Fsas Technologies

P3AG-6042-03Z0

ETERNUS AB series オールフラッシュアレイ，ETERNUS HB series ハイブリッドアレイ

SANtricity セキュリティ強化ガイド

SANtricity のセキュアなデプロイメントのためのガイドライン

目次

図目次

表目次

はじめに

このセキュリティ強化ガイドでは、SANtricity 11.70.4 以降を導入して、情報システムの機密性、整合性、可用性に関する所定のセキュリティ目標を達成する際に役立つガイダンスを提供します。

第 3 版
2025 年 3 月

登録商標

本製品に関連する他社商標については、以下のサイトを参照してください。
https://www.fujitsu.com/jp/products/computing/storage/trademark/
本書では、本文中の™、® などの記号は省略しています。

本書の読み方

対象読者

本書は、ETERNUS AB/HB の設定、運用管理を行うシステム管理者、または保守を行うフィールドエンジニアを対象としています。必要に応じてお読みください。

関連マニュアル

ETERNUS AB/HB に関連する最新の情報は、以下のサイトで公開されています。
https://www.fujitsu.com/jp/products/computing/storage/manual/

本書の表記について

本文中の記号

本文中では、以下の記号を使用しています。

注意	お使いになるときに注意していただきたいことを記述しています。必ずお読みください。
備考	本文を補足する内容や、参考情報を記述しています。

1. 概要

セキュリティ上の脅威の拡大により、組織では、最も価値の高い資産である「データ」と「情報」を保護するにあたって固有の課題に直面しています。組織が直面する高度で動的な脅威と脆弱性は、ますます複雑化しています。潜在的な侵入者側の難読化技術と偵察技術の有効性が増しているため、システム管理者はデータおよび情報のセキュリティに積極的に取り組む必要があります。本書では、当社のソリューションに不可欠な機密性、整合性、可用性を活用することにより、そのようなタスクにおいてオペレータと管理者を支援することを目的としています。

2. ローカルストレージ管理者アカウント

2.1 ロール

ロールベースアクセス制御 (RBAC) を使用すると、ローカルユーザーは自分のロールと機能に必要なシステムとオプションにのみアクセスできます。SANtricity の RBAC ソリューションは、ユーザーの管理者アクセスをユーザーに定義されたロールのレベルに制限し、割り当てられたロールによって管理者がローカルユーザーを管理できるようにします。SANtricity には、いくつかのロールが事前定義されています。ローカルユーザーアカウントは静的で、割り当てられたロールは変更できません。表 2.1 に、SANtricity の事前定義済みのロールをリストします。

表 2.1 ローカルユーザー用に定義済みのロール

ロール	簡単な説明
Admin	最上位の管理アカウント。これは、ユーザーがすべてのローカルユーザーのパスワードを変更し、ストレージシステムでサポートされているすべてのコマンドを実行できる唯一のロールです。
Security	このロールは、監査ログの表示、安全な syslog サーバーの構成、LDAP/LDAPS サーバー接続の設定、証明書の管理など、ストレージシステムのセキュリティ構成を変更できます。 このロールには、プールやボリュームの作成 / 削除など、ストレージシステムのプロパティへの書き込みアクセス権はありませんが、読み取りアクセス権はあります。 また、ストレージシステムへの SYMbol アクセスを有効 / 無効にする権限も持っています。
Storage	このロールは、ストレージシステムのプロパティに対する完全な読み取り / 書き込みアクセス権がありますが、セキュリティ構成機能を実行する権限はありません。
Support	このロールは、ストレージシステム上のすべてのハードウェアリソース、障害データ、MEL/監査ログ、CFW アップグレードにアクセスできます。
Monitor	このロールは、ストレージシステムの全プロパティへの読み取り専用アクセス権を付与します。このユーザーはセキュリティ構成を表示できません。

表 2.2 に、SANtricity で事前定義されているユーザーとマップされているロールを示します。

表 2.2 ローカルユーザーとロールのマッピング

ローカルユーザー	マップされたロール
Admin	Root admin, security admin, storage admin, support admin, monitor
Security	Security admin, monitor
Storage	Storage admin, support admin, monitor
Support	Support admin, monitor
Monitor	Monitor

SANtricity は、Lightweight Directory Application Protocol (LDAP/LDAPS) および Active Directory もサポートします。LDAP/Active Directory ユーザーアカウントを使用すると、管理者はカスタムアクセス制御のロールを作成、変更、削除したり、特定のユーザーのアカウント制限を指定したりできます。

注意

セキュリティを強化するために Secure LDAP with TLS (LDAPS) を構成することを推奨します。

2.2 ログインパラメータとパスワードパラメータ

効果的なセキュリティ体制は、確立された組織のポリシー、ガイドライン、および組織に適用されるガバナンスまたは基準に準拠します。これらの要件の例には、パスワード長の要件、失敗したログイン試行の処理、およびそのようなアカウントの自動非アクティブログアウトが含まれます。SANtricity ソリューションは、これらのセキュリティ構成要素に対応する機能を提供します。

2.2.1 パスワードの最小桁数 (1 ～ 30) を設定するパスワードポリシーの構成

パスワード長の最小桁数を増やすことを推奨します。デフォルトは 8 桁です。この設定は、REST API または System Manager を使用して変更できます。

REST API

REST API を使用してパスワード最小長を変更するには、以下の REST API を使用します。

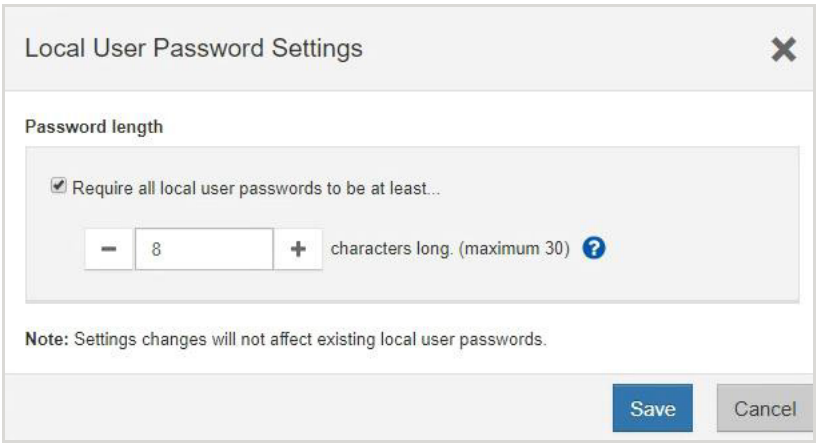
- API
  `Administration > POST /storage-systems/{system-id}/local-users/password-length`
- URL パラメータ
  `system-id`
- POST の本文

```
{
  "minimumPasswordLength": "＜文字列の長さ＞" // 長さを数値で指定
}
```

System Manager

System Manager を使用してパスワード最小長を変更するには、Settings > Access Management > View/Edit Settings ウィンドウ（図 2.1）に移動します。

図 2.1 System Manager のパスワード最小長変更ウィンドウ



2.2.2 ログイン試行の失敗によるロックアウト設定の構成

ログイン試行の失敗に対するロックアウトモード (IP アドレスとユーザー名)

デフォルトでは、SANtricity は、ストレージシステムへのログインに失敗した Web クライアントの IP アドレスを追跡します。この設定では、許可された最大試行回数を超えてロックアウトされた攻撃者は、別の IP アドレスを持つ別のホストに移動して、再度試行できます。ユーザーロックアウトモードに移行すると、SANtricity は、ログイン試行回数が最大回数に達した後にユーザー名をロックします。

注意

ロックアウトモードをデフォルトの IP アドレスベースではなく、ユーザーベースに変更することを推奨します。この設定は、以下に示すように REST API を使用して変更できます。

最大ログイン試行数に達した後のロックアウト時間 (分)

`lockoutTime` は、ユーザーが自分のアカウントにログインできない時間を分単位で表します。デフォルト値は 10 分です。

注意

ロックアウト時間を 10 分 (デフォルト) のままにすることを推奨します。

アカウントがロックアウトされるまでの最大ログイン試行回数

`maximumLoginAttempts` は、アカウントがロックアウトされる前にユーザーが試行できる最大試行回数を示します。デフォルト値は 6 です。たとえば、値が 6 に設定されている場合、6 回の失敗を入力すると、アカウントはロックアウトされます。7 回目のログイン試行には正しい資格情報が含まれている可能性がありますが、SANtricity は設定した `lockoutTime` 時間の終了後にユーザーが正しい資格情報で再度ログインするまでアクセスを許可しません。

ストレージアレイには最大 2 つのコントローラを搭載可能ですが、上記の設定はストレージアレイ全体に適用されます。ただし、ロックアウトは各コントローラで個別に管理され、アカウンティングは 2 つのコントローラ間で共有されません。ユーザーは、コントローラー A またはコントローラー B(どちらかのコントローラの IP アドレスを指定する) のいずれかを介してストレージアレイへのアクセスを試みることができるため、技術的に、試行できる回数は `maximumLoginAttempts` に設定した値の最大 2 倍になります。

