle supplémentaire en s'assurant que les appels de déchiffrement aboutiront uniquement si le contexte de chiffrement correspond à ce qui a été transmis dans l'appel de chiffrement. Si le contexte de chiffrement d'une application spécifique ne change pas, vous pouvez inclure ce contexte dans la stratégie de clé KMS AWS comme instruction conditionnelle. Par exemple, si vous avez une application qui nécessite la possibilité de chiffrer et de déchiffrer les données, vous pouvez créer une stratégie de clé sur la clé CMK qui permet de s'assurer qu'il fournit des valeurs attendues. Dans la stratégie suivante, elle vérifie que le nom d'application « ExampleApp » et sa version actuelle « 1.0.24 » sont bien les valeurs transmises à AWS KMS lors des appels de chiffrement et déchiffrement. Si plusieurs valeurs sont transmises, l'appel sera refusé et l'action de chiffrement ou déchiffrement ne sera pas exécutée.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Ressource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp",
      "kms:EncryptionContext:Version": "1.0.24"
    }
  }
}
```

Cette utilisation du contexte de chiffrement permettra de garantir avec plus de certitude que seules les parties ou les applications autorisées peuvent accéder et utiliser les clés CMK.

Maintenant, la partie aura besoin des autorisations IAM pour AWS KMS, d'une stratégie CMK qui lui permet d'utiliser la clé selon les réquisitions, et enfin de connaître le contexte de chiffrement des valeurs attendues.

Multi-Factor Authentication

Pour fournir une couche supplémentaire de sécurité sur des actions spécifiques, vous pouvez mettre en œuvre une couche supplémentaire de protection à l'aide du Multi-Factor Authentication (MFA) pour les appels d'API KMS critiques. Certains de ces appels sont `PutKeyPolicy`, `ScheduleKeyDeletion`, `DeleteAlias` et `DeleteImportedKeyMaterial`. Cette action peut s'effectuer par le biais d'une instruction conditionnelle dans la stratégie de clé qui vérifie si et quand un périphérique MFA a été utilisé dans le cadre de l'authentification.

Si une personne tente d'effectuer l'une des actions AWS KMS critiques, la stratégie CMK suivante va valider le fait que leur MFA a été authentifiée dans les dernières 300 secondes, ou 5 minutes, avant d'exécuter l'action.

```
{
  “Sid”: “MFACriticalKMSEvents”,
  “Effect”: “Allow”,
  “Principal”: {
    “AWS”: “arn:aws:iam::111122223333:user/ExampleUser”
  },
  “Action”: [
    “kms:DeleteAlias”,
    “kms:DeleteImportedKeyMaterial”,
    “kms:PutKeyPolicy”,
    “kms:ScheduleKeyDeletion”
  ],
  “Ressource”: “*”,
  “Condition”:{
    “ NumericLessThan “:{“aws: MultiFactorAuthAge”:”300”}
  }
}
```

Contrôles de détection

Les contrôles de détection de capacité garantissent que vous avez correctement configuré AWS KMS pour consigner les informations nécessaires dont vous avez besoin pour obtenir une plus grande visibilité dans votre environnement.

Audit CMK

AWS KMS est intégré à CloudTrail. Pour vérifier l'utilisation de vos clés dans AWS KMS, vous devez activer la journalisation CloudTrail dans votre compte AWS. Cela permet de garantir que tous les appels d'API effectués sur les clés KMS de votre compte AWS sont automatiquement enregistrés dans les fichiers qui sont ensuite livrés dans un compartiment Amazon Simple Storage Service (S3) que vous spécifiez. Les informations collectées par CloudTrail vous permettent de déterminer quelle demande a été envoyée, l'adresse IP source à partir de laquelle la demande a été effectuée, qui a effectué la demande, à quel moment, etc.

AWS KMS est intégré de manière native avec de nombreux autres services AWS pour faciliter la surveillance. Vous pouvez utiliser ces services AWS ou votre suite d'outils de sécurité existante, pour surveiller vos journaux CloudTrail à la recherche d'actions spécifiques telles que `ScheduleKeyDeletion`, `PutKeyPolicy`, `DeleteAlias`, `DisableKey`, `DeleteImportedKeyMaterial` sur votre clé KMS. En outre, AWS KMS émet des Amazon CloudWatch Events lorsque votre clé CMK est en rotation, supprimée et que des clés importées dans votre clé CMK arrivent à expiration.

