



属性ベースのアクセス制御 ONTAP 9

NetApp
December 20, 2024

目次

属性ベースのアクセス制御	1
ONTAPによる属性ベースのアクセス制御	1
ONTAPを使用したABACへのアプローチ	1

属性ベースのアクセス制御

ONTAPによる属性ベースのアクセス制御

ONTAPを使用して、属性と属性ベースアクセス制御（ABAC）を使用した強化されたRBACを実装できます。ONTAPには、ファイルレベルのABACを実現するためにお客様が使用できるいくつかのアプローチが用意されています。たとえば、ラベル付きのNFS 4.2や、NFSやSMB / CIFSを使用するXATTRSなどです。

属性ベースアクセス制御（ABAC）は、ユーザ属性、リソース属性、および環境条件を考慮してアクセス権を管理するための高度な方法です。国立標準技術研究所(NIST)は、ABACの標準を確立し、安全で一貫した実装のためのフレームワークを提供しています。

ONTAP 9 12.1以降では、ONTAPにNFSv4.2セキュリティラベルと拡張属性（XATTRS）を設定して、ロールベースアクセス制御（RBAC）および属性ベースアクセス制御（ABAC）のIDと統合できるようになりました。この統合により、ONTAPは、NIST ABAC準拠のデータ管理ソリューションに分類されるアクセス制御ソフトウェアにアクセスできます。これにより、ポリシー実行ポイント(PEP)、ポリシー決定ポイント(PDP)、およびユーザー、リソース、および環境に関連する属性を考慮したポリシーなど、複雑な環境でアクセス権を管理するための堅牢で高度なアプローチが提供されます。

NetApp ONTAPと拡張属性(XATTRS)および属性ベースアクセス制御(ABAC)ソフトウェアの統合は、NIST Special Publication 800-162で規定されているガイドラインに沿っており、ABAC実装のためのNIST標準への準拠を保証します。NFS 4.2セキュリティラベルとXATTRSを使用すると、ユーザ定義属性をファイルに関連付けることができ、アクセス制御の決定でリソース属性を考慮するというNIST ABAC標準の要件を満たすことができます。ABACソフトウェアのPEPとPDPは、アクセス制御プロセスにおけるこれらのコンポーネントに対するNIST ABAC標準の要件に準拠しています。複数の属性と条件を考慮した複雑なポリシーを定義する機能は、NIST ABAC標準のポリシーベースのアクセス制御の要件と一致しています。

関連情報

- ["ONTAPを使用したABACへのアプローチ"](#)
- ["NetApp ONTAPのNFS：ベストプラクティスおよび実装ガイド"](#)
- コメント要求（RFC）
 - RFC 2203：RPCSEC_GSS Protocol Specification
 - RFC 3530：Network File System (NFS) Version 4 Protocol

ONTAPを使用したABACへのアプローチ

ONTAPには、ファイルレベルのABACを実現するためにお客様が使用できるさまざまなアプローチが用意されています。たとえば、NFSv4.2や、NFSやSMB / CIFSを使用するXATTRSなどです。

NFSv4.2というラベル

nfs.9.1以降でONTAP 9は、nfsというラベル付きのNFSv4.2機能がサポートされます。

ラベル付きNFSは、SELinuxラベルとMandatory Access Control（MAC；強制アクセス制御）を使用して、フ

ファイルやフォルダへのきめ細かなアクセスを管理する方法です。これらのMACラベルはファイルおよびフォルダに格納され、UNIX権限およびNFSv4.x ACLと連携して機能します。

ラベル付きNFSがサポートされるため、ONTAPはNFSクライアントのSELinuxラベル設定を認識して認識できるようになりました。ラベル付きNFSはRFC-7204でカバーされています。

NFSv4.2というラベルのユースケースには、次のようなものがあります。

- 仮想マシン (VM) イメージのMACラベル付け
- 公共機関のデータセキュリティ分類 (シークレット、トップシークレット、その他の分類)
- セキュリティコンプライアンス
- ディスクレス Linux

NFSv4.2を有効にする

ラベル付きNFSを有効または無効にするには、次のadvanced権限オプションを使用します。

```
[-v4.2-seclabel {enabled|disabled}] - NFSV4.2 Security Label Support  
(privilege: advanced)
```

