# 安全性
## Astra Trident

NetApp
April 04, 2024

# 目录

安全性 . . . . . 1
安全性 . . . . . 1
Linux统一密钥设置(LUKS) . . . . . 2

# 安全性

## 安全性

使用此处列出的建议确保Astra Trident安装安全。

### 在自己的命名空间中运行 **Astra Trident**

请务必防止应用程序，应用程序管理员，用户和管理应用程序访问 Astra Trident 对象定义或 Pod ，以确保存储可靠并阻止潜在的恶意活动。

要将其他应用程序和用户与Astra Trident分开、请始终在自己的Kubernetes命名空间中安装Astra Trident (`trident`）。将 Astra Trident 置于自己的命名空间中可确保只有 Kubernetes 管理人员才能访问 Astra Trident Pod 以及存储在命名空间 CRD 对象中的项目（如适用，还包括后端和 CHAP 密码）。
您应确保仅允许管理员访问Astra Trident命名空间、从而访问 `tridentctl` 应用程序。

### 对 **ONTAP SAN** 后端使用 **CHAP** 身份验证

Astra Trident支持对ONTAP SAN工作负载进行基于CHAP的身份验证(使用 `ontap-san` 和 `ontap-san-economy` 驱动程序)。NetApp 建议将双向 CHAP 与 Astra Trident 结合使用，以便在主机和存储后端之间进行身份验证。

对于使用SAN存储驱动程序的ONTAP 后端、Astra Trident可以通过设置双向CHAP并管理CHAP用户名和密码 `tridentctl`。
请参见 "此处" 了解 Astra Trident 如何在 ONTAP 后端配置 CHAP 。

> Trident 20.04 及更高版本支持 ONTAP 后端的 CHAP 。

### 对 **NetApp HCI** 和 **SolidFire** 后端使用 **CHAP** 身份验证

NetApp 建议部署双向 CHAP ，以确保主机与 NetApp HCI 和 SolidFire 后端之间的身份验证。Astra Trident 使用一个机密对象，每个租户包含两个 CHAP 密码。当Trident作为CSI配置程序安装时、它会管理CHAP密码并将其存储在中 `tridentvolume` 相应PV的CR对象。创建 PV 时， CSI Astra Trident 会使用 CHAP 密码启动 iSCSI 会话并通过 CHAP 与 NetApp HCI 和 SolidFire 系统进行通信。

> CSI Trident 创建的卷不与任何卷访问组关联。

在非 CSI 前端中，作为辅助节点上的设备连接卷的操作由 Kubernetes 处理。创建卷后，如果该租户的密钥尚不存在，则 Astra Trident 会对 NetApp HCI/SolidFire 系统进行 API 调用，以检索这些密钥。然后， Astra Trident 将这些机密传递给 Kubernetes 。位于每个节点上的 kubelet 通过 Kubernetes API 访问这些机密，并使用它们在访问卷的每个节点与卷所在的 NetApp HCI/SolidFire 系统之间运行 / 启用 CHAP 。

### 将**Astra Trident**与**NVE**和**NAE**结合使用

NetApp ONTAP 提供空闲数据加密、可在磁盘被盗、退回或重新利用时保护敏感数据。有关详细信息，请参见 "配置 NetApp 卷加密概述"。

- 如果在后端启用了NAE、则在Astra Trident中配置的任何卷都将启用NAE。

- 如果后端未启用NAE、则在Astra Trident中配置的任何卷都将启用NVE、除非将NVE加密标志设置为 `false` 在后端配置中。

> 在启用了NAE的后端的Astra Trident中创建的卷必须经过NVE或NAE加密。
>
> - 您可以将NVE加密标志设置为 `true` 在Trident后端配置中、覆盖NAE加密并按卷使用特定的加密密钥。
> - 将NVE加密标志设置为 `false` 在启用了NAE的后端、将创建启用了NAE的卷。您不能通过将NVE加密标志设置为来禁用NAE加密 `false`。

- 您可以通过将NVE加密标志显式设置为来在Astra Trident中手动创建NVE卷 `true`。

有关后端配置选项的详细信息、请参见：

- "ONTAP SAN配置选项"
- "ONTAP NAS配置选项"

