Erste Schritte mit externem Verschlüsselungsmanagement

Element Software

NetApp
October 01, 2024

Inhalt

Erste Schritte mit externem Verschlüsselungsmanagement

EKM (External Key Management) bietet eine sichere Verwaltung des Authentifizierungsschlüssels (AK) in Verbindung mit einem externen EKS-Server (Off-Cluster). Die AKS werden zum Sperren und Entsperren von Self-Encrypting Drives (SEDs) verwendet, wenn "Verschlüsselung für Daten im Ruhezustand"auf dem Cluster aktiviert ist. Der EKS sorgt für die sichere Erzeugung und Lagerung der AKS. Der Cluster verwendet für die Kommunikation mit dem EKS das Key Management Interoperability Protocol (KMIP), ein OASIS-definiertes Standardprotokoll.

- "Externe Verwaltung einrichten"
- "Verschlüsselung der Software beim Rest-Master-Schlüssel"
- "Wiederherstellen von nicht zugänglichen oder ungültigen Authentifizierungsschlüsseln"
- "Befehle für externes Verschlüsselungsmanagement-API"

Weitere Informationen

- "CreateCluster API, die zur Aktivierung der Softwareverschlüsselung im Ruhezustand verwendet werden kann"
- "Dokumentation von SolidFire und Element Software"
- "Dokumentation für frühere Versionen von NetApp SolidFire und Element Produkten"

Externes Verschlüsselungsmanagement einrichten

Sie können diese Schritte ausführen und die aufgeführten Element-API-Methoden verwenden, um Ihre externe Verschlüsselungsmanagementfunktion einzurichten.

Was Sie benötigen

- Wenn Sie externes Verschlüsselungsmanagement in Kombination mit Softwareverschlüsselung im Ruhezustand einrichten, haben Sie die Softwareverschlüsselung im Ruhezustand mit der Methode auf einem neuen Cluster aktiviert"CreateCluster erstellen", das keine Volumes enthält.

Schritte

1. Bauen Sie eine Vertrauensbeziehung mit dem externen Key Server (EKS) auf.
   a. Erstellen Sie ein öffentliches/privates Schlüsselpaar für den Element-Cluster, das zum Aufbau einer Vertrauensbeziehung mit dem Schlüsselserver verwendet wird, indem Sie die folgende API-Methode aufrufen: "CreatePublicPrivateKeyPair"
   b. Holen Sie sich die Zertifikatsign-Anforderung (CSR), die die Zertifizierungsstelle unterzeichnen muss. Der CSR ermöglicht dem Schlüsselserver zu überprüfen, ob das Element-Cluster, das auf die Schlüssel zugreift, als Element-Cluster authentifiziert ist. Rufen Sie die folgende API-Methode auf: "GetClientCertificateSignRequest"
   c. Verwenden Sie die EKS/Zertifizierungsstelle, um den abgerufenen CSR zu unterzeichnen. Weitere Informationen finden Sie in der Dokumentation von Drittanbietern.
2. Erstellen Sie auf dem Cluster einen Server und Provider, um mit dem EKS zu kommunizieren. Ein

Schlüsselanbieter legt fest, wo ein Schlüssel abgerufen werden soll, und ein Server definiert die spezifischen Attribute der EKS, die mit kommuniziert werden.

a. Erstellen Sie einen Schlüsselanbieter, bei dem sich die Schlüsselserverdetails befinden, indem Sie die folgende API-Methode aufrufen: "CreateKeyProviderKmip"

b. Erstellen Sie einen Schlüsselserver, der das signierte Zertifikat und das öffentliche Schlüsselzertifikat der Zertifizierungsstelle bereitstellt, indem Sie die folgenden API-Methoden aufrufen: "CreateKeyServerkmip" "TestKeyServerkmip"

   Wenn der Test fehlschlägt, überprüfen Sie die Serverkonnektivität und -Konfiguration. Wiederholen Sie dann den Test.

c. Fügen Sie den Schlüsselserver in den Container des Schlüsselanbieters ein, indem Sie die folgenden API-Methoden aufrufen: "AddKeyServerToProviderKmip" "TestKeyProviderKmip"

   Wenn der Test fehlschlägt, überprüfen Sie die Serverkonnektivität und -Konfiguration. Wiederholen Sie dann den Test.