このパラメータはオプションで、デフォルト設定はです。 disabled

ラベル付けされた**NFSv4.2**の適用モード

ONTAP 9 .9.1以降では、ONTAPで次の強制モードがサポートされます。

- 制限付きサーバーモード：ONTAPはラベルを強制できませんが、ラベルを保存および送信できます。



MAC ラベルを変更する機能も、クライアント側で適用されます。

- ゲストモード：クライアントにNFS対応 (v4.1以前) のラベルが付けられていない場合、MACラベルは送信されません。



ONTAPは現在、フルモード (MACラベルの保存と適用) をサポートしていません。

NFSv4.2というラベルの設定例

次の設定例は、Red Hat Enterprise Linuxリリース9.3 (Plow) を使用した概念を示しています。

このユーザ `jrsmith` は、John R. Smithのクレデンシャルに基づいて作成され、次のアカウントPrivilegesを持ちます。

- ユーザ名= jrsmith
- Privileges = uid=1112(jrsmith) gid=1112(jrsmith) groups=1112(jrsmith)
context=user_u:user_r:user_t:s0

ロールには2つあります。管理者アカウントは、次のMLS Privilegesの表で説明されているように、特権ユーザおよびユーザ `jrsmith` です。

ユーザ	ロール	タイプ	レベル
admins	sysadm_r	sysadm_t	t:s0
jrsmith	user_r	user_t	t:s1 - t:s4

この例の環境では、ユーザー jrsmith`はに `s3`あるレベルのファイルにアクセスできます `s0。以下に概説する既存のセキュリティ分類を強化して、管理者がユーザー固有のデータにアクセスできないようにすることができます。

- S0 =権限管理者ユーザデータ
- S0 =未分類データ
- S1 =社外秘
- S2 =シークレットデータ
- S3 =トップシークレットデータ



組織のセキュリティポリシーに従う

MCSを使用するNFSv4.2セキュリティレベルの例

マルチレベルセキュリティ (MLS) に加えて、マルチカテゴリセキュリティ (MCS) と呼ばれる別の機能を使用すると、プロジェクトなどのカテゴリを定義できます。

NFSセキュリティラベル	値
entitySecurityMark	t:s01 = UNCLASSIFIED

拡張属性 (XATTRS)

ONTAP 9 12.1以降、ONTAPはxattrs.xattrsをサポートしています。xattrsを使用すると、アクセスコントロールリスト(ACL)やユーザ定義属性など、システムによって提供されるもの以外のファイルやディレクトリにメタデータを関連付けることができます。

xattrsを実装するには、Linuxでと getfattr`コマンドラインユーティリティを使用してファイルシステムオブジェクトのxattrsを管理できます `setfattr。これらのツールを使用すると、ファイルやディレクトリの追加メタデータを強力に管理できます。不適切な使用は予期しない動作やセキュリティ上の問題につながる可能性があるため、注意して使用する必要があります。使用方法の詳細については、および `getfattr`のマニュアルページを参照するか、信頼性の高いその他のドキュメントを参照して `setfattr`ください。

ONTAPファイルシステムでxattrsが有効になっている場合`ユーザーはファイルの任意の属性を設定`変更`取得`できますこれらの属性は、アクセス制御情報など、標準のファイル属性セットではキャプチャされないファイルに関する追加情報を格納するために使用できます。

ONTAPでxattrsを使用するための要件

- Red Hat Enterprise Linux 8.4以降

- Ubuntu 22.04以降
- 各ファイルには最大128個のxattrsを含めることができます。
- xattrキーは255バイトに制限されています
- キーまたは値の合計サイズは'xattrごとに1,729バイトです
- ディレクトリとファイルはxattrsを持つことができる
- xattrsを設定および取得するには、ユーザおよびグループに対して書き込みモードビットが有効になっている必要があります。 w

xattrsのユースケース

xattrsはユーザネームスペース内で使用され、ONTAP自体に本質的な意味を持たない。代わりに、それらの実用的なアプリケーションは、ファイルシステムとやり取りするクライアント側のアプリケーションによって排他的に決定され、管理されます。

xattrの使用例：

- ファイルの作成を担当するアプリケーションの名前を記録します。
- ファイルの取得元の電子メールメッセージへの参照を維持する。
- ファイルオブジェクトを整理するための分類フレームワークの確立。
- ファイルに元のダウンロード元のURLをラベル付けする。

xattrsの管理用コマンド

- setfattr:ファイルまたはディレクトリの拡張属性を設定します

```
setfattr -n <attribute_name> -v <attribute_value> <file or directory name>
```