Validation de l'utilisation d'une clé CMK

En plus de la capture des données d'audit concernant la gestion et l'utilisation de clé, vous devez vous assurer que les données que vous examinez sont conformes aux bonnes pratiques et stratégies que vous avez établies. Une méthode consiste à contrôler et vérifier en permanence les journaux CloudTrail au fur et à mesure qu'ils arrivent. Une autre méthode consiste à appliquer des règles AWS Config. En utilisant les règles AWS Config, vous pouvez vous assurer que la configuration de la plupart des services AWS sont configurés de façon appropriée. Par exemple, avec des volumes EBS, vous pouvez utiliser la règle AWS Config `ENCRYPTED_VOLUMES` pour valider le fait que les volumes EBS attachés sont chiffrés.

Balises de clé

Une balise peut être associée à une clé CMK pour différents objectifs. L'utilisation la plus courante consiste à établir une corrélation entre une clé CMK spécifique et une catégorie métier (par exemple, nom d'application, centre de coûts ou propriétaire). Les balises peuvent ensuite être utilisées pour vérifier que la bonne clé CMK est appliquée à une action donnée. Par exemple, dans les journaux de CloudTrail, pour une action KMS, vous pouvez vérifier que la clé CMK utilisée appartient à la même catégorie métier que la ressource sur laquelle elle est utilisée. Auparavant, il aurait pu être nécessaire d'effectuer une recherche au sein du catalogue de ressources, désormais cette recherche externe n'est plus nécessaire en raison du balisage dans AWS KMS, et de nombreux autres services AWS.

Sécurité de l'infrastructure

La fonctionnalité de sécurité d'infrastructure vous propose des bonnes pratiques sur la façon de configurer AWS KMS afin de vous assurer que vous disposez d'une mise en œuvre agile qui peut évoluer avec votre activité, tout en protégeant vos informations sensibles.

Clés principales client

Dans AWS KMS, votre hiérarchie de clé démarre avec une clé CMK. Une clé CMK peut être utilisée pour chiffrer directement des blocs de données allant jusqu'à 4 Ko, elle peut être utilisée pour sécuriser les clés de données, qui protègent les données sous-jacentes de n'importe quelle taille.

Clés CMK gérées par AWS et gérées par le client

Les clés CMK se répartissent en deux grandes catégories : gérées par AWS et gérées par le client. Une clé CMK gérée par AWS est créée lorsque vous choisissez d'activer le chiffrement côté serveur à l'aide d'une ressource AWS sous la clé CMK gérée par AWS pour ce service pour la première fois (par exemple SSE-KMS). La clé CMK gérée par AWS est unique pour votre compte AWS et pour la région dans laquelle elle est utilisée. Une clé CMK gérée par AWS ne peut être utilisée que pour protéger les ressources au sein du service AWS spécifique pour lequel elle est créée. Il ne permet pas le niveau de contrôle granulaire qui fournit une clé CMK gérée par le client. Pour plus de contrôle, une bonne pratique consiste à utiliser une clé CMK gérée par le client dans tous les services AWS pris en charge et dans vos applications. Une clé CMK gérée par le client est créée à votre demande et doit être configurée en fonction de votre cas d'utilisation explicite.

Le tableau suivant résume les principales similitudes et différences entre les clés CMK gérées par AWS et les clés CMK gérées par le client.