注意

最大ログイン試行回数を 6 回 (デフォルト) から 5 回に減らすことを推奨します。
これは、6 回ログインに失敗すると、アカウントがロックアウトされることを意味します。

ロックアウトモード、ロックアウト時間、および最大ログイン試行回数の設定を変更するには、以下の REST API を使用します。

- API
  `Administration > POST /storage-systems/{system-id}/settings/lockout`
- URL パラメータ
  `system-id`
- POST の本文

```
{
  "lockoutMode": "＜使用するモード＞",     // 「user」「ip」のいずれか
  "lockoutTime": "＜ロックアウト時間（分）＞",
  "maximumLoginAttempts": "＜ロックアウトされるまでの試行回数＞"
}
```

2.2.3 アカウントの非アクティブ制限の設定

セッションの非アクティブ期間を 30 分 (デフォルト) から 15 分に短縮することを推奨します。この設定は、REST API または System Manager を使用して変更できます。

REST API

非アクティブ期間の設定を変更するには、以下の REST API を使用します。

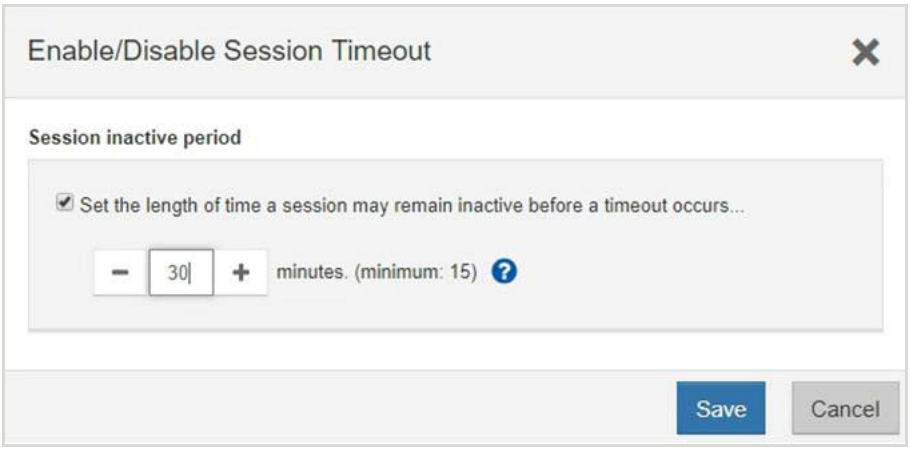
- API
  `Administration > POST /storage-systems/{system-id}/settings/sessions`
- URL パラメータ
  `system-id`
- POST の本文

```
{
  "sessionInactivePeriod": "<秒数>"
}
```

System Manager

System Manager を使用して非アクティブ期間の設定を変更するには、Settings > System > Enable/Disable Session Timeout（図 2.2）に移動します。

図 2.2 System Manager の非アクティブ期間変更ウィンドウ



2.2.4 SHA-512 のサポート

パスワードセキュリティを強化するために、SANtricity は SHA-2 パスワードハッシュ関数をサポートしており、新しく作成されたパスワードまたは変更されたパスワードのハッシュにデフォルトで SHA-512 を使用します。オペレータと管理者は、必要に応じてアカウントを期限切れにしたりロックしたりすることもできます。

3. システム管理者メソッド

3.1 コンソールアクセス

ストレージシステムのコンソールへの安全なアクセスを確立することは、安全なトラブルシューティングを行う上で重要です。最も一般的なコンソールアクセスオプションは、SSH、Telnet、および RSH です。SSH はリモートコンソールアクセスのための最も安全な業界標準のベストプラクティスです。

コンソールアクセスは、当社のサポートエンジニアがトラブルシューティングを行うためのものです。コンソールへのアクセスは次の方法で可能です。

- Serial USB ケーブル
  接続後、コンソールはユーザーログインとパスワードで保護されます。パスワードは、ユーザーが設定したストレージシステムのパスワードと同じです。
- SSH
  セキュリティ上の理由から、SSH はデフォルトで無効になっています。セキュリティで保護されたリモートアクセスが必要な場合、または固有のニーズがある場合は、SSH を手動で有効にする必要があります。ユーザーは、Secure CLI または SANtricity System Manager を使用して、SSH を手動で有効または無効にできます。

注意

当社のサポートエンジニアの指示がない限り、SSH を常に無効にすることを推奨します。次のセクションでは、HTTP を使用して SSH を無効および有効にする方法について説明します。

3.1.1 HTTP を介した SSH の無効化と有効化

HTTP を介して SSH を無効および有効にするには、以下の手順で行います。

1. リモートログインを許可します。

   注意

   - LAN の外部からリモートログインするユーザーは、SSH セッションを開始し、コントローラの設定を変更する必要があります。
   - セキュリティ上の理由から、リモートログインはテクニカルサポートのみが使用できるようにしてください。

2. [Hardware] を選択します。
3. 上部にある [Controllers & Components] タブをクリックします。
   グラフィックが変化し、ドライブではなくコントローラが表示されます。
4. リモートログインを有効にするコントローラをクリックします。
   コントローラのコンテキストメニューが表示されます。
5. [Enable remote login] チェックボックスにチェックを入れます。

3.2 SSH の多要素認証

ユーザーがシステムにアクセスする際、SSH 鍵と SSH パスワードの使用を強制することで SSH アクセスのセキュリティが強化できます。システム管理者による多要素認証（MFA）の設定には、REST API と System Manager がどちらも使用できます。

SSH で MFA を有効にするには、以下を準備します。

- SSH 共通鍵
  値を SSH サーバーの authorized_keys ファイルに配置するという、システムユーザーに対して SSH 認証を行う際の従来の手法となります。
- SSH に SSH 鍵と SSH パスワードを強制的に要求させる設定
  設定の有効化後、システムへのアクセス時に SSH 鍵と SSH パスワードが要求されます。

> **注意**
>
> SSH 鍵の利用の有効化は、パスワードの強制化とは個別に行います。SSH 鍵と密接な関係があるアカウントは、ローカルアカウントの diag と eos のみです。

SSH 鍵の利用を有効にすると、authorized_keys ファイルの内容に変更がある場合は毎回、ファイルの内容全体を提供する必要があります。REST API を使用する場合は、GET 呼び出しを行いファイルの内容を取得してから、POST 呼び出しを行いファイルの内容を更新します。こうして、ファイルに対して変更、移動、更新、削除の操作が行えるようになります。SSH 鍵をすべて削除すると、空の authorized_keys ファイルが書き込まれ、SSH では SSH 鍵を使用した認証ができなくなります。

ユーザーインターフェースとして、authorized_keys ファイルのエディタと MFA 有効化のスイッチを備えたダイアログボックスが表示されます。

最も安全な環境を有効にすること（SAML）を推奨しますが、その場合はシステムへのアクセスがすべてユーザーインターフェース（UI）層にある MFA を通る必要があります。そのため、SSH の設定を変更すると、UI 経由の MFA も設定を変更する必要があります。

REST API

SSH で MFA を有効にするには、以下の REST API を使用します。

- API
  `Administration > GET/devmgr/v2/ssh/enable-ssh-mfa`
- GET の戻りデータ

```
{
  "mfaEnablement": [{
      "controllerRef": "＜controllerRef＞",  // 本設定が属するcontrollerRef
      "mfaEnabled": true    //  「true」または「false」
     }],
  "authorizedKeys": "＜認証鍵の内容＞"
}
```

- API
  `Administration > POST/devmgr/v2/ssh/enable-ssh-mfa`
- POST の本文

```
{
  "mfaEnablement": [{
       "controllerRef": "＜controllerRef＞",    // 本設定が属するcontrollerRef
       "mfaEnabled": true    //  「true」または「false」
     }],
  "authorizedKeys": "＜認証鍵の内容＞"
}
```

mfaEnabled パラメータの値が true に設定されている場合は、SSH 鍵とパスワードを使用してシステムにログインします。なお、設定項目は数多く存在しますが、設定はコントローラごとに行うため、両コントローラで設定を合わせるために、設定対象コントローラを指定する必要があります。