コマンド例：

```
setfattr -n user.comment -v test example.txt
```

- getfattr:特定の拡張属性の値を取得するか'ファイルまたはディレクトリのすべての拡張属性を一覧表示します

特定の属性： getfattr -n <attribute_name> <file or directory name>

すべての属性： getfattr <file or directory name>

コマンド例：

```
getfattr -n user.comment example.txt
```

xattr	値
user.digitalIdentifier	CN=John Smith jrsmith, OU=Finance, OU=U.S.ACME, O=US, C=US

xattr	値
user.countryOfAffiliations	USA

拡張属性に対するACEによるユーザ権限

Access Control Entry (ACE; アクセス制御エントリ) は、アクセス制御リスト (ACL) 内のコンポーネントで、ファイルやディレクトリなど、特定のリソースに対して個々のユーザまたはユーザグループに付与されるアクセス権または権限を定義します。各ACEは、許可または拒否されるアクセスのタイプを指定し、特定のセキュリティプリンシパル (ユーザまたはグループのID) に関連付けます。

ファイルタイプ	xattrの取得	xattrsの設定
ファイル	R	A、w、T
ディレクトリ	R	T

xattrsに必要な権限の説明：

retrieve xattr: ユーザーがファイルまたはディレクトリの拡張属性を読み取るために必要な権限。「R」は、読み取り権限が必要であることを示します。**Set xattrs:** 拡張属性を変更または設定するために必要な権限。「A」、「w」、「T」は、append、write、xattrsに関連する特定のパーミッションなど、パーミッションの異なる例を表しています。ファイル: 拡張属性を設定するには、追加、書き込み、およびxattrsに関連する特別な権限が必要です。ディレクトリ: 拡張属性を設定するには、特定の権限「T」が必要です。

xattrsのSMB / CIFSプロトコルのサポート

ONTAPのSMB/CIFSプロトコルのサポートはWindows環境のファイルメタデータに不可欠なxattrsの包括的な処理にまで拡張されています。拡張属性を使用すると、ユーザーとアプリケーションは、作成者の詳細、カスタムセキュリティ記述子、アプリケーション固有のデータなど、標準のファイル属性セット以外の追加情報を保存できます。ONTAPのSMB/CIFS実装により、これらのxattrsが完全にサポートされ、機能とポリシーの適用にこのメタデータに依存するWindowsのサービスやアプリケーションとのシームレスな統合が可能になります。

ONTAPで管理されているSMB/CIFS共有を介してファイルにアクセスまたは転送されると、xattrsの整合性が維持され、すべてのメタデータが保持され、整合性が維持されます。これは、セキュリティ設定を維持したり、構成や操作をxattrsに依存するアプリケーションで特に重要です。ONTAPのSMB/CIFSコンテキスト内でのxattrsの堅牢な処理により、異なるプラットフォームや環境間でのファイル共有の信頼性と安全性が確保されます。これにより、ユーザーはシームレスなエクスペリエンスを提供し、管理者はデータガバナンスポリシーを確実に維持できます。コラボレーション、データアーカイブ、コンプライアンスのいずれにおいても、SMB/CIFS共有内のxattrsに対するONTAPの関心は、マルチプラットフォーム環境における優れたデータ管理と相互運用性への取り組みを表しています。

ABACのポリシー施行ポイント (PEP) およびポリシー決定ポイント (PDP)

Attribute-Based Access Control (ABAC; 属性ベースアクセス制御) システムでは、Policy Enforcement Point (PEP; ポリシー適用ポイント) と Policy Decision Point (PDP; ポリシー決定ポイント) が重要な役割を果たします。PEPはアクセス制御ポリシーの適用を担当し、PDPはポリシーに基づいてアクセスを許可するか拒否するかを決定します。

提供されるPythonコードスニペットのコンテキストでは、スクリプト自体がPEPとして機能します。ファイ

ルを開いて内容を読み取ることでアクセスを許可するか、を発行してアクセスを拒否することで、アクセス制御の決定を強制します `PermissionError`。

一方、PDPは基盤となるSELinuxシステムの一部となる。スクリプトが特定のSELinuxコンテキストでファイルを開こうとすると、SELinuxシステムはポリシーをチェックして、アクセスを許可するか拒否するかを決定します。この決定はスクリプトによって実行されます。