	Clé CMK gérée par AWS	Clé CMK gérée par le client
Création	Générée par AWS au nom du client	Générée par le client
Rotation	Une fois tous les trois ans automatiquement	Une fois par an automatiquement après adhésion ou manuellement à la demande
Suppression	Ne peut pas être supprimée	Peut être supprimée
Portée de l'utilisation	Limitée à un service AWS	Contrôlée via la stratégie KMS/IAM
Stratégie de clé	Gérée par AWS	Gérée par le client
Gestion de l'accès utilisateur	Stratégie IAM	Stratégie IAM

Pour les clés CMK gérées par le client, vous avez deux options pour créer des clés sous-jacentes. Lorsque vous choisissez de créer une clé CMK à l'aide d'AWS KMS, vous pouvez laisser KMS créer le chiffrement pour vous, ou vous pouvez choisir d'importer vos propres clés. Ces deux options vous fournissent le même niveau de contrôle et d'audit pour l'utilisation de la clé CMK au sein de votre environnement. La possibilité d'importer vos propres éléments de chiffrement vous permet d'effectuer les opérations suivantes :

- Prouvez que vous avez généré les clés à partir de votre source approuvée dont caractère aléatoire répond à vos besoins.
- Utilisez des clés de votre propre infrastructure avec des services AWS, et utilisez AWS KMS pour gérer le cycle de vie de ces clés au sein d'AWS.
- Acquérez la possibilité de définir un délai d'expiration pour les clés dans AWS et de la supprimer manuellement, mais également le rendre disponible à nouveau à l'avenir.
- Possédez la copie originale des clés et la conserver en dehors d'AWS pour plus de durabilité et d'une reprise après sinistre au cours du cycle de vie des éléments de clé.

La décision d'utiliser des clés importées ou des clés générées par KMS dépend des stratégies de votre organisation et de vos exigences en matière de conformité.

Création et gestion de clés

AWS facilitant la création et la gestion des clés grâce à l'utilisation d'AWS KMS, nous vous recommandons de prévoir un plan sur la façon d'utiliser le service afin de mieux contrôler la zone autour des clés individuelles. Auparavant, vous avez peut-être utilisé la même clé dans différentes régions géographiques, environnements, ou même applications. Avec AWS KMS, vous devez définir les niveaux de classification des données et avoir au moins une clé CMK par niveau. Par exemple, vous pouvez définir une clé CMK pour les données classées comme « confidentielles », et ainsi de suite. Cela permet de garantir que seuls les utilisateurs autorisés n'ont des autorisations que pour les clés dont ils ont besoin pour effectuer leurs tâches.

Vous devez également décider de la façon dont vous souhaitez gérer l'utilisation d'AWS KMS. Pour la plupart des clients, la solution préférable consiste à créer des clés KMS dans chaque compte qui nécessite la possibilité de chiffrer et déchiffrer des données sensibles, mais il existe une autre option qui consiste à partager les clés CMK à partir de quelques comptes centralisés. Conserver les clés CMK dans le même compte que la majorité de l'infrastructure qui les utilise aide les utilisateurs à mettre en service et exécuter les services AWS qui utilisent ces clés. Les services AWS n'autorisent pas les recherches inter-comptes, sauf si le mandataire procédant à la recherche possède explicitement les autorisations List* sur les ressources détenues par le compte externe. Cela s'effectue uniquement via l'interface de ligne de commande ou le kit SDK, et pas par les recherches dans la console de service. En outre, en stockant les informations d'identification dans les comptes locaux, il peut être plus facile de déléguer des autorisations aux personnes connaissant les mandataires IAM qui ont besoin d'accéder aux

clés CMK spécifiques. Si vous partagez les clés via un modèle centralisé, les administrateurs AWS KMS doivent connaître l'Amazon Resource Name (ARN) complet de tous les utilisateurs de la CMK pour garantir le principe de moindre privilège. Sinon, les administrateurs risquent de fournir des autorisations exagérées sur les clés.

Votre organisation doit également tenir compte de la fréquence de rotation pour les clés CMK. De nombreuses organisations effectuer une rotation des clés CMK chaque année. Pour les clés CMK gérées par le client avec des clés générées par KMS, la mise en application est facile. Il vous suffit d'adhérer à un programme de rotation annuelle pour votre clé CMK. Lorsque la clé CMK doit faire l'objet d'une rotation, une nouvelle clé de stockage est créée et marquée comme clé active pour toutes les nouvelles demandes de protection des informations. L'ancienne clé de stockage reste disponible pour déchiffrer n'importe quel texte ayant été chiffré à l'aide de cette clé. Pour effectuer une rotation des clés CMK plus fréquemment, vous pouvez également appeler `UpdateAlias` pour pointer un alias vers une nouvelle clé CMK, comme il est décrit dans la section suivante. La méthode `UpdateAlias` fonctionne à la fois pour les clés CMK gérées par le client et les clés CMK avec clés importées. AWS a constaté que la fréquence de rotation de clé dépend grandement des lois, règlementations et stratégies d'entreprise.