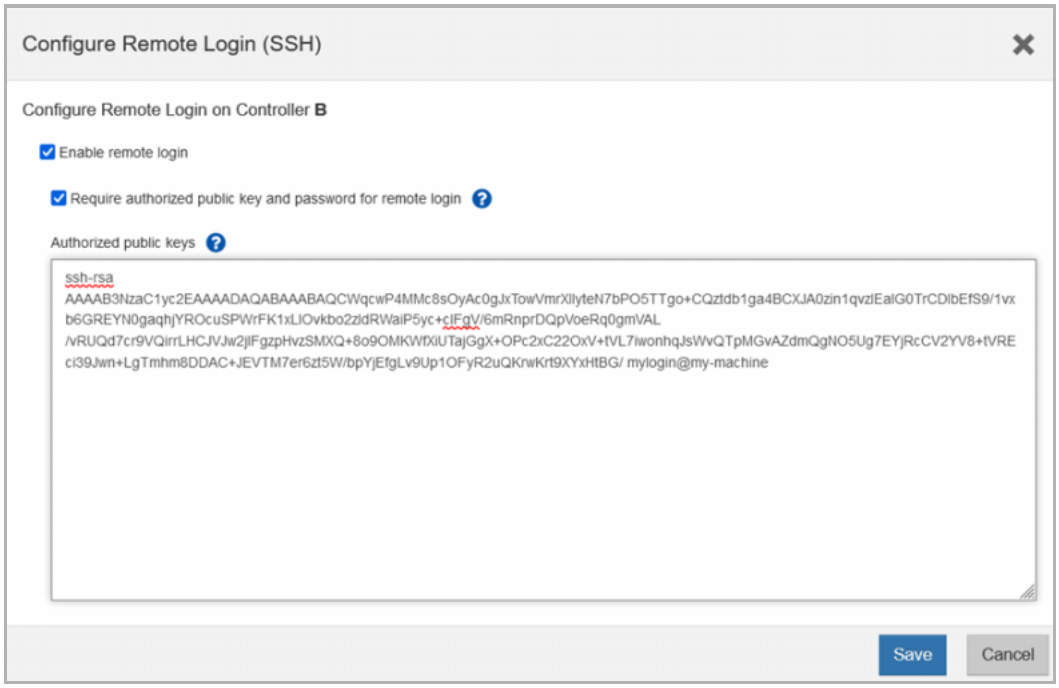
System Manager

System Manager には、システム管理者による SSH 鍵の管理と MFA 設定の有効化操作が可能なダイアログボックス（図 3.1）が含まれています。

備考

[Authorized Public Keys] テキストボックスには、SSH 鍵を複数入力できます。

図 3.1　　パスワード付き SSH の有効化



3.3　SSH サーバー鍵の再生成

セキュリティ対策のため、SSH サーバー鍵を定期的に再生成する場合があるかと思います。このような操作を実行するための機能は REST API インターフェースを通じて公開されており、System Manager には存在しません。

SSH サーバー鍵の再生成後、ストレージアレイに接続するクライアントは、自身のところにキャッシュ済みのサーバー鍵をリセットするよう求められます。

SSH サーバー鍵を再生成するには、以下の API を使用します。

- API
  `Administration > POST/devmgr/v2/ssh/regenerate-server-keys`

POST の本文と API パラメータは、存在しません。
鍵の生成に成功した場合の応答ステータスコードは、200 です。

注意

SSH サーバー鍵の再生成時は、上記 API を両コントローラーに対して呼び出し、双方の SSH サーバー鍵を再生成する必要があります。

3.4 コマンドラインアクセス

ETERNUS AB/HB series ストレージシステムには、複数の安全なコマンドラインアクセスポイントがあります。

3.4.1 SANtricity Web Services REST API

プロキシがインストールされているホストで Web ブラウザを使用して REST API にアクセスするには、以下に進みます。

```
https://localhost/devmgr/docs/#/
```

REST API に初めてアクセスする場合は、各タイプのブラウザに次の情報が表示されます。

- Chrome に「この接続ではプライバシーが保護されません」と表示される場合。[詳細設定] をクリックして Web サイトに進みます。
- Internet Explorer で、「この Web サイトのセキュリティ証明書には問題があります」と表示される場合。「このサイトの閲覧を続行する（推奨されません）」をクリックして、Web サイトに進みます。
- Firefox で 「安全な接続ができませんでした」と表示される場合。[詳細] ボタンをクリックし、証明書の例外を追加して Web サイトに進みます。

注意

セキュリティ上の理由から、REST API の HTTP Basic アクセス認証を無効にすることを推奨します。

Basic 認証を無効にするには、以下の REST API を実行します。

- API
  `Administration > POST /storage-systems/{system-id}/settings/authentication`
- URL パラメータ
  `system-id`
- POST の本文

```
{
  "disableBasicAuthentication": true    // 「true」または「false」
}
```

3.4.2 Secure CLI

セキュアな SMcli を使用すると、SMcli クライアントは、安全な HTTPS チャネルを介してストレージシステムと通信できます。従来の SANtricity SMcli 文法とコマンドセマンティクスを使用し、安全なプロトコルでストレージシステムと相互運用できる HTTPS シンクライアントを提供します。

3.4.3 CLI セッションのタイムアウト

デフォルトの CLI セッションタイムアウトは 30 分です。タイムアウトは、古いセッションとセッションのピギーバックを防ぐために重要です。

3.4.4 レガシー管理インターフェース

ストレージ管理用に新しい REST API が導入されてはいますが、特定の外部管理ツールが機能しないのを防ぐために、独自のレガシー管理インターフェース (SYMbol、ポート 2463) をデフォルトで使用できるように出荷時設定しています。

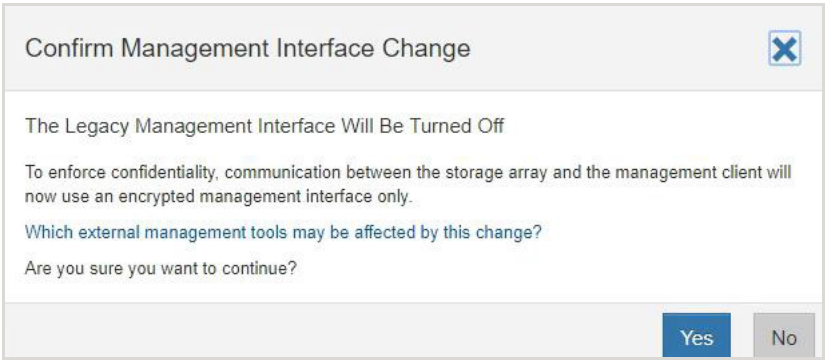
SANtricity SMI-S Provider や OnCommand Insight など、従来の管理インターフェースと直接通信するツールは、従来の管理インターフェース設定が使用可能になっていないと機能しません。また、この設定が無効になっている場合、従来の CLI コマンドを使用したり、ミラーリング操作を実行したりすることはできません。

今後リリースされる SANtricity では、レガシー管理インターフェースをデフォルトで使用できる設定が削除される予定です。

注意

影響を受ける外部管理ツールを使用していない場合は、従来の管理インターフェースを無効にして、アレイを安全なインターフェースに変更することを推奨します。その前に、両方のストレージアレイコントローラーに適切な認証局 (CA) ルート証明書、中間証明書、および署名済みサーバー証明書をインストールする必要があります。インストールが完了したら、図 3.2 に示すように、System Manager (Settings > System > Additional Settings > Change Management Interface) を使用してアレイ管理インターフェースをセキュアモードに変更します。

図 3.2 System Manager のレガシー管理インターフェース無効化ウィンドウ



3.4.5 JSON Web Token アクセス

SAML（MFA）が有効なシステムでは、従来の自動化ツールの利用がデフォルトではできません。MFA のワークフローでは、認証にあたり ID プロバイダーサーバーへのリダイレクトが必要なため、REST API と Secure CLI は事実上操作できなくなります。以前は、ID プロバイダーサーバーの役割を十分に果たし、かつ SAML が有効なエンティティは、System Manager のみでした。

SAML 有効時に JSON Web Token（JWT）アクセスを使用すると、MFA のような認証が行えます。SAML が有効な場合は System Manager からトークンの生成を行う必要があるため、ユーザーが MFA 経由で認証されていることが必須になります。

トークンは、（LDAP ユーザーを含む）特定ユーザー、一連の権限、有効期間と関連づくためのプロパティがあります。セキュリティ管理者のほうで、発行したトークンの有効期間の最大値を設定できます。有効期間がこの最大値を超えるようなトークンは生成できません。トークンは、生成されてユーザーに提供されると、REST API または Secure CLI を経由してシステムにアクセスするのに使用できます。トークンにはパスワードがないため、管理は慎重に行う必要があります。アクセストークンの有効期間が切れると、認証しようとしても失敗します。

備考

JSON Web Token の使用にあたり、SAML の有効化は必要ではないですが、セキュリティを最大限に高めるために推奨します。

トークンの失効化

セキュリティ管理者のほうで、あるトークンが危殆化していると判断した場合、該当トークンの署名に使用する鍵を再生成できます。鍵がリセットされると、発行済みトークンはすべて直ちに無効になります。新規生成するトークンは、有効期間を可能な限り短くして、危殆化のおそれを減らすことを推奨します。トークンは、ストレージアレイ内では管理されないため、個別に失効させることはできません。トークンの管理は、REST API または System Manager から行えます。

REST API

SAML が有効でない場合、トークンの生成および管理は REST API から行えます。利用可能な API 呼び出しは、以下のとおりです。

トークンを新規発行するには、以下の REST API を使用します。

- API
  `Authentication > POST/devmgr/v2/access-token`
- POST の本文

```
{
  "duration": ＜トークンの有効期間（秒数）＞
}
```