3. Führen Sie als nächsten Schritt für die Verschlüsselung im Ruhezustand einen der folgenden Schritte aus:

   a. (Für Hardware-Verschlüsselung im Ruhezustand) Aktivieren Sie"Hardware-Verschlüsselung für Daten im Ruhezustand", indem Sie die ID des Schlüsselanbieters angeben, der den Schlüsselserver enthält, der zum Speichern der Schlüssel verwendet wird, indem Sie die API-Methode aufrufen"EnableVerschlüsselungAtZiel".

   > Sie müssen die Verschlüsselung im Ruhezustand über die aktivieren"API". Die Aktivierung der Verschlüsselung im Ruhezustand mithilfe der vorhandenen Element UI-Schaltfläche bewirkt, dass die Funktion mithilfe intern generierter Schlüssel zurückgesetzt wird.

   b. (Für Softwareverschlüsselung im Ruhezustand) um "Softwareverschlüsselung für Daten im Ruhezustand"den neu erstellten Schlüsselanbieter zu nutzen, geben Sie die ID des Schlüsselanbieters an die API-Methode weiter"RekeySoftwareVerschlüsselungAtRestMasterKey".

Weitere Informationen

- "Aktivieren und Deaktivieren der Verschlüsselung für ein Cluster"
- "Dokumentation von SolidFire und Element Software"
- "Dokumentation für frühere Versionen von NetApp SolidFire und Element Produkten"

Verschlüsselung der Software beim Rest-Master-Schlüssel

Mit der Element-API können Sie einen vorhandenen Schlüssel neu Schlüssel rekeykey. Durch diesen Prozess wird ein neuer Master-Ersatzschlüssel für Ihren externen Verschlüsselungsmanagement-Server erstellt. Master-Schlüssel werden immer durch neue Master-Schlüssel ersetzt und nie dupliziert oder überschrieben.

Unter Umständen müssen Sie die Daten im Rahmen eines der folgenden Verfahren erneut keywichtigen:

- Erstellen Sie einen neuen Schlüssel im Rahmen einer Änderung vom internen Verschlüsselungsmanagement bis zum externen Verschlüsselungsmanagement.
- Erstellen Sie einen neuen Schlüssel als Reaktion auf oder als Schutz gegen sicherheitsrelevante

Ereignisse.

> Dieser Prozess ist asynchron und gibt eine Antwort zurück, bevor der Rekeyvorgang abgeschlossen ist. Sie können die Methode verwenden"GetAsyncResult", um das System abzufragen, um zu sehen, wann der Vorgang abgeschlossen ist.

Was Sie benötigen

- Sie haben die Softwareverschlüsselung im Ruhezustand mit der Methode auf einem neuen Cluster aktiviert"CreateCluster erstellen", das keine Volumes enthält und keine I/O-Vorgänge hat Verwenden Sie Link:../API/reference_element_api_getsoftwareencryptionatrestinfo.html[`GetSoftwareEncryptionatRestInfo`], um zu bestätigen, dass der Status vor dem Fortfahren ist `enabled`.
- Sie haben "Sie haben eine Vertrauensbeziehung aufgebaut" zwischen dem SolidFire-Cluster und einem externen Schlüsselserver (EKS). Führen Sie die Methode aus"TestKeyProviderKmip", um zu überprüfen, ob eine Verbindung zum Schlüsselanbieter hergestellt wurde.

Schritte

1. Führen Sie den Befehl aus "ListKeyProvidersKmip"und kopieren Sie die Key Provider ID (`keyProviderID`).
2. Führen Sie den "RekeySoftwareVerschlüsselungAtRestMasterKey" mit dem `keyManagementType` Parameter als `external` und `keyProviderID` als ID-Nummer des Schlüsselanbieters aus dem vorherigen Schritt aus:

```
{
 "method": "rekeysoftwareencryptionatrestmasterkey",
 "params": {
   "keyManagementType": "external",
   "keyProviderID": "<ID number>"
 }
}
```