Alias de clé

Un alias de clé vous permet d'extraire des utilisateurs de clé depuis l'ID de clé d'accès spécifique à la région sous-jacente et les ARN de clé. Les personnes autorisées peuvent créer un alias de clé permettant à leurs applications d'utiliser une clé CMK indépendamment de la région ou ldu programme de rotation. Ainsi, les applications à régions multiples peuvent utiliser le même alias de clé pour faire référence aux clés KMS dans plusieurs régions, sans se préoccuper de l'ID de clé ou de l'ARN de clé. Vous pouvez également déclencher manuellement la rotation d'une clé CMK en pointant un alias de clé donnée vers une autre clé CMK. Semblable à la façon dont DNS (Domain Name Services) permet à la production d'adresses IP, un alias de clé est le même pour l'ID de clé. Lorsque vous créez un alias de clé, nous vous recommandons de définir un schéma de dénomination applicable à vos comptes tel que *alias/<Environment>-<Function>-<Service Team>*.

Il est important de noter que l'alias de la clé CMK ne peut pas être utilisé dans les stratégies. Cela tient au fait que le mappage d'alias pour des clés peut être manipulé en dehors de la stratégie, ce qui permettrait d'une escalade de privilèges. Par conséquent, les ID de clés d'accès doivent être utilisés dans les stratégies de clé KMS, stratégies IAM et droits KMS.

Utilisation d'AWS KMS à l'échelle

Comme il a été indiqué précédemment, une bonne pratique consiste à utiliser au moins une clé CMK pour une catégorie de données particulière. Cela vous permet de définir des stratégies qui délimitent les autorisations pour la clé et, par conséquent, les données à des utilisateurs

autorisés. Vous pouvez choisir de mieux répartir vos données sur plusieurs clés CMK afin de fournir des contrôles de sécurité plus importants au sein d'une même catégorie de données.

AWS recommande d'utiliser le chiffrement d'enveloppe KMS pour mettre à l'échelle votre implémentation. Le chiffrement de l'enveloppe correspond à la pratique de chiffrement des données en texte brut avec une clé de données unique, suivi du chiffrement de la clé de données avec une clé de chiffrement de clé (KEK). Dans AWS KMS, la clé CMK est la KEK. Vous pouvez chiffrer votre message avec la clé de données, puis chiffrer la clé de données avec la clé CMK. Ensuite, la clé de données chiffrée peut être stockée en même temps que le message chiffré. Vous pouvez mettre en cache la version de la clé de données en texte brut en vue d'une utilisation répétée, ce qui réduit le nombre de demandes à AWS KMS. En outre, le chiffrement d'enveloppe peut vous aider à concevoir votre application pour la reprise après sinistre. Vous pouvez transférer vos données chiffrées en l'état entre régions et vous n'aurez qu'à chiffrer de nouveau les clés de données avec les clés CMK propres à la région.

L'équipe cryptographie AWS a publié un kit SDK de chiffrement AWS qui facilite une utilisation efficace d'AWS KMS. Ce kit SDK met en œuvre en toute transparence les détails de bas niveau pour l'utilisation d'AWS KMS. Il fournit également aux développeurs des options pour protéger leurs clés de données après utilisation et ainsi veiller à ce que les performances de leur application ne soient pas considérablement affectées par le chiffrement de vos données sensibles.

Protection des données

La fonctionnalité de protection des données d'AWS concerne certains des cas d'utilisation courants d'AWS KMS au sein de votre organisation pour protéger vos informations sensibles.