以下は、ABAC環境でこのコードがどのように機能するかについて、手順を追って説明した例です。

1. スクリプトは、関数を使用してSELinuxコンテキストをコンテキスト ``selinux.setcon()`` に設定し ``jrsmith`` ます。これは、ファイルにアクセスしようとする場合と同じ ``jrsmith`` です。
2. スクリプトはファイルを開こうとします。ここでPEPが登場します。
3. SELinuxシステムは、ポリシーをチェックして、（具体的には、SELinuxコンテキストを持つユーザ `jrsmith``）がファイルへのアクセスを許可されているかどうかを確認し ``jrsmith`` ます。これがPDPの役割です。
4. がファイルへのアクセスを許可されている場合、``jrsmith`` SELinuxシステムはスクリプトがファイルを開くことを許可し、スクリプトはファイルの内容を読み取り、印刷します。
5. がファイルへのアクセスを許可されていない場合、`jrsmith`` SELinuxシステムはスクリプトがファイルを開くことを禁止し、スクリプトはを生成します ``PermissionError`。
6. このスクリプトは、一時的なコンテキストの変更が他の処理に影響しないように、元のSELinuxコンテキストをリストアします。

Pythonを使用すると、コンテキストを取得するためのコードを以下に示します。変数ファイルのパスはチェックするドキュメントです。

```
#Get the current context

context = selinux.getfilecon(file_path)[1]
```

ONTAPクローニングとSnapMirror

ONTAPのクローニングおよびSnapMirrorテクノロジーは、効率的で信頼性の高いデータレプリケーションおよびクローニング機能を提供するように設計されています。ファイルに関連する追加のメタデータ（セキュリティラベル、アクセス制御情報、ユーザ定義データなど）を格納するため、拡張属性（`xattrs`）を含むファイルデータのすべての要素が`file.xattrs`とともに保持および転送されます。`xattrs`は、ファイルのコンテキストと整合性の維持に不可欠です。

ONTAPのFlexCloneテクノロジーを使用してボリュームをクローニングすると、ボリュームの完全な書き込み可能なレプリカが作成されます。このクローニングプロセスは瞬時に実行されるスペース効率に優れており、すべてのファイルデータとメタデータが含まれているため、`xattrs`を完全にレプリケートできます。同様に、SnapMirrorでは、データが完全に忠実にセカンダリシステムにミラーリングされます。これには`xattrs`も含まれます。`xattrs`は、このメタデータに依存するアプリケーションが正しく機能するために非常に重要です。

NetApp ONTAPでは、クローニング処理とレプリケーション処理の両方に`xattrs`を含めることで、プライマリストレージシステムとセカンダリストレージシステム全体で、すべての特性を含む完全なデータセットを使用して一貫性を確保します。この包括的なデータ管理アプローチは、一貫したデータ保護、迅速なリカバリ、コンプライアンスと規制基準への準拠を必要とする組織にとって不可欠です。また、オンプレミスでもクラウド

でも、さまざまな環境にわたってデータの管理が簡易化されるため、ユーザはプロセス中もデータが完全で変更されていないという安心感を得ることができます。



NFSv4.2セキュリティラベルには、に定義されている注意事項があります [NFSv4.2というラベル](#)。

データアクセスの制御例

次に、John R SmithのPKI証明書に格納されているデータのエン트리例を示します。これは、NetAppのアプローチをファイルに適用し、きめ細かなアクセス制御を提供する方法を示しています。



これらの例は説明を目的としたものであり、NFSv4.2セキュリティラベルおよびxattrsとはどのメタデータであるかを定義するのは政府の責任です。わかりやすいように更新とラベルの保持の詳細は省略しています。

キー	値
entitySecurityMark	T : S01 =未分類
情報	<pre>{ "commonName": { "value": "Smith John R jrsmith" }, "emailAddresses": [{ "value": "jrsmith@dod.mil" }], "employeeId": { "value": "00000387835" }, "firstName": { "value": "John" }, "lastName": { "value": "Smith" }, "telephoneNumber": { "value": "938/260-9537" }, "uid": { "value": "jrsmith" } }</pre>