3. Kopieren Sie den `asyncHandle` Wert aus der `RekeySoftwareEncryptionAtRestMasterKey` Befehlsantwort.
4. Führen Sie den Befehl mit dem Wert aus dem `asyncHandle` vorherigen Schritt aus"GetAsyncResult", um die Konfigurationsänderung zu bestätigen. In der Befehlsantwort sollten Sie sehen, dass die ältere Master Key-Konfiguration mit neuen Schlüsselinformationen aktualisiert wurde. Kopieren Sie die neue Schlüssel-Provider-ID zur Verwendung in einem späteren Schritt.

```
{
   "id": null,
   "result": {
     "createTime": "2021-01-01T22:29:18Z",
     "lastUpdateTime": "2021-01-01T22:45:51Z",
     "result": {
       "keyToDecommission": {
         "keyID": "<value>",
         "keyManagementType": "internal"
     },
     "newKey": {
       "keyID": "<value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "operation": "Rekeying Master Key. Master Key management being
transferred from Internal Key Management to External Key Management with
keyProviderID=<value>",
     "state": "Ready"
   },
   "resultType": "RekeySoftwareEncryptionAtRestMasterKey",
   "status": "complete"
}
```

5. Führen Sie den Befehl aus `GetSoftwareEncryptionatRestInfo`, um zu bestätigen, dass neue Schlüsseldetails, einschließlich der `keyProviderID`, aktualisiert wurden.

```
{
   "id": null,
   "result": {
     "masterKeyInfo": {
       "keyCreatedTime": "2021-01-01T22:29:18Z",
       "keyID": "<updated value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "rekeyMasterKeyAsyncResultID": <value>
     "status": "enabled",
     "version": 1
   },
}
```

Weitere Informationen

- "Storage-Management mit der Element API"
- "Dokumentation von SolidFire und Element Software"
- "Dokumentation für frühere Versionen von NetApp SolidFire und Element Produkten"

Wiederherstellen von nicht zugänglichen oder ungültigen Authentifizierungsschlüsseln

Gelegentlich kann es zu einem Fehler kommen, der Benutzereingriff erfordert. Im Fehlerfall wird ein Cluster-Fehler (auch als Cluster-Fehlercode bezeichnet) generiert. Die beiden wahrscheinlichsten Fälle werden hier beschrieben.

Das Cluster kann die Laufwerke nicht entsperren, da ein KmipServerFault-Clusterfehler vorliegt.

Dies kann auftreten, wenn das Cluster zum ersten Mal gebootet wird und der Schlüsselserver nicht zugänglich ist oder der erforderliche Schlüssel nicht verfügbar ist.

1. Befolgen Sie ggf. die Wiederherstellungsschritte in den Cluster-Fehlercodes.

Es kann ein SliceServiceUnHealthy Fehler gesetzt werden, weil die Metadaten-Laufwerke als fehlgeschlagen markiert und in den Status „verfügbar“ gesetzt wurden.

Schritte zum Löschen:

1. Fügen Sie die Laufwerke erneut hinzu.
2. Prüfen Sie nach 3 bis 4 Minuten, ob der `sliceServiceUnhealthy` Fehler behoben ist.

Weitere Informationen finden Sie unter "Cluster-Fehlercodes" .

Befehle für externes Verschlüsselungsmanagement-API

Liste aller zur Verwaltung und Konfiguration von EKM verfügbaren APIs.

Wird zum Aufbau einer Vertrauensbeziehung zwischen dem Cluster und externen Servern im Kundenbesitz verwendet:

- CreatePublicPrivateKeyPair
- GetClientCertificateSignRequest

Wird zur Definition der spezifischen Details externer kundeneigener Server verwendet:

- CreateKeyServerkmip
- ModifyKeyServerkmip
- DeleteKeyServerkmip
- GetKeyServerkmip
- ListKeyServersKmip

- TestKeyServerkmip

Wird zur Erstellung und Verwaltung von Schlüsselanbietern verwendet, die externe Schlüsselserver verwalten:

- CreateKeyProviderKmip
- DeleteKeyProviderKmip
- AddKeyServerToProviderKmip
- RemoveKeyServerFromProviderKmip
- GetKeyProviderKmip
- ListKeyProvidersKmip
- RekeySoftwareVerschlüsselungAtRestMasterKey
- TestKeyProviderKmip

Informationen zu den API-Methoden finden Sie unter "API-Referenzinformationen".