Cas d'utilisation courants d'AWS KMS

Chiffrement de données PCI à l'aide d'AWS KMS

Les contrôles de sécurité et de qualité mis en place dans AWS KMS ayant été validés et certifiés pour répondre aux exigences de la norme PCI DSS de niveau 1, vous pouvez directement chiffrer votre numéro PAN (numéro de compte primaire) avec une clé CMK AWS KMS. L'utilisation d'une clé CMK pour chiffrer directement les données allège en partie les tâches de gestion des bibliothèques de chiffrement. En outre, une clé CMK ne peut pas être exportée à partir d'AWS KMS, ce qui résout le problème de stockage non sécurisé de la clé de chiffrement. Comme toutes les demandes KMS sont consignées dans CloudTrail, l'utilisation de la clé CMK peut être vérifiée en examinant les journaux CloudTrail. Il est important de connaître la limite de demandes par seconde lors de la conception des applications qui utilisent directement la clé CMK pour protéger les données PCI (Payment Card Industry).

Gestion secrète avec AWS KMS et Amazon S3

Même si AWS KMS fournit principalement les fonctions de gestion des clés, vous pouvez exploiter AWS KMS et Amazon S3 pour créer votre propre solution de gestion secrète.

Créez un nouveau compartiment Amazon s3 pour contenir vos secrets. Déployez une stratégie de compartiment sur le compartiment pour limiter l'accès aux seules personnes autorisées et services. Les codes secrets stockés dans le compartiment utilisent un préfixe prédéfini par fichier pour permettre un contrôle granulaire de l'accès aux secrets. Chaque clé secrète, lorsqu'elle est placée dans le compartiment S3, est chiffrée à l'aide d'une clé KMS gérée par le client. En outre, en raison de la nature hautement sensible des informations qui sont stockées dans ce compartiment, la journalisation des accès S3 où des CloudTrail datèrent Events est activée à des fins d'audit. Ensuite, lorsqu'un utilisateur ou un service requiert l'accès à la clé secrète, il endosse une identité au sein d'AWS qui dispose des autorisations pour utiliser l'objet dans le compartiment S3, ainsi que la clé KMS. Une application qui s'exécute dans une instance EC2 utilise un rôle de profil d'instance qui possède les autorisations nécessaires.

Chiffrement des variables d'environnement Lambda

Par défaut, lorsque vous créez ou mettez à jour des fonctions Lambda qui utilisent les variables d'environnement, ces variables sont chiffrées à l'aide d'AWS KMS. Lorsque votre fonction Lambda est appelée, ces valeurs sont déchiffrées et mises à disposition pour le code Lambda. Vous avez la possibilité d'utiliser la clé KMS par défaut pour Lambda ou de spécifier une clé CMK de votre choix.

Pour mieux protéger vos variables d'environnement, vous devez sélectionner la case à cocher « Enable Encryption helpers (Activer les assistants de chiffrement) ». En sélectionnant cette option, vos variables d'environnement seront également chiffrées individuellement à l'aide d'une clé CMK de votre choix, puis votre fonction Lambda devra spécifiquement déchiffrer chaque variable d'environnement chiffrée qui est nécessaire.

Chiffrement de données au sein de Systems Manager Parameter Store

Amazon EC2 Systems Manager est un ensemble de fonctionnalités qui vous aide à automatiser des tâches de gestion à l'échelle. Pour stocker de manière efficace et référence des données de configuration sensibles telles que les mots de passe, clés de licence et certificats, le Parameter Store vous permet de protéger les informations sensibles dans des paramètres de chaîne sécurisée.

Une chaîne sécurisée correspond à des données sensibles qui doivent être stockées et référencées de manière sécurisée. Si vous avez des données que vous ne voulez pas que les utilisateurs modifient ou référencent en texte clair, telles que les mots de passe de jointure de domaine ou les clés de licence, spécifiez ces valeurs à l'aide du type de données Secure String. Vous devez utiliser les chaînes sécurisées dans les cas suivants :

- Vous voulez utiliser les données/paramètres sur les services AWS sans exposer les valeurs en texte clair dans les commandes, les fonctions, les journaux de l'agent ou les journaux CloudTrail.
- Vous voulez contrôler qui a accès aux données sensibles.
- Vous souhaitez être en mesure de vérifier les accès aux données sensibles à l'aide de CloudTrail.