キー	値
仕様	"DoD"
UUID	b4111349-7875-4115-AD30-0928565f2e15
管理組織	<pre>{ "value": "DoD" }</pre>
ブリーフィング	<pre>[{ "value": "ABC1000" }, { "value": "DEF1001" }, { "value": "EFG2000" }]</pre>
市民権ステータス	<pre>{ "value": "US" }</pre>

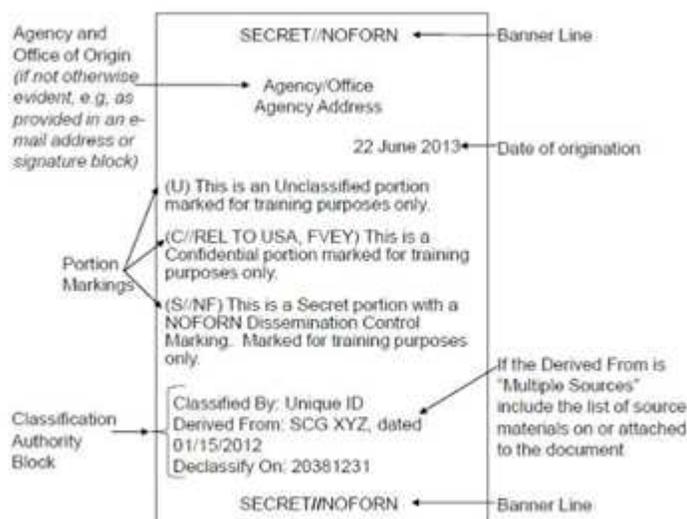
キー	値
クリアランス	<pre>[{ "value": "TS" }, { "value": "S" }, { "value": "C" }, { "value": "U" }]</pre>
加盟国	<pre>[{ "value": "USA" }]</pre>
デジタル識別子	<pre>{ "classification": "UNCLASSIFIED", "value": "cn=smith john r jrsmith, ou=dod, o=u.s. government, c=us" }</pre>
転送先	<pre>{ "value": "DoD" }</pre>
DutyOrganization	<pre>{ "value": "DoD" }</pre>

キー	値
エンティティタイプ	<pre>{ "value": "GOV" }</pre>
FineAccessControls	<pre>[{ "value": "SI" }, { "value": "TK" }, { "value": "NSYS" }]</pre>

これらのPKIエンタイトルメントには、データ型やアトリビュションによるアクセスなど、John R. Smithのアクセスの詳細が表示されます。

John R. Smithが "sample_analysis.doc" というドキュメントを作成して保存した場合、関連するポリシーガイダンスの発行に従って、ユーザーは次の図に示すように、ドキュメントの分類に基づいて適切なバナーと部分マーキング、代理店および原産地オフィス、および適切な分類権限ブロックを追加します。この豊富なメタデータは、自然言語処理（NLP）によってスキャンされ、マーキングから意味を与えるためのルールが適用された後で初めて理解できます。NetApp BlueXP Classificationなどのツールはこれを行うことができますが、ドキュメント内を参照する権限が必要なため、アクセス制御の決定にはあまり効率的ではありません。

未分類のCAPCOドキュメント部分マーキング



IC-TDFメタデータがファイルとは別に格納されているシナリオでは、NetAppは詳細なアクセス制御レイヤを追加することを推奨しています。これには、アクセス制御情報がディレクトリレベルおよび各ファイルに関連付けられて格納されることが含まれます。例として、次のタグがファイルにリンクされているとします。

- NFSv4.2セキュリティラベル：セキュリティの決定に使用されます。
- xattrs：ファイルおよび組織のプログラム要件に関連する補足情報を提供します。

次のキーと値のペアは、xattrsとして保存できるメタデータの例であり、ファイルの作成者と関連するセキュリティ分類に関する詳細情報を提供します。クライアントアプリケーションでこのメタデータを使用すると、十分な情報に基づいてアクセスに関する意思決定を行い、組織の標準や要件に従ってファイルを整理できます。

キー	値
user.uuid	"761d2e3c-e778-4ee4-997b-3bb9a6a1d3fa"
user.entitySecurityMark	"UNCLASSIFIED"
user.specification	"INFO"

