Attributbasierte Zugriffssteuerung

ONTAP 9

NetApp
January 17, 2025

Inhalt

Attributbasierte Zugriffssteuerung

Attributbasierte Zugriffssteuerung mit ONTAP

Sie können erweiterte RBAC mit Attributen und attributbasierter Zugriffssteuerung (Attribute Based Access Control, ABAC) mithilfe von ONTAP implementieren. ONTAP bietet verschiedene Ansätze, mit denen ein Kunde ABAC auf Dateiebene realisieren kann, einschließlich der Bezeichnung NFS 4.2 und XATTRS mit NFS und SMB/CIFS.

Attributbasierte Zugriffskontrolle (ABAC) ist eine ausgefeilte Methode zur Verwaltung von Zugriffsrechten, die Benutzerattribute, Ressourcenattribute und Umgebungsbedingungen berücksichtigt. Das National Institute of Standards and Technology (NIST) hat einen Standard für ABAC entwickelt, der ein Framework für seine sichere und konsistente Implementierung bietet.

Ab ONTAP 9.12.1 können Sie ONTAP mit NFSv4.2 Security Labels und Extended Attributes (XATTRS) so konfigurieren, dass es mit einer rollenbasierten Zugriffskontrolle (RBAC) und einer attributbasierten Zugriffskontrolle (ABAC)-Identität integriert werden kann. Durch diese Integration kann ONTAP eine als NIST ABAC-konforme Datenmanagement-Lösung kategorisierte Zugriffskontroll-Software nutzen. Sie bietet einen robusten und fortschrittlichen Ansatz für das Management von Zugriffsrechten in komplexen Umgebungen, einschließlich Policy Enforcement Point (PEP), einem Policy Decision Point (PDP) und Richtlinien, die Attribute berücksichtigen, die mit dem Benutzer, der Ressource und der Umgebung verknüpft sind.

Die Integration von NetApp ONTAP mit erweiterten Attributen (XATTRS) und der attributbasierten Zugriffskontrollsoftware (ABAC) entspricht den Richtlinien der NIST-Sonderveröffentlichung 800-162, um die Einhaltung der NIST-Standards für die ABAC-Implementierung sicherzustellen. Die Verwendung von NFS 4.2 Security Labels und XATTRS ermöglicht die Zuordnung von benutzerdefinierten Attributen mit Dateien und erfüllt damit die Anforderung des NIST ABAC Standards zur Berücksichtigung von Ressourcenattributen bei Entscheidungen zur Zugriffskontrolle. PEP und PDP der ABAC-Software entsprechen den Anforderungen des NIST ABAC-Standards für diese Komponenten im Zutrittskontrollprozess. Die Fähigkeit, komplexe Richtlinien zu definieren, die mehrere Attribute und Bedingungen berücksichtigen, entspricht den Anforderungen des NIST ABAC-Standards für richtlinienbasierte Zugriffssteuerung.

Verwandte Informationen

- "Ansätze für ABAC mit ONTAP"
- "NFS in NetApp ONTAP: Best Practice und Implementierungsleitfaden"
- Anforderung von Kommentaren (RFC)
  - RFC 2203: RPCSEC_GSS-Protokollspezifikation
  - RFC 3530: Network File System (NFS) Version 4 Protocol

Ansätze für ABAC mit ONTAP

ONTAP bietet verschiedene Ansätze, die ein Kunde zur Erreichung eines ABAC auf Dateiebene einsetzen könnte, einschließlich der Kennzeichnung NFSv4.2 und XATTRS mit NFS und SMB/CIFS.

Mit NFSv4.2 gekennzeichnet

Ab ONTAP 9.9.1 wird die NFS v4.2 Funktion mit der Bezeichnung NFS unterstützt.

Label NFS ist eine Möglichkeit, den granularen Datei- und Ordnerzugriff mithilfe von SELinux-Labels und Mandatory Access Control (MAC) zu managen. Diese MAC-Labels werden mit Dateien und Ordnern gespeichert und funktionieren in Verbindung mit UNIX-Berechtigungen und NFSv4.x ACLs.

Durch die Unterstützung für gekennzeichnete NFS erkennt ONTAP jetzt die SELinux-Label-Einstellungen des NFS-Clients und versteht sie. Die Bezeichnung NFS ist in RFC-7204 enthalten.

Die folgenden Anwendungsfälle sind für die Kennzeichnung von NFSv4.2 enthalten:

- MAC-Beschriftung von Virtual Machine (VM) Images
- Datensicherheitsklassifizierung für den öffentlichen Sektor (geheime, streng geheime und andere Klassifizierungen)
- Sicherheits-Compliance
- Diskless Linux

Aktivieren Sie die Bezeichnung NFSv4.2

Sie können die Kennzeichnung von NFS mit der folgenden erweiterten Berechtigungsoption aktivieren oder deaktivieren:

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

Dieser Parameter ist optional, und die Standardeinstellung ist `disabled`.