- Vous voulez un chiffrement au niveau AWS pour vos données sensibles et vous voulez utiliser vos propres clés de chiffrement pour gérer l'accès.

Si vous sélectionnez cette option lorsque vous créez votre paramètre, Systems Manager chiffre la valeur lorsqu'elle est passée dans une commande et la déchiffre lors de son traitement sur l'instance gérée. Le chiffrement est géré par AWS KMS et peut être une clé KMS par défaut pour le Systems Manager, sinon vous pouvez également spécifier une clé CMK par paramètre.

L'application de chiffrement de données au repos au sein des services AWS

Votre organisation peut exiger le chiffrement de toutes les données qui correspondent à une classification spécifique. En fonction du service spécifique, vous pouvez appliquer le chiffrement des données par le biais de` politiques préventives ou de contrôles de détection. Pour certains services, tels qu'Amazon S3, une stratégie peut empêcher le stockage des données non chiffrées. Pour les autres services, le mécanisme le plus efficace consiste à surveiller la création de ressources de stockage et à vérifier si le chiffrement est activé de manière appropriée. Dans le cas où un stockage non chiffré est créé, un certain nombre de réponses sont possibles allant de la suppression de la ressource de stockage à la notification à un administrateur.

Chiffrement de données au repos avec Amazon S3

Avec d'Amazon S3, il est possible de déployer une stratégie de compartiment S3 qui permet de garantir que tous les objets en cours de téléchargement sont chiffrés. La stratégie se présente à peu près comme suit :

```
{
   “Version”:”2012-10-17”,
   “Id”:”PutObjPolicy”,
   “Statement”:[{
         “Sid”:”DenyUnEncryptedObjectUploads”,
         “Effect”:”Deny”,
         “Principal”:”*”,
         “Action”:”s3:PutObject”,
```

```
            "Resource":"arn:aws:s3:::YourBucket/*",
            "Condition":{
               "StringNotEquals":{
                  "s3:x-amz-server-side-encryption":"aws:kms"
               }
            }
         }
      ]
}
```

Notez que cela n'entraîne pas le chiffrement des objets déjà présents dans le compartiment. Cette stratégie refuse les tentatives d'ajout de nouveaux objets dans le compartiment, sauf si ces objets sont chiffrés. Les objets déjà présents dans le compartiment avant l'application de cette stratégie resteront chiffrés ou non chiffrés en fonction de la façon dont ils ont été téléchargés.

Chiffrement de données au repos avec Amazon EBS

Vous pouvez créer des Amazon Machine Images (AMI) qui utilisent les volumes de démarrage EBS chiffrés et utilisent l'AMI pour lancer des instances EC2. Les données stockées sont chiffrées, tout comme le chemin de transfert de données entre le volume EBS et l'instance EC2. Les données sont déchiffrées sur l'hyperviseur de cette instance selon vos besoins, puis stockées uniquement dans la mémoire. Cette fonctionnalité contribue à vos efforts de sécurité, de conformité, et d'audit en vous permettant de vérifier que toutes les données que vous stockez sur le volume EBS sont chiffrées, et si elles sont stockées sur un volume de démarrage ou sur un volume de données. De plus, dans la mesure où cette fonctionnalité permet l'utilisation d'AWS KMS, vous pouvez suivre et contrôler toutes les utilisations des clés de chiffrement.

Il existe deux méthodes pour garantir que les volumes EBS sont toujours chiffrés. Vous pouvez vérifier que l'indicateur de chiffrement dans le cadre du contexte `CreateVolume` est défini sur « true » via une stratégie IAM. Si l'indicateur n'est pas activé (« true »), alors la stratégie IAM peut empêcher une personne de créer le volume EBS. L'autre méthode consiste à surveiller la création de volumes EBS. Si un nouveau volume EBS est créé, CloudTrail consigne un événement. Une fonction Lambda peut être déclenchée par les événements CloudTrail pour vérifier si le volume EBS est chiffré ou non, et également pour vérifier quelle clé KMS a été utilisée pour le chiffrement.