キー	値
user.Info	<pre>{ "commonName": { "value": "Smith John R jrsmith" }, "currentOrganization": { "value": "TUV33" }, "displayName": { "value": "John Smith" }, "emailAddresses": ["jrsmith@example.org"], "employeeId": { "value": "00000405732" }, "firstName": { "value": "John" }, "lastName": { "value": "Smith" }, "managers": [{ "value": "" }], "organizations": [{ "value": "TUV33" }, { "value": "WXY44" }], "personalTitle": { "value": "" }, "secureTelephoneNumber": { "value": "506-7718" }, "telephoneNumber": { "value": "264/160-7187" }, "title": { "value": "Software Engineer" }, }</pre>

キー	値
user.geo_point	[-78.7941, 35.7956]
	}

ラベルに対する変更の監査

xattrsまたはNFSセキュリティラベルに対する変更の監査は、ファイルシステムの管理とセキュリティの重要な側面です。標準のファイルシステム監査ツールを使用すると、拡張属性やセキュリティラベルの変更など、ファイルシステムに対するすべての変更を監視およびロギングできます。

Linux環境では、auditd`ファイルシステムイベントの監査を確立するために一般にデーモンが使用されます。管理者は、xattrの変更（、`lsetxattr`など）に関連する特定のシステムコールを監視し、`fsetxattr`属性と、`lremovexattr``removexattr`の設定、および`removexattr`属性の削除を監視するルールを設定でき`setxattr`ます。

ONTAP FPolicyは、ファイル操作をリアルタイムで監視および制御するための堅牢なフレームワークを提供することで、これらの機能を拡張します。FPolicyは、さまざまな属性xattrイベントをサポートするように設定できます。これにより、ファイル操作をきめ細かく制御したり、包括的なデータ管理ポリシーを適用したりできます。

xattrsを使用するユーザ（特にNFSv3およびNFSv4環境）では、ファイル処理とフィルタの特定の組み合わせのみが監視対象としてサポートされます。FPolicyによるNFSv3およびNFSv4のファイルアクセスイベントの監視でサポートされるファイル処理とフィルタの組み合わせを次に示します。

サポートされているファイル操作	サポートされているフィルタ
setattr	offline-bit, setattr_with_owner_change, setattr_with_group_change, setattr_with_mode_change, setattr_with_modify_time_change, setattr_with_access_time_change, setattr_with_size_change, exclude_directory

属性設定操作のauditdログスニペットの例：

```
type=SYSCALL msg=audit(1713451401.168:106964): arch=c000003e syscall=188
success=yes exit=0 a0=7fac252f0590 a1=7fac251d4750 a2=7fac252e50a0 a3=25
items=1 ppid=247417 pid=247563 auid=1112 uid=1112 gid=1112 euid=1112
suid=1112 fsuid=1112 egid=1112 sgid=1112 fsgid=1112 tty=pts0 ses=141
comm="python3" exe="/usr/bin/python3.9"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
key="*set-xattr*"ARCH=x86_64 SYSCALL=*setxattr* AUID="jrsmith"
UID="jrsmith" GID="jrsmith" EUID="jrsmith" SUID="jrsmith"
FSUID="jrsmith" EGID="jrsmith" SGID="jrsmith" FSGID="jrsmith"
```

xattrsを使用するユーザに対してONTAP FPolicyを有効にすると、ファイルシステムの整合性とセキュリティを維持するために不可欠な可視性と制御のレイヤが提供されます。FPolicyの高度な監視機能を活用することで、組織はxattrsに対するすべての変更を追跡、監査し、セキュリティおよびコンプライアンス基準に準拠させることができます。ファイルシステム管理に対するこのプロアクティブなアプローチが、データガバナンス

と保護戦略を強化したいと考えている組織にとって、ONTAP FPolicyを有効にすることが強く推奨される理由です。

ABAC IDおよびアクセス制御ソフトウェアとの統合

属性ベースアクセス制御(ABAC)の機能を最大限に活用するために、ONTAPはABAC指向のIDおよびアクセス管理ソフトウェアと統合できます。



このコンテンツと並行して、NetAppにはGreyBoxを使用したリファレンス実装があります。このコンテンツの1つの前提は、政府のアイデンティティ、認証、およびアクセスサービスには、少なくともファイルシステムへのアクセスの仲介者として機能するPolicy Enforcement Point (PEP) とPolicy Decision Point (PDP) が含まれていることです。