- POST の戻り値

```
{
  "accessToken": "＜トークンの値（文字列）＞",
  "duration": ＜トークンの有効期間（秒数）＞
}
```

生成済みトークンをすべて利用できなくするには、以下の REST API を使用します。

- API
  `Authentication > DELETE/devmgr/v2/access-token`

本文と戻り値は存在しません。本 API を呼び出すと、鍵が新しい値に再設定されます。以前に生成済みのトークンはすべて直ちに無効になります。

トークン生成パラメータを設定するには、以下の REST API を使用します。

- API
  `Authentication > POST/devmgr/v2/access-token/settings`
- POST の本文

```
{
  "maxDuration": ＜トークンの有効期間の最大値（秒）＞
}
```

レスポンスボディは存在しません。

トークンの使用方法について

- REST API で使用する場合
  トークンを REST API のリクエストで使用するには、以下のとおりリクエストに HTTP ヘッダを追加します。
  `Authorization: Bearer < アクセストークンの値 >`
- Secure CLI で使用する場合
  Web トークンの使用には、以下の 2 つのパラメーターが利用できます。
  - `-t ＜アクセストークンの値＞`（トークンの値をコマンドライン上で指定）
  - `-T ＜アクセストークンのファイル＞`（トークンの値が格納されたファイルのパスを指定）

  上記パラメータは互いに排他関係にあるため、指定する際はどちらか一方のみを指定します。アクセストークンとユーザー名／パスワードを両方指定することもありません。

> **備考**
>
> ユーザー名／パスワードとトークンをどちらも指定しなかった場合、コマンドライン上でアクセストークンの値の入力が求められます。

- System Manager から使用する場合
  System Manager の UI からトークンを新規生成できます（図 3.3）。トークンの有効期間の日数を選択するよう求められます（図 3.4）。新規トークンは、現在ログイン中のユーザーに向けて生成されます。生成されたトークンは、ユーザーの権限が内部にエンコードされ、選択した有効期間の日数だけ有効になります。トークン生成後、新規トークンの文字列がダイアログボックス内に表示されるので、コピーして（図 3.5）安全な場所に保管する必要があります。ダイアログボックスを閉じると、以降はトークンの値を確認する機会がありません。アクセストークンは失効させることもできます（図 3.6）。

図 3.3　アクセストークンの作成

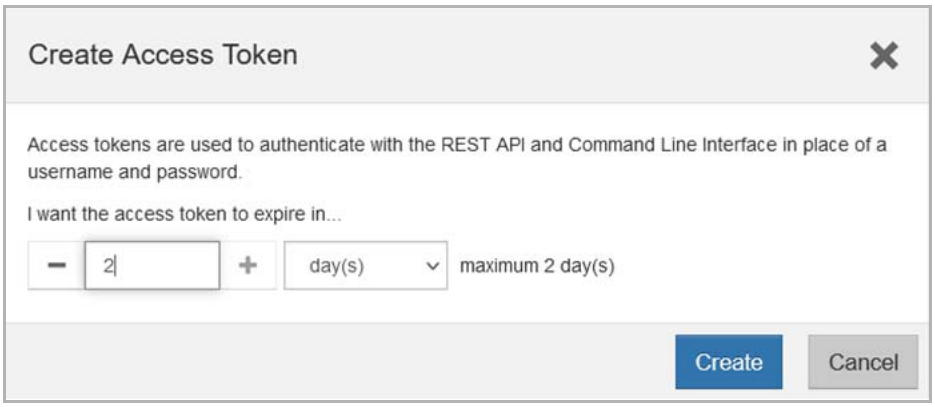


図 3.4　トークンの有効期間の変更

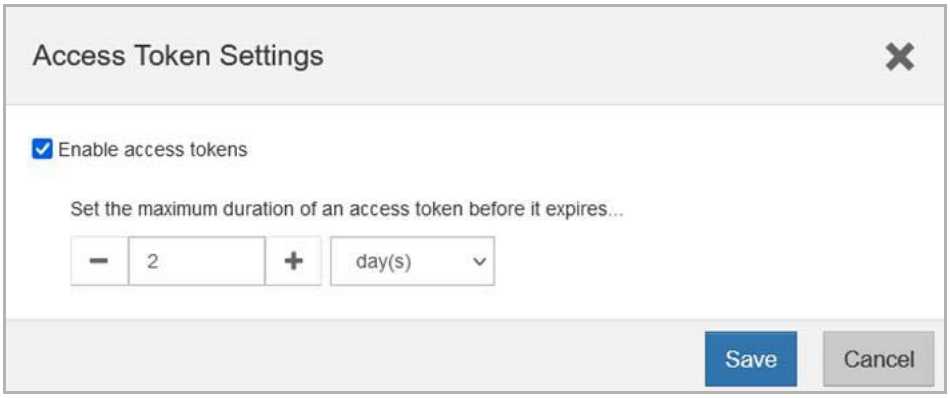


図 3.5 アクセストークンのコピー

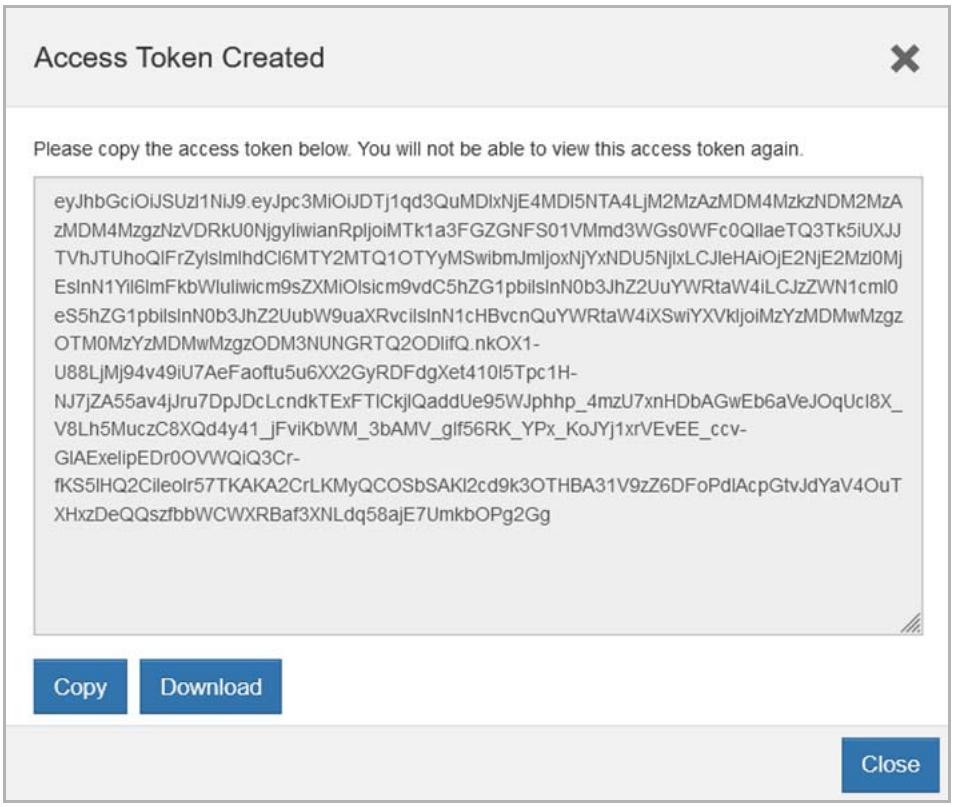
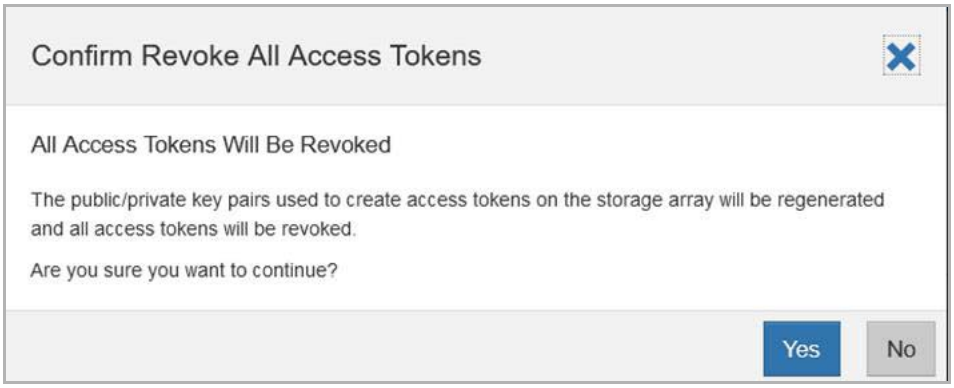