Une fonction AWS Lambda peut répondre à la création d'un volume non chiffré de différentes façons. La fonction peut appeler l'API `CopyImage` avec l'option chiffrée pour créer une nouvelle version chiffrée du volume EBS, puis attachez-le à l'instance et supprimez l'ancienne version. Certains clients choisissent de supprimer automatiquement l'instance EC2 qui possède le volume non chiffré. D'autres choisissent de placer automatiquement l'instance en

quarantaine en appliquant des groupes de sécurité qui empêchent la plupart des connexions entrantes. Il est également facile d'écrire une fonction Lambda qui publie un message sur une rubrique Amazon Simple Notification Service (SNS) indiquant aux administrateurs d'effectuer une enquête et une intervention manuelle. Notez que la plupart des réponses d'exécution être apportées par programmation sans intervention humaine.

Chiffrement de données au repos avec Amazon RDS

Amazon Relational Database Service (RDS) s'appuie sur Amazon EBS Encryption pour fournir un chiffrement de disque complet pour les volumes de base de données. Lorsque vous créez une instance de base de données chiffrée avec Amazon RDS, Amazon RDS crée un volume EBS chiffré en votre nom pour stocker la base de données. Les données stockées au repos sur le volume, les instantanés de base de données, les sauvegardes automatisées et les réplicas en lecture sont tous chiffrés sous la clé CMK KMS que vous avez spécifiée lorsque vous avez créé l'instance de base de données.

Comme c'est le cas avec Amazon EBS, vous pouvez configurer une fonction AWS Lambda à surveiller pour la création de nouvelles instances RDS par l'appel d'API `CreateDBInstance` via CloudTrail. Dans l'événement `CreateDBInstance`, assurez-vous que l'option `KmsKeyId` est définie sur la clé CMK.

Réponse aux incidents

La fonctionnalité de gestion des incidents se concentre sur les fonctionnalités de votre organisation pour corriger des incidents qui peuvent impliquer AWS KMS.

Automatisation de la sécurité d'AWS KMS

Au cours de la supervision de vos clés CMK, si une action spécifique est détectée, une fonction AWS Lambda peut être configurée pour désactiver la clé CMK ou appliquer des actions de réponse aux incidents dictées par vos politiques de sécurité locale. Sans intervention humaine, un risque d'exposition pourrait être coupé en quelques minutes grâce à l'utilisation d'outils d'automatisation au sein d'AWS.

Suppression et désactivation des clés CMK

S'il est possible de supprimer une clé CMK, celle-ci présente de nombreuses ramifications dans une organisation. Vous devez tout d'abord déterminer si elle est suffisante pour définir l'état de la clé CMK sur désactivée pour les clés que vous n'avez plus l'intention d'utiliser. Cela permet d'empêcher toute utilisation future de la clé CMK. La clé CMK est toutefois toujours disponible, et peut être réactivée à l'avenir si c'est nécessaire. Les clés désactivées sont toujours stockées par AWS KMS ; par conséquent, elles continuent à encourir des frais de

stockage récurrents. Vous devez envisager sérieusement la désactivation des clés au lieu de les supprimer jusqu'à ce que vous ayez confiance dans la gestion de leurs données chiffrées.

La suppression d'une clé doit se faire avec la plus grande précaution. Les données ne peuvent pas être déchiffrées si la clé CMK correspondante a été supprimée. En outre, toute suppression de clé CMK est définitive. AWS n'a aucun moyen de récupérer une clé CMK une fois qu'elle a été supprimée. Tout comme avec les autres opérations critiques dans AWS, vous devez appliquer une stratégie qui requiert l'authentification MFA pour la suppression CMK.

Pour garantir qu'une clé CMK n'est pas supprimée par erreur, KMS applique une période d'attente minimum de 7 jours avant que la clé CMK soit réellement supprimée. Vous pouvez choisir d'allonger cette période d'attente jusqu'à une valeur maximale de 30 jours. Pendant la période d'attente, la clé CMK est toujours stockée dans KMS à l'état « En attente de suppression ». Elle ne peut pas être utilisée pour chiffrer ou déchiffrer des opérations. Toute tentative d'utiliser une clé qui est à l'état « En attente de suppression » pour le chiffrement ou le déchiffrement est consignée dans CloudTrail. Vous pouvez définir une alarme Amazon CloudWatch pour ces événements dans vos journaux CloudTrail. Vous avez ainsi la possibilité d'annuler le processus de suppression si nécessaire. Jusqu'à la fin de la période d'attente, la clé CMK peut être récupérée à partir de l'état « En attente de suppression » et restaurée à l'état activé ou désactivé.