実際的な設定では、NFSセキュリティラベルとxattrsを組み合わせて使用します。これらは、分類、セキュリティ、アプリケーション、コンテンツなど、さまざまなメタデータを表すために使用されます。これらはすべて、ABACの決定に役立ちます。たとえば、xattrは、PDPが意思決定プロセスに使用するリソース属性を格納するために使用できます。属性は、ファイルの分類レベルを表すように定義できます（「未分類」、「機密」、「シークレット」、「トップシークレット」など）。その後、PDPはこの属性を使用して、ユーザーがクリアランスレベル以下の分類レベルを持つファイルのみにアクセスするように制限するポリシーを適用できます。

ABACのプロセスフローの例

1. ユーザーは、PEPへのシステムアクセスにクレデンシャル（PKI、OAuth、SAMLなど）を提示し、PDPから結果を取得します。

PEPの役割は、ユーザーのアクセス要求を代行受信してPDPに転送することです。

2. PDPは、確立されたABACポリシーに照らしてこの要求を評価します。

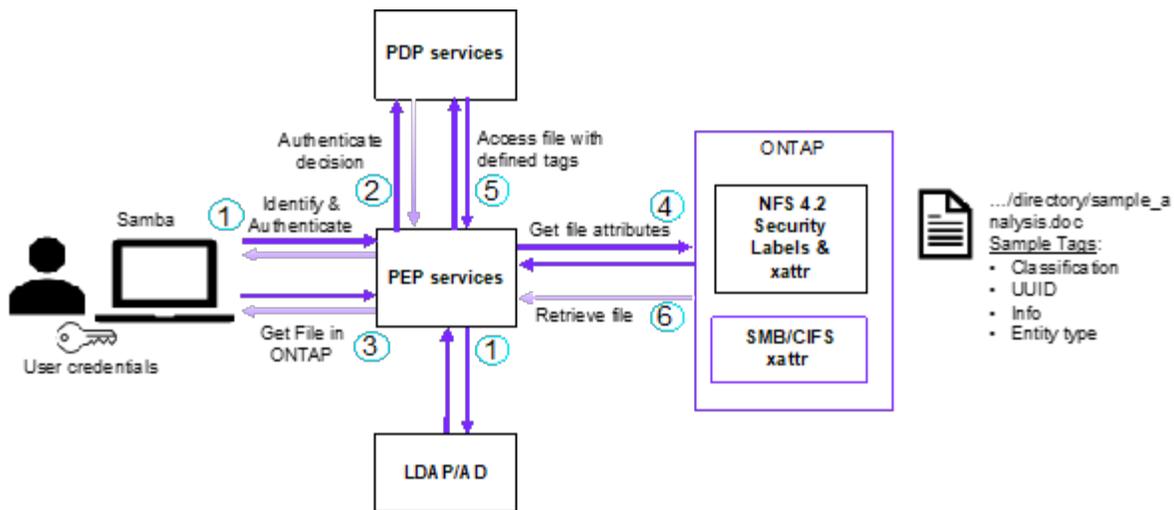
これらのポリシーでは、ユーザー、問題のリソース、および周囲の環境に関連するさまざまな属性が考慮されます。これらのポリシーに基づいて、PDPはアクセスを許可するか拒否するかを決定し、その決定をPEPに伝えます。

PDPはPEPにポリシーを提供して実施します。PEPはこの決定を実行し、PDPの決定に従ってユーザーのアクセス要求を許可または拒否します。

3. 要求が成功すると、ユーザーはONTAPに格納されているファイル（AFF、AFF -Cなど）を要求します。
4. 要求が成功すると、PEPはドキュメントから詳細なアクセス制御タグを取得します。
5. PEPは、そのユーザーの証明書に基づいてユーザーのポリシーを要求します。
6. ユーザーがファイルにアクセスできる場合、PEPはポリシーとタグに基づいて決定を行い、ユーザーがファイルを取得できるようにします。



実際のアクセスは、プロキシされていないトークンを使用して行われる場合があります。



関連情報

- ["NetApp ONTAPのNFS：ベストプラクティスおよび実装ガイド"](#)
- コメント要求 (RFC)
 - RFC 2203：RPCSEC_GSS Protocol Specification
 - RFC 3530：Network File System (NFS) Version 4 Protocol

著作権に関する情報

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。