図 3.6 アクセストークンの失効化



3.5 Web アクセス

3.5.1 SANtricity System Manager

SANtricity 管理者が、ストレージシステムへのアクセスと管理に CLI の代わりにグラフィカルインターフェースを使用する場合は、SANtricity System Manager を使用する必要があります。これは、Web サービスとして SANtricity に含まれており、デフォルトで有効になっており、ブラウザを使用してアクセスできます。ブラウザ (https://) にホスト名 (DNS を使用する場合) または IPv4 か IPv6 アドレスを指定します。

コントローラが自己署名デジタル証明書を使用している場合は、証明書が信頼されていないことを示す警告がブラウザに表示されることがあります。アクセスを続行するリスクを認識するか、CA 署名のデジタル証明書をインストールできます。

注意

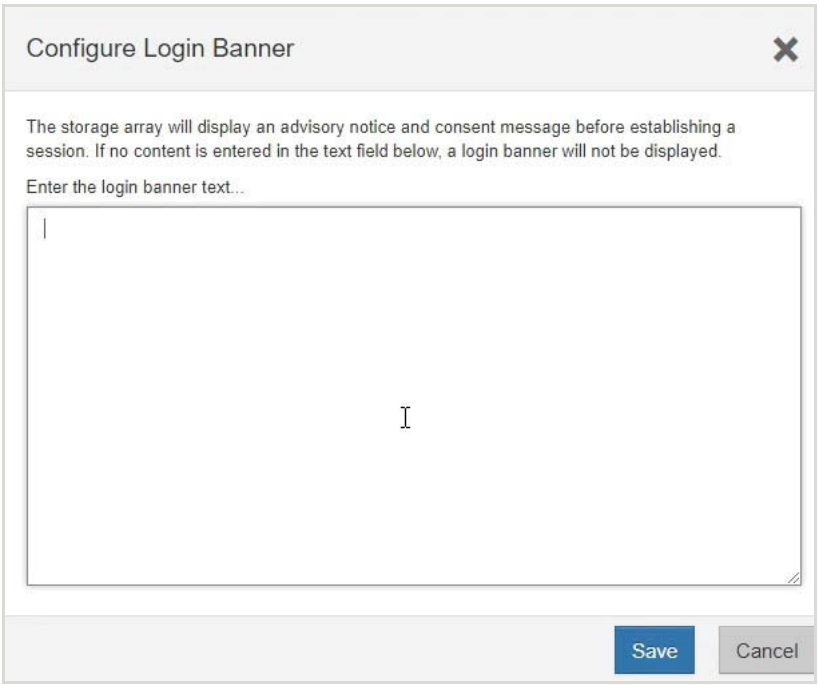
セキュリティ上の理由から、サーバー認証用の CA 署名デジタル証明書をストレージシステムにインストールすることを推奨します。

SANtricity System Manager には、Security Assertion Markup Language (SAML) 認証を使用できます。

3.5.2 ログインバナー

ログインバナーを使用することで、任意のオペレータ、管理者、および不正行為者に対しても、許容される使用条件を提示し、システムへのアクセスを許可されているユーザーを示すことができます。このアプローチは、システムへのアクセスと使用に対する予測を立てる一助になります。図 3.7 は、ログインバナーの設定方法に関する System Manager ウィンドウ (Settings > System > Configure Login Banner) を示しています。

図 3.7 [System Manager] ウィンドウでのログインバナーの設定方法



3.5.3 SANtricity System Manager の SAML 認証

SAML 2.0 は、複数のシステム間で安全に認証要求とユーザーデータを送信するための業界標準です。この標準では、多くのアプリケーションが 1 つのサービスを使用して、すべてのユーザー認証とセッションを管理できます。

多要素認証 (MFA) では、認証に成功するために、ユーザーが本人であることの証明として 2 つ以上の項目を提供する必要があります。認証に必要な項目で典型的なものは知識 (パスワードなどユーザーが記憶しているもの)、所有物 (コード変更をする機械などユーザーが所持しているもの)、または個人のもの (指紋などの生体認証をはじめとするユーザーに固有のもの) の内の、少なくとも二種類です。必要な証拠の具体的なタイプは、エンドユーザー組織のセキュリティチームが設定します。

SAML は ETERNUS AB/HB series 製品に統合され、複数の形式の認証を使用してユーザーを認証し、認証の成功または失敗を SANtricity System Manager アプリケーションに報告できる外部システムと通信できます。外部システムは、単一要素認証、二要素認証、または多要素認証を使用するように構成できます。外部システムは、他のアプリケーションとのシングルサインオン機能をサポートする機能も提供します。

SAML がストレージシステムで構成された後は、構成されたアイデンティティプロバイダ (IdP) 経由でのみ SANtricity System Manager にログインできます。ユーザーが SANtricity System Manager にアクセスしようとすると、デフォルトの SANtricity System Manager ログインページではなく、IdP のログインページにアクセスされます。認証情報を入力すると、ユーザーは認証されたセッションとともに SANtricity System Manager に送り返され、ユーザーに関連付けられた属性に基づいて認証されます。

SAML が使用可能になっている場合、SAML は、SANtricity System Manager にアクセスするユーザーの認証に使用される唯一の方法です。他の形式の管理は、認証できないために機能しなくなりました。

これには、EMW、SMcli クライアント、ソフトウェア開発キットクライアント、UTM を使用したインバンド管理、HTTP 基本認証を使用した REST API クライアント、および標準ログインエンドポイントを使用した REST API クライアントが含まれます。

注意

セキュリティを強化するために MFA を構成することを推奨します。

3.6 SNMP 監視

SANtricity は、電子メール、SNMP トラップ、および syslog を介して送信されるアラート通知をサポートします。アラートは、ストレージアレイで発生した重要なイベントを管理者に通知します。SANtricity OS 11.70.2 以降では SNMPv3 をサポートし、アラート通知の認証と暗号化を行います。当社では、SHA-256 または SHA-512 のどちらかの認証プロトコルとプライバシープロトコル AES-128 を使用して SNMP USM ユーザーを設定することを推奨します。

11.70.2 よりも前のリリースでは、SANtricity は認証と暗号化をサポートしない SNMPv2c のみをサポートします。

注意

- セキュリティ上の理由から、SNMPv2c を設定しないことを推奨します。
- SNMPv2c が必要な場合、セキュリティを確保するために SNMPv2c コミュニティストリングを設定することを推奨します。SNMP コミュニティストリングは、デバイスの統計情報へのアクセスを許可するユーザー ID またはパスワードのようなものです。

4. ストレージ管理システムの監査

4.1 Syslog の送信

ログおよび監査情報は、サポートおよび可用性の観点から、組織にとって非常に重要です。さらに、ログ (syslog)、監査レポート、および出力に含まれる情報と詳細は、一般に機密性が高くなります。セキュリティ管理とセキュリティ体制を維持するには、ログデータと監査データを安全に管理することが不可欠です。

syslog 情報のオフロードは、侵害の範囲または影響を単一のシステムまたはソリューションに制限するために必要です。

注意

syslog 情報を安全なストレージまたは保存場所に安全にオフロードすることを推奨します。

セキュアな監査ログチャネルの構成は、以下のセクションで説明するように、SANtricity System Manager UI または REST API 呼び出しのいずれかを介して行われます。監査ログ用に syslog を設定する方法の詳細については、SANtricity System Manager のオンラインヘルプトピック「Configure Syslog Server for Audit Logs」を参照してください。

4.1.1 SANtricity System Manager UI を使用した監査ログ用の syslog 設定

SANtricity System Manager UI を使用して監査ログ用に syslog を構成するには、以下の手順を完了します。

1. Settings > Access Management を選択します。
2. [Audit Log] タブから、[Configure Syslog Servers] を選択します。
   [Configure Syslog Servers] ダイアログボックスが表示されます。
3. [Add] をクリックします。
   [Add Syslog Server] ダイアログボックスが表示されます。
4. サーバー情報を入力し、[Add] をクリックします。
   - サーバーアドレス
     完全修飾ドメイン名、IPv4 アドレス、または IPv6 アドレスを入力します。
   - プロトコル
     ドロップダウンメニューからプロトコルを選択します (TLS、UDP、TCP など)。
   - 証明書のアップロード (オプション)
     TLS プロトコルを選択し、署名された CA 証明書をまだアップロードしていない場合は、[Browse] をクリックして証明書ファイルをアップロードします。信頼できる証明書がないと、監査ログは syslog サーバーにアーカイブされません。

     注意

     後で証明書が無効になった場合、TLS ハンドシェイクは失敗します。その結果、監査ログにエラーメッセージが出力され、syslog サーバーには送信されなくなります。
     この問題を解決するには、syslog サーバーで証明書を修正してから、Settings > Audit Log > Configure Syslog Servers > Test All に移動します。

- ポート
  syslog レシーバのポート番号を入力します。
  Add をクリックすると、Configure Syslog Servers ダイアログボックスが開き、設定した syslog サーバーがページに表示されます。