Enfin, il convient également de noter que si vous utilisez une clé CMK avec des clés importées, vous pouvez supprimer les clés importées immédiatement. Cette opération diffère de la suppression directe d'une clé CMK de différentes manières. Lorsque vous effectuez l'action `DeleteImportedKeyMaterial`, AWS KMS supprime les clés et la clé CMK passe à l'état en attente d'importation. Lorsque les clés sont supprimées, la clé CMK est immédiatement inutilisable. Il n'y a pas de période d'attente. Pour permettre à nouveau l'utilisation de la clé CMK, vous devez réimporter les mêmes clés. La suppression des clés affecte la clé CMK immédiatement, mais les clés de chiffrement des données qui sont activement en cours d'utilisation par les services AWS ne sont pas immédiatement affectées.

Par exemple, supposons qu'une clé CMK utilisant une clé importée a été utilisée pour chiffrer un objet dans un compartiment S3 à l'aide de SSE-KMS.[8] Juste avant que vous chargiez l'objet dans le compartiment S3, vous devez placer la clé importée dans votre clé CMK. Une fois que l'objet est transféré, vous pouvez supprimer vos clés de cette clé CMK. L'objet continuera à rester dans le compartiment S3 à l'état chiffré, mais personne ne pourra y accéder jusqu'à ce que les mêmes clés soient réimportées dans la clé CMK. Ce flux nécessite évidemment une automatisation précise pour importer et supprimer les clés d'une clé CMK, mais il apporte un niveau supplémentaire de contrôle au sein d'un environnement.

Conclusion

AWS KMS fournit à votre organisation un service entièrement géré pour contrôler de manière centralisée vos clés de chiffrement. Son intégration native avec d'autres services AWS permet à AWS KMS de chiffrer plus facilement les données et processus que vous stockez.

En prenant le temps de créer et de mettre en œuvre AWS KMS correctement, vous pouvez garantir que vos clés de chiffrement sont sécurisées et disponibles pour les applications et leurs utilisateurs autorisés. De plus, vous pouvez présenter à vos auditeurs des journaux détaillés associés à leur utilisation.

Participants

Les personnes et organisations suivantes ont participé à l'élaboration de ce document :

- Matthew Bretan, Conseiller senior en sécurité, AWS Professional Services
- Sree Pisharody, Chef de produit senior - Technical, AWS Cryptography
- Ken Beer, Directeur principal développement logiciel, AWS Cryptography
- Brian Wagner, Conseiller en sécurité, AWS Professional Services
- Eugene Yu, Conseiller en gestion, AWS Professional Services
- Michael St.Onge, Architecte Global Cloud Security, Services professionnels AWS
- Balaji Palanisamy, Conseiller principal, Services professionnels AWS
- Jonathan Rault, Conseiller principal, Services professionnels AWS
- Reef Dsouza, Conseiller principal, Services professionnels AWS
- Paco Hope, Conseiller principal, Services professionnels AWS

Révisions du document

Pour la version la plus récente de ce livre blanc, veuillez consulter :

https://d0.awsstatic.com/whitepapers/KMS-Best-Practices.pdf

Remarques

[1] http://docs.aws.amazon.com/kms/latest/developerguide/overview.html

[2] https://d0.awsstatic.com/whitepapers/KMS-Cryptographic-Details.pdf

[3] https://d0.awsstatic.com/whitepapers/aws_cloud_adoption_framework.pdf

[4] https://d0.awsstatic.com/whitepapers/AWS_CAF_Security_Perspective.pdf

[5] http://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

[6] http://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html#key-policy-default-allow-root-enable-iam

[7] http://docs.aws.amazon.com/kms/latest/developerguide/policy-conditions.html#conditions-kms-via-service

[8] http://docs.aws.amazon.com/kms/latest/developerguide/services-s3.html#sse