Durchsetzungsmodi für die Bezeichnung NFSv4.2

Ab ONTAP 9.9.1 unterstützt ONTAP die folgenden Erzwingungsmodi:

- **Eingeschränkter Servermodus**: ONTAP kann die Labels nicht erzwingen, sondern speichern und übertragen.

  Die Möglichkeit, MAC-Labels zu ändern, ist auch vom Client zu erzwingen.

- **Gastmodus**: Wenn der Client nicht NFS-aware (v4.1 oder niedriger) ist, werden MAC-Labels nicht übertragen.

ONTAP unterstützt derzeit nicht den Vollmodus (Speichern und Erzwingen von MAC-Etiketten).

Beispielkonfiguration mit der Bezeichnung NFSv4.2

Die folgende Beispielkonfiguration zeigt Konzepte mit Red hat Enterprise Linux Version 9.3 (Plough).

Der Benutzer `jrsmith`, der basierend auf den Anmeldeinformationen von John R. Smith erstellt wurde, hat das folgende Konto Privileges:

- Benutzername = `jrsmith`
- Privileges = `uid=1112(jrsmith) gid=1112(jrsmith) groups=1112(jrsmith) context=user_u:user_r:user_t:s0`

Es gibt zwei Rollen: Das Administratorkonto, das ein privilegierter Benutzer und ein Benutzer ist `jrsmith`, wie in der folgenden MLS-Privileges-Tabelle beschrieben:

Benutzer	Rolle	Typ	Stufen
admins	sysadm_r	sysadm_t	t:s0
jrsmith	user_r	user_t	t:s1 - t:s4

In dieser Beispielumgebung hat der Benutzer `jrsmith` Zugriff auf Dateien auf den Ebenen `s0` bis `s3`. Wir können die bestehenden Sicherheitsklassifizierungen wie unten beschrieben verbessern, um sicherzustellen, dass Administratoren keinen Zugriff auf benutzerspezifische Daten haben.

- s0 = Berechtigungsverwaltung Benutzerdaten
- s0 = nicht klassifizierte Daten
- s1 = vertraulich
- s2 = geheime Daten
- s3 = Top-Geheimdaten

Befolgen Sie die Sicherheitsrichtlinien Ihres Unternehmens

Beispiel für das NFSv4.2-Sicherheitsetikett mit MCS

Zusätzlich zu Multi-Level Security (MLS) können Sie mit einer weiteren Funktion namens Multi-Category Security (MCS) Kategorien wie Projekte definieren.

NFS-Sicherheitsetikett	Wert
entitySecurityMark	t:s01 = UNCLASSIFIED

Erweiterte Attribute (XATTRS)

Ab ONTAP 9.12.1 unterstützt ONTAP xattrs. Xattrs ermöglicht die Zuordnung von Metadaten zu Dateien und Verzeichnissen über das, was vom System bereitgestellt wird, wie z. B. Zugriffskontrolllisten (ACLs) oder benutzerdefinierte Attribute.

Um xattrs zu implementieren, können Sie und `getfattr` Befehlszeilendienstprogramme in Linux für die Verwaltung von xattrs von Dateisystemobjekten verwenden `setfattr`. Diese Tools bieten eine leistungsstarke Möglichkeit, zusätzliche Metadaten für Dateien und Verzeichnisse zu managen. Sie sollten jedoch mit Vorsicht verwendet werden, da eine unsachgemäße Verwendung zu unerwartetem Verhalten oder Sicherheitsproblemen führen kann. Detaillierte Anweisungen zur Verwendung finden Sie stets auf den `setfattr` Manpages und `getfattr` in anderen zuverlässigen Dokumentationen.

Wenn xattrs auf einem ONTAP-Dateisystem aktiviert ist, können Benutzer beliebige Attribute auf Dateien festlegen, ändern und abrufen. Diese Attribute können verwendet werden, um zusätzliche Informationen über die Datei zu speichern, die nicht von den standardmäßigen Dateiattributen erfasst werden, z. B. Informationen zur Zugriffssteuerung.

Voraussetzungen für die Verwendung von xattrs in ONTAP

- Red hat Enterprise Linux 8.4 oder höher

- Ubuntu 22.04 oder höher
- Jede Datei kann bis zu 128 xattrs haben
- Xattr-Schlüssel sind auf 255 Byte begrenzt
- Die kombinierte Schlüssel- oder Wertgröße beträgt 1,729 Byte pro xattr
- Verzeichnisse und Dateien können xattrs haben
- Zum Festlegen und Abrufen von xattrs `w` oder Schreibmodus müssen Bits für den Benutzer und die Gruppe aktiviert sein

Anwendungsfälle für xattrs

Xattrs werden innerhalb des Benutzer-Namespace verwendet und haben keine intrinsische Bedeutung für ONTAP selbst. Stattdessen werden ihre praktischen Anwendungen ausschließlich von der Client-seitigen Anwendung bestimmt und verwaltet, die mit dem Dateisystem interagiert.

Anwendungsbeispiele für xattr:

- Aufzeichnen des Namens der Anwendung, die für die Erstellung einer Datei verantwortlich ist.
- Beibehalten eines Verweises auf die E-Mail-Nachricht, aus der eine Datei abgerufen wurde.
- Einrichten eines Kategorisierungsrahmens für die Organisation von Dateiobjekten.
- Beschriften von Dateien mit der URL ihrer ursprünglichen Download-Quelle.