5 ストレージアレイとのサーバー接続をテストするには、[Test All] を選択します。

4.1.2 REST API を使用した監査ログ用の Syslog の設定

REST API を使用して監査ログ用に syslog を設定するには、次のコマンドを実行します。

- API
  - `POST /storage-systems/{system-id}/syslog`
  - `POST /storage-systems/{system-id}/syslog/{id}`（特定の syslog サーバーの更新に使用）
- URL パラメータ
  - `system-id`
  - `id`（更新対象の syslog サーバーを指定）
- POST の本文（追加／更新する syslog サーバー構成を記載）

```
{
  "serverAddress": "string", // 完全修飾ドメイン名またはIPアドレス
  "port": 6514,              // syslogサーバーが待ち受けているポート番号
  "protocol": "tls",         // 「udp」「tcp」「tls」のいずれか
  "components": [
      {
          "type": "auditLog" // 「auditLog」のみ指定可能
      }]
}
```

注意

接続および通信チャネルを保護するために、TLS プロトコルタイプの使用を推奨します。したがって、監査ログメッセージを受信する syslog サーバーも、TLS プロトコル (TLS バージョン 1.2 以上) をサポートするように設定する必要があります。

5. ストレージ暗号化

保存データの暗号化は、ディスクが盗まれたり、返却されたり、転用されたりした場合に、機密データを保護するために重要です。ETERNUS AB/HB series ストレージシステムは、自己暗号化ドライブを使用して、保存データを暗号化します。これらのドライブは、フルディスク暗号化機能が有効かどうかに関係なく、書き込み操作ではデータを暗号化し、読み取り操作ではデータを復号化します。SANtricity 機能が有効になっていない場合、データはメディア上で暗号化されますが、読み取り要求時には自動的に復号化されます。

ストレージシステムでフルディスク暗号化機能が有効になっている場合、ストレージシステムが正しいセキュリティキーまたは認証キーを提供しない限り、ドライブは読み取りまたは書き込み操作からドライブをロックすることによって、保存データを保護します。このプロセスにより、適切なセキュリティキーファイルをインポートしてドライブのロックを解除しない限り、別のストレージシステムからデータにアクセスできなくなります。また、サードパーティ製のユーティリティや OS からもデータにアクセスできなくなります。

フルディスク暗号化機能をさらに拡張し、Key Management Interoperability Protocol (KMIP) 標準に準拠した Gemalto SafeNet KeySecure Enterprise Encryption Key Management のような集中型鍵管理システムを介してフルディスク暗号化セキュリティキーを管理できるようにしました。この機能は ETERNUS AB2100/AB3100/AB5100/AB6100 および ETERNUS HB2x00/HB5x00 で使用できます。

ドライブ内のハードウェアによって実行される暗号化および復号化操作は、ユーザーには見えず、パフォーマンスやユーザーのワークフローには影響しません。各ドライブには固有の暗号化キーがあり、ドライブから転送、コピー、または読み取ることはできません。暗号化キーは、National Institute of Standards and Technology (NIST) AES で指定されている 256 ビットキーです。一部だけでなく、ドライブ全体が暗号化されます。

注意

集中型鍵管理システムを使用してドライブのセキュリティ機能を有効にすることを推奨します。

6. 転送中データのセキュリティ

状況によっては、ETERNUS AB/HB series ストレージシステムにネットワーク経由で転送されるクライアントデータをすべて保護する要件がある場合があります。すなわち、転送中の機密データを暗号化することで中間者攻撃から保護するということです。現在のところ、ETERNUS AB/HB series ストレージシステムのホストインターフェースは、転送中データの暗号化をネイティブでサポートしていません。転送中データの暗号化要件に準拠するには、転送中データの暗号化をより高いレベルで実装することを推奨します。実装すると、ETERNUS AB/HB series ストレージシステムへの書き込み処理の前にデータが暗号化され、読み取り処理中も暗号化されたままになります。ETERNUS AB/HB series ストレージシステムのソフトウェアでは暗号化データの復号化や読み取りができないため、ホストレベルの暗号化が有利です。コマンド処理のために復号化が必要なため装置ではデータが平文で見えてしまうネットワークレベルの暗号化とは異なり、データの安全性が維持されます。攻撃者がネットワーク上でファイルシステムのデータを傍受しても、データはホスト側で暗号化されているため保護されます。これを実現する 2 つの方法は、暗号化ファイルシステムまたはデータベースレベルの暗号化を使用することです。

6.1 暗号化ファイルシステム

ETERNUS AB/HB series ストレージシステムからマッピングされたディスク上で暗号化ファイルシステムを使用する方法です。BitLocker、eCryptfs、gocryptfs などの暗号化ファイルシステムでは、AES などの NIST 承認の鍵暗号アルゴリズムを使用してファイルを暗号化してからディスクに書き込み、読み取り時に復号化します。ファイルシステムレベルの暗号化を行うと、暗号化および復号化の処理のために CPU 負荷が増加するため、ホスト側でパフォーマンスのオーバーヘッドが発生する可能性があります。

6.2 データベース暗号化

ETERNUS AB/HB series ストレージシステムを使用してデータベースのデータを保存する場合は、データベースに組み込まれている暗号化機能を利用できます。MySQL、PostgreSQL、Oracle、SQL Server などの最新のデータベースでは、AES などの NIST 承認のキー暗号アルゴリズムを使用する Transparent Data Encryption（TDE）がサポートされているものがほとんどです。

6.3 ホストインターフェイスの相互認証

相互認証を行うと、通信対象の両者は互いの ID を確認してから接続を確立することが求められるため、攻撃者がオープンなインターフェイスへの接続を確立するのを防げます。ETERNUS AB/HB series ストレージシステムでは、相互認証の実装に、iSCSI ポート用または iSER over InfiniBand ポート用の Challenge Handshake Authentication Protocol（CHAP）を使用できます。CHAP は、初期リンク設定時に構成され、ホストがストレージアレイに対して自身を検証し、ストレージアレイがホストに対して自身を検証できるようにします。ストレージアレイ（ターゲット）とホスト（イニシエータ）で CHAP を構成するには、次の手順に従います。

System Manager

System Manager で CHAP 認証を構成するには、次の手順を実行します。

注意

手順の実行前に、CHAP 認証用に構成するホストをストレージアレイ上で定義する必要があります。

1. [Settings] > [System] > [Configure Authentication] と選択します。
2. [Two-way (mutual) authentication] を選択し、[Next] をクリックします。
3. ターゲットの CHAP シークレットを設定して確認し、[Next] をクリックします。

   注意

   ターゲットにアクセスしようとするイニシエータは、ターゲットの CHAP シークレットを提供する必要があります。CHAP シークレットの長さは、12 ～ 57 文字である必要があります。

4. 相互 CHAP 認証を構成する各ホストについてイニシエータのシークレットを入力し、[Finish] をクリックします。

   注意

   本手順を行うことで、各ホストで構成済みの、または構成予定の CHAP シークレットがストレージアレイに提供されます。

ホスト側での iSCSI CHAP の設定

スタンドアロン Linux ホストで CHAP 認証を設定するには、次の手順を実行します。

1. ホストの /etc/iscsi/iscsid.conf ファイルの「CHAP settings」セクションを編集して、次の情報が書き込まれるようにします。フィールド値はすべて、構成に合わせて記入します。

```
node.session.auth.authmethod = CHAP
node.session.auth.username = ＜ホストのiSCSI 修飾名（IQN）＞
node.session.auth.password = ＜ターゲット（ストレージ）のシークレット＞
node.session.auth.username_in = ＜ホストのiSCSI 修飾名（IQN）＞
node.session.auth.password_in = ＜イニシエータ（ホスト）のシークレット＞
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = ＜ホストのiSCSI 修飾名（IQN）＞
discovery.sendtargets.auth.password = ＜ターゲット（ストレージ）のシークレット＞
discovery.sendtargets.auth.username_in = ＜ホストのiSCSI 修飾名（IQN）＞
discovery.sendtargets.auth.password_in = ＜イニシエータ（ホスト）のシークレット＞
```

2. iSCSI サービスを再起動して、変更を適用します。

```
sudo systemctl restart iscsid
sudo systemctl restart iscsi
```

3. このホストから検出および開始された iSCSI セッションはすべて、相互に認証されます。

スタンドアロンの Windows ホストで CHAP 認証を設定するには、次の手順を実行します。

1. ホストのターミナルウィンドウから [Execute] > `iscsicpl.exe` を選択して、**iSCSI Initiator Properties** ダイアログボックスを開きます。
2. **Configuration** タブで **CHAP** を選択し、ホストの CHAP シークレットを設定します。
3. **Discovery** タブで、**Discover Portal** を選択します。iSCSI ターゲットポートのうちいずれか 1 つの IP アドレスを入力し、**Advanced** を選択します。
4. **Advanced Settings** で、次のように設定します。
   - 4-1 ストレージアレイへの接続に使用する **Initiator IP** を選択します。
   - 4-2 [Enable CHAP log on] ボックスをオンにします。
   - 4-3 ターゲットとなるストレージアレイのターゲットシークレットを入力します。
   - 4-4 [Perform mutual authentication] チェックボックスをオンにします。
   - 4-5 [OK] をクリックします。
5. **OK** を再度クリックして検出を完了します。
6. 手順 1 ～手順 5 をすべての iSCSI 接続に対して繰り返します。これらのターゲットポータルから開始された接続ではすべて、相互認証が実行されます。