# Linux统一密钥设置(LUKS)

## 您可以启用Linux统一密钥设置(LUKS)来对Astra Trident上的ONTAP SAN和ONTAP SAN经济卷进行加密。Astra Trident支持对LUKS加密卷执行密码短语轮换和卷扩展。

在Astra Trident中、LUKS加密的卷会按照建议使用AES-XTS-plain64 Cypher和模式 "NIST"。

开始之前

- 工作节点必须安装加密设置2.1或更高版本(但低于3.0)。有关详细信息，请访问 "Gitlab：密码设置"。
- 出于性能原因、我们建议员工节点支持高级加密标准新指令(AES-NI)。要验证AES-NI支持、请运行以下命令：

```
grep "aes" /proc/cpuinfo
```

如果未返回任何内容、则您的处理器不支持AES-NI。有关AES-NI的详细信息、请访问： "Intel：高级加密标准说明(AES-NI)"。

### 启用LUKS加密

您可以对ONTAP SAN和ONTAP SAN经济卷使用Linux统一密钥设置(Unified Key Setup、LUKS)启用每个卷的主机端加密。

步骤

1. 在后端配置中定义LUKS加密属性。有关ONTAP SAN的后端配置选项的详细信息、请参见 "ONTAP SAN配置选项"。

```
"storage": [
    {
        "labels":{"luks": "true"},
        "zone":"us_east_1a",
        "defaults": {
            "luksEncryption": "true"
        }
    },
    {
        "labels":{"luks": "false"},
        "zone":"us_east_1a",
        "defaults": {
            "luksEncryption": "false"
        }
    },
]
```

2. 使用 ... `parameters.selector` 使用LUKS加密定义存储池。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: netapp.io/trident
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

3. 创建一个包含LUKS密码短语的密钥。例如：

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

限制

LUKS加密的卷无法利用ONTAP 重复数据删除和数据压缩功能。

## 用于导入LUKS卷的后端配置

要导入LUKS卷、必须设置 `luksEncryption` to(`true` 在后端。。 `luksEncryption` Option可告知A作用 中的三端磁盘是否符合LUKS (`true`)或不符合LUKS (`false`)、如以下示例所示。

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

## 轮换LUKS密码短语

您可以轮换LUKS密码短语并确认轮换。

> 请勿忘记密码短语、除非您确认任何卷、快照或密钥不再引用它。如果引用的密码短语丢失、您可能无法挂载此卷、并且数据将保持加密状态且无法访问。

关于此任务

如果在指定新的LUKS密码短语后创建了挂载卷的POD、则会发生LUKS密码短语轮换。创建新的Pod时、Astra Trident会将卷上的LUKS密码短语与密钥中的活动密码短语进行比较。

- 如果卷上的密码短语与密钥中的活动密码短语不匹配、则会发生轮换。
- 如果卷上的密码短语与密钥中的活动密码短语匹配、则会显示 `previous-luks-passphrase` 参数将被忽略。

步骤

1. 添加 `node-publish-secret-name` 和 `node-publish-secret-namespace` StorageClass参数。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}
```

2. 确定卷或快照上的现有密码短语。

**Volume**

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

**Snapshot**

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. 更新卷的LUKS密钥以指定新密码短语和上一密码短语。确保 `previous-luke-passphrase-name` 和 `previous-luks-passphrase` 匹配上一个密码短语。

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. 创建一个新的装载卷的POD。这是启动轮换所必需的。
5. 验证密码短语是否已轮换。

**Volume**

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

**Snapshot**

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["B"]
```

结果

仅在卷和快照上返回新密码短语时、才会轮换密码短语。

> 如果返回两个密码短语、例如 `luksPassphraseNames: ["B", "A"]`、转出不完整。您可以触发新POD以尝试完成轮换。

## 启用卷扩展

您可以在LUKS加密的卷上启用卷扩展。

步骤

1. 启用 `CSINodeExpandSecret` 功能门(测试版1.25以上)。请参见 "Kubernetes 1.25：使用机密进行节点驱动型CSI卷扩展" 了解详细信息。
2. 添加 `node-expand-secret-name` 和 `node-expand-secret-namespace` StorageClass参数。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: netapp.io/trident
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

结果

启动联机存储扩展时、kubelet会将相应的凭据传递给驱动程序。