Befehle zum Verwalten von xattrs

- `setfattr`: Legt ein erweitertes Attribut einer Datei oder eines Verzeichnisses fest:

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

  Beispielbefehl:

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

- `getfattr`: Ruft den Wert eines bestimmten erweiterten Attributs ab oder listet alle erweiterten Attribute einer Datei oder eines Verzeichnisses auf:

  Spezifisches Attribut: `getfattr -n <attribute_name> <file or directory name>`

  Alle Attribute: `getfattr <file or directory name>`

  Beispielbefehl:

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

Xattr	Wert
`user.digitalIdentifier`	`CN=John Smith jrsmith, OU=Finance, OU=U.S.ACME, O=US, C=US`
`user.countryOfAffiliations`	`USA`

Benutzerberechtigungen mit ACE für erweiterte Attribute

Ein Access Control Entry (ACE) ist eine Komponente innerhalb einer Access Control List (ACL), die die Zugriffsrechte oder Berechtigungen definiert, die einem einzelnen Benutzer oder einer Benutzergruppe für eine bestimmte Ressource, z. B. eine Datei oder ein Verzeichnis, gewährt werden. Jeder ACE gibt die Art des erlaubten oder abgelehnten Zugriffs an und ist mit einem bestimmten Sicherheitsprinzipal (Benutzer- oder Gruppenidentität) verknüpft.

Dateityp	Xattr. Abrufen	Xattrs einstellen
Datei	R	A,w,T
Verzeichnis	R	T

Erläuterung der für xattrs erforderlichen Berechtigungen:

Retrieve xattr: Die Berechtigungen, die ein Benutzer benötigt, um die erweiterten Attribute einer Datei oder eines Verzeichnisses zu lesen. Das „R“ bedeutet, dass Leseberechtigung erforderlich ist. **Set xattrs**: Die Berechtigungen, die benötigt werden, um die erweiterten Attribute zu ändern oder einzustellen. „A“, „w“ und „T“ stellen verschiedene Beispiele für Berechtigungen wie Append, Write und eine bestimmte Berechtigung in Bezug auf xattrs dar. **Dateien**: Benutzer benötigen Append, Write und möglicherweise eine spezielle Berechtigung im Zusammenhang mit xattrs, um erweiterte Attribute zu setzen. **Directories**: Eine spezielle Berechtigung "T" wird benötigt, um erweiterte Attribute zu setzen.

Unterstützung des SMB/CIFS-Protokolls für xattrs

Die Unterstützung von ONTAP für das SMB/CIFS-Protokoll erstreckt sich auch auf die umfassende Handhabung von xattrs, die einen integralen Bestandteil von Datei-Metadaten in Windows Umgebungen darstellen. Erweiterte Attribute ermöglichen es Benutzern und Anwendungen, zusätzliche Informationen über die standardmäßigen Dateiattribute hinaus zu speichern, wie z. B. Autorendetails, benutzerdefinierte Sicherheitsdeskriptoren oder anwendungsspezifische Daten. Die SMB/CIFS-Implementierung von ONTAP stellt sicher, dass diese xattrs vollständig unterstützt werden. Dies ermöglicht eine nahtlose Integration in Windows-Dienste und Anwendungen, die zur Funktions- und Richtliniendurchsetzung auf diese Metadaten angewiesen sind.

Wenn Dateien über von ONTAP gemanagte SMB/CIFS Shares abgerufen oder übertragen werden, bewahrt das System die Integrität von xattrs und sorgt so dafür, dass alle Metadaten erhalten bleiben und konsistent bleiben. Dies ist besonders wichtig für die Aufrechterhaltung der Sicherheitseinstellungen und für Anwendungen, die für die Konfiguration oder den Betrieb auf xattrs angewiesen sind. Die robuste Handhabung von xattrs im SMB/CIFS-Kontext von ONTAP gewährleistet, dass die gemeinsame Nutzung von Dateien über verschiedene Plattformen und Umgebungen hinweg zuverlässig und sicher ist. Dies bietet Benutzern eine nahtlose Erfahrung und Administratoren die Sicherheit, dass Data Governance-Richtlinien eingehalten werden. Ob für Zusammenarbeit, Datenarchivierung oder Compliance: Die Aufmerksamkeit von ONTAP auf xattrs innerhalb von SMB/CIFS Shares steht für herausragendes Datenmanagement und Interoperabilität in Umgebungen mit gemischten Betriebssystemen.

Policy Enforcement Point (PEP) und Policy Decision Point (PDP) in ABAC

In einem attributbasierten Zugriffskontrollsystem (ABAC) spielen der Policy Enforcement Point (PEP) und der Policy Decision Point (PDP) eine entscheidende Rolle. Der PEP ist für die Durchsetzung von Zugriffssteuerungsrichtlinien verantwortlich, während der PDP die Entscheidung darüber trifft, ob der Zugriff auf der Grundlage der