7. SSL と TLS の管理

Secure Sockets Layer (SSL) は、インターネット接続を安全に保つための標準技術です。暗号化アルゴリズムを使用して転送中のデータをスクランブルすることにより、2 つのシステム間で送信される機密データを保護することで、攻撃者によるデータの読み取りを防止します。

Transport Layer Security(TLS) は、SSL の更新された、よりセキュアなバージョンです。この 2 つの用語 (SSL と TLS) は、業界では同じ意味で使用されます。

SANtricity は、TLS を使用して次の通信チャネルを保護します。

- SANtricity System Manager Web クライアントとストレージシステムで実行されている Web サーバー間
- ストレージシステムで実行されている LDAP クライアントと LDAP/AD サーバー間
- ホストで実行されている Unified Manager/Web サーバープロキシとストレージシステムで実行されている Web サーバー間
- ストレージシステムと SAML アイデンティティプロバイダ間
- ストレージシステムと syslog サーバー間
- ストレージシステムで実行されている自己暗号化ドライブ (SED/FDE) の Lock Key Manager とサードパーティ製の外部キーマネージャー間

注意

- SANtricity OS 11.70.3 未満では　TLS 1.2 および TLS 1.3 をサポートしています。
- 証明書チェーン全体の有効期限をチェックするために、厳密な証明書検証を有効にすることを推奨します。デフォルトでは無効になっており、サーバー証明書の有効期限のみがチェックされます。

7.1 TLS 1.3 のサポート

SANtricity OS 11.70.3 からは、TLS 1.3 もサポートされています。これは、TLS 1.2 よりも安全で高速であると一般的に考えられているためです。コモンクライテリアなど、TLS 1.3 の評価未完了のため TLS 1.3 の有効化を許容していないセキュリティ要件があります。SANtricity では、TLS 1.3 を無効にして任意のセキュリティ要件を満たすことができます。この設定は、REST API を使用して変更できます。

TLS 1.3 を無効にするには、以下の REST API を使用します。

- API
  `Administration > POST /settings`
- POST の本文

```
{
  "serverSettings":
  {
    "tls13Disabled": true    // 「true」または「false」
  }
}
```

設定の変更は、両方のコントローラに反映されます。TLS 1.3 を無効にした後、TLS/SSL 設定を更新するために両方のコントローラの Web サーバーを再起動する必要があります。デュプレックスシステムでは、両方のコントローラの Web サーバーを再起動する必要があります。

コントローラの Web サーバーを再起動するには、以下の REST API を使用します。

備考

本リクエストでは、リクエスト送信先のコントローラのみが再起動され、応答は返りません。

- API
  `Administration > POST /restart`

Web サーバーを再起動すると、現在サポートされている TLS プロトコルと暗号が取得できます。

サポートされている TLS/SSL プロトコルと暗号を Web サーバーに取得するには、以下の REST API を使用します。

注意

デュプレックスシステムでは、リクエストを両方のコントローラに送信する必要があります。

- API
  `Administration > GET /settings/tls-params`
- GET の戻りデータの例

```
{
  "protocols": [
    "TLSv1.3",
    "TLSv1.2"
  ],
  "ciphers": [
    "TLS_AES_256_GCM_SHA384",
    "TLS_AES_128_GCM_SHA256",
    "TLS_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
    "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
    "TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
    "TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
    "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384",
    "TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_DHE_RSA_WITH_AES_128_GCM_SHA256"
  ]
}
```

8. 外部鍵管理サーバー

8.1 外部鍵管理サーバーの既定の鍵サイズ変更

外部鍵管理サーバーのセットアップの一部に、（ストレージアレイから生成された）クライアント証明書の CSR 生成があります。証明書の鍵サイズは 3072 ビット（以前は 2048 ビット）です。鍵サイズは、管理インターフェイス（UI）からは選択できません。ユーザーが証明書の鍵サイズを変更するには、NVSRAM の変更が必要です。NVSRAM のユーザー設定可能領域を変更し、鍵サイズの値を大きくしたり小さくしたりすることができます。

8.1.1 SMcli を使用した外部鍵管理サーバーの鍵サイズ設定

選択可能な鍵サイズを表 8.1 に示します。

表 8.1 外部鍵管理サーバーの鍵サイズ

値	鍵サイズ
0（デフォルト）	3027
1	2048
2	3027
3	4096

SMcli を使用して外部鍵管理サーバーの鍵サイズを設定するには、以下のコマンドを実行します。

```
set controller[a] globalNVSRAMByte[0xc0]=3 // 外部鍵管理サーバーの鍵サイズを4096に設定
set controller[b] globalNVSRAMByte[0xc0]=3 // 外部鍵管理サーバーの鍵サイズを4096に設定
show allControllers globalNVSRAMByte[0xc0] // 変更後の値をダンプ
```

8.1.2 REST API を使用した外部鍵管理サーバーの鍵サイズ設定

REST API を使用して外部鍵管理サーバーの鍵サイズを設定するには、以下の API を使用します。

- API
  `POST /storage-systems/1/symbol/setControllerNVSRAM`
- POST の本文

```
{
  "regionId": "userConfig2Data",
  "offset": 32,
  "regionData": "3"                  // 鍵サイズを4096に設定
}
```

8.2 外部証明書署名要求ワークフロー

SANtricity OS 11.90 の時点では、外部鍵管理サーバーの証明書機能は、すべてのアーティファクトを外部で生成し、すべての証明書を一度にインポートすることをサポートしています（図 8.1）。このため、秘密鍵と署名付き証明書をセキュリティポリシーに準拠するよう頻繁に変更できるような、より高度な鍵ローテーション計画を採用できます。

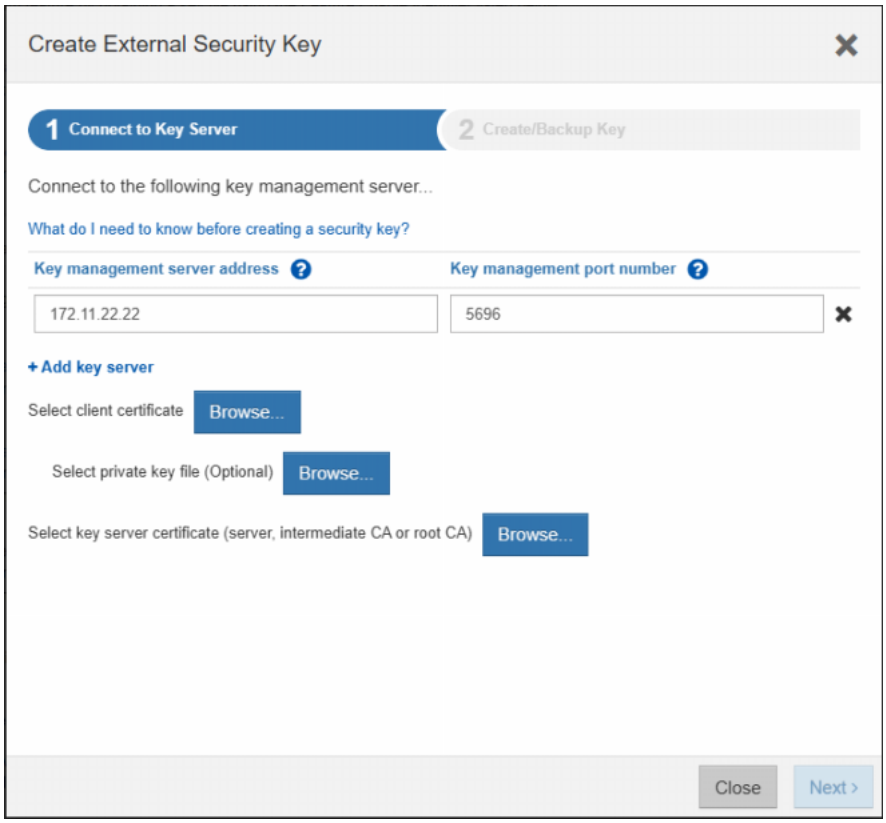
備考

従来の内部 CSR ワークフローも引き続き適用可能です。CSR は生成後、新しい署名付き証明書の取得のため外部鍵管理サーバーに再送信されます。署名付き証明書は、秘密鍵が変更されていないため、鍵管理サーバーとの通信損失のリスクにさらされることなく自由に再インポートできます。

ワークフローを実行する手順は、次のとおりです。

1. リーフ証明書の公開鍵と秘密鍵を生成します。
2. リーフ秘密鍵によって署名されたリーフ証明書署名要求を生成します。
3. 生成された CSR に基づいて鍵管理サーバーから署名付き証明書を取得します。証明書は、外部鍵管理サーバーの秘密鍵によって署名されます。
4. 鍵管理サーバーの信頼された証明書を取得します。
5. 鍵管理サーバーの信頼された証明書、新しい署名付き証明書、およびリーフ証明書の秘密鍵をアレイにインポートします。

図 8.1 鍵管理サーバー証明書（秘密鍵を含む）のインポート



鍵管理サーバー証明書のインポートは、CLI でもサポートされています。なお、`privateKeyFile` パラメータの指定は任意です。

```
download storageArray keyManagementClientCertificate certificateType=(client|server)
file="＜証明書のファイル名＞" [privateKeyFile = "＜秘密鍵のファイル名＞"]
```

9. Online Certificate Status Protocol

Online Certificate Status Protocol (OCSP) を使用すると、LDAP over TLS などの TLS 通信を使用する SANtricity アプリケーションは、OCSP が有効な場合にデジタル証明書のステータスを受信できます。アプリケーションは、要求された証明書が有効であるか、失効しているか、または不明であることを示す署名付き応答を受信します。

OCSP では、証明書失効リスト (CRL) を必要とせずに、デジタル証明書の現在の状態を判断できます。

既定では、OCSP 証明書の状態チェックは無効になっています。REST API で有効にできます。

注意

ご使用の環境に OCSP サーバーがある場合、OCSP を構成することを推奨します。

OCSP 証明書の状況判断を有効にするには、以下の REST API を使用します。

- API
  `Administration > POST /certificates/settings`
- POST の本文

```
{
  "revocationChecking": true,
  "ocspResponderAddress": "<文字列>",    // OCSPレスポンダーの名前またはIPアドレス
  "strictCertVerificationEnabled": true
}
```

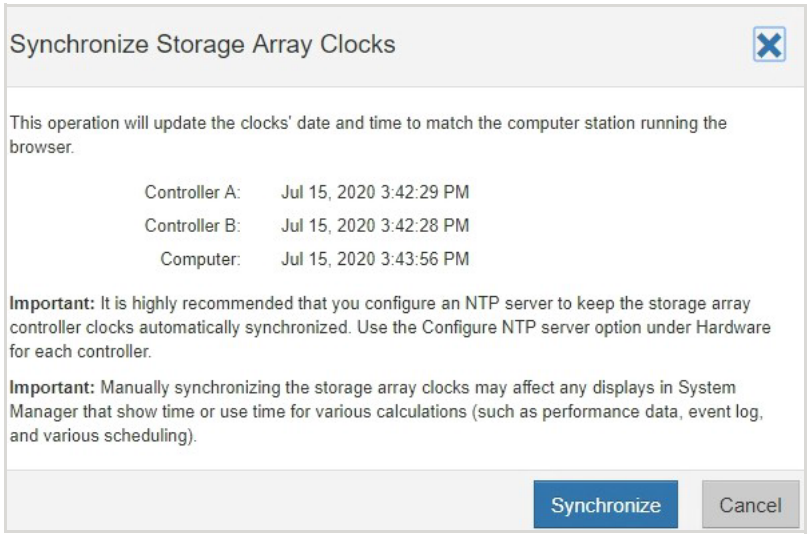
10. ネットワークタイムプロトコル

SANtricity System Manager (Settings > System > Synchronize Storage Array Clocks) では、ブラウザを実行しているコンピュータステーション (図 10.1 参照) と時刻を同期させることにより、ストレージシステムのタイムゾーンおよび日付時刻を手動で設定できます。NTP もサポートしていますが、セキュア NTP はサポートしていません。パケットが認証用に暗号で署名される機能を持たないと、NTP サーバーは中間者攻撃を受けやすくなります。

注意

セキュリティを優先する場合は、ストレージシステムのタイムゾーンおよび日付時刻を手動で設定することを推奨します。

図 10.1　[System Manager] ウィンドウでストレージシステムのクロックを手動で同期する方法



11. プロトコルとポートの保護

ソリューションの強化には、オンボックスセキュリティの操作と機能を実行するだけでなく、オフボックスセキュリティメカニズムも含める必要があります。ファイアウォールや侵入防止システム (IPS) といったセキュリティデバイスなど、追加の基盤デバイスを活用して、SANtricity へのアクセスをフィルタリングおよび制限することは、厳格なセキュリティ体制を確立し、維持するための効果的な方法です。表 11.1 に、SANtricity ソリューションで使用される一般的なプロトコルとポートを示します。この情報は、環境とそのリソースへのアクセスをフィルタリングして制限するための重要な要素です。

表 11.1　一般的なプロトコルとポート

サービス	ポート/プロトコル	説明
SSH	22/TCP	セキュアシェルログイン
SMTP	25/TCP	簡易メール転送プロトコル
HTTP	80/TCP	管理用 REST インターフェース (8443 にリダイレクトする)
NTP	123/UDP	ネットワークタイムプロトコル
SNMP	161/UDP	簡易ネットワーク管理プロトコル
SNMP	162/UDP	簡易ネットワーク管理プロトコル
LDAP	389/(UCP/TCP)	ローカルディレクトリ
HTTPS	443/TCP	管理 REST インターフェース用のセキュア HTTP
Syslog	515/UDP	Syslog サーバー
LDAPS	636/TCP	セキュア LDAP
SYMbol	2463/TCP	レガシー管理インターフェース
iSCSI	3260/TCP	iSCSI ターゲットポート
External Key Mgmt.	5696/TCP	外部キー管理
HTTP	8080/TCP	管理用 REST インターフェース (8443 にリダイレクトする)
HTTPS	8443/TCP	管理用 REST インターフェース

12. サービス拒否機能

受信 HTTP 要求は、Web サーバーが要求の拒否を開始するまでに 1 秒間に受け入れる要求数を制御する一連の設定を使用してレート制限できます。この機能は、REST API を使用して管理する必要があります。System Manager では公開されていせん。

- **enabled** パラメータは、レート制限を有効（on）または無効（off）にします。デフォルトは off です。
- **maxRequestsPerSec** パラメータは、Web サーバーが要求の拒否を開始するまでに 1 秒間に受け入れる受信要求の合計数を制御します。要求の拒否は、発信元 IP アドレスを問わず行われます。発信元クライアントに関係なく、レート制限を超えた後に受信された順に行われます。
- **ipExcludeList** パラメータを使用すると、レート制限から除外される IPv4 または IPv6 アドレスの一覧を指定できます。これにより、オーバーフロー攻撃中に悪意のないクライアントがサービス拒否されるのを防ぎます。一覧中のアドレスには、CIDR 表記を使用できます。

受信レートが制限を超えると、要求は 429（Too many requests）ステータスで拒否され始めます。拒否された要求の IP アドレスを含む監査ログエントリも生成されます。**ipExcludeList** パラメータは、Web サーバーが受信要求で過負荷になっている場合でも、重要なトラフィックのフローを継続できるようにする手段を提供します。

レート制限の設定内容を取得するには、以下の REST API を使用します。

- API
  `Administration > GET /devmgr/v2/settings`
- GET の戻り値の本文

```
{
  "serverSettings": {
    "httpResponseHeaders": [],
    "relativeRedirectAllowed": true,
    "tls13Disabled": false,
    "rateLimit": {
      "enabled": true,
      "maxRequestsPerSec": 50,
      "ipExcludeList": []
    }
  }
}
```

レート制限の設定を変更するには、以下の REST API を使用します。

- API
  `Administration > POST /devmgr/v2/settings`
- POST のリクエストボディ

```
{
  "serverSettings": {
    "httpResponseHeaders": [],
    "relativeRedirectAllowed": true,
    "tls13Disabled": false,
    "rateLimit": {
      "enabled": true,
      "maxRequestsPerSec": 50,
      "ipExcludeList": []
    }
  }
}
```

レート制限の設定を変更するには、戻り値の本文にある `rateLimit` オブジェクトを操作します。リスト値 `ipExcludeList` は、レート制限機能から除外する IP アドレス一式です。

13. まとめ

攻撃者が脆弱なシステムを探しているため、サイバーセキュリティの脅威は近年、より一般的になっています。このガイドに記載されている推奨事項に従って実装することにより、潜在的な攻撃ベクターを排除し、攻撃対象を保護することで、ETERNUS AB/HB series ストレージシステムのセキュリティリスクを軽減できます。

A. セキュリティリソース

脆弱性とインシデントのレポート、セキュリティ対応、およびお客様の機密性に関する情報については、当社サポート部門にご連絡ください。

ETERNUS AB series オールフラッシュアレイ , ETERNUS HB series ハイブリッドアレイ
SANtricity セキュリティ強化ガイド

P3AG-6042-03Z0

発行年月　2025 年 3 月
発行責任　エフサステクノロジーズ株式会社

Fsas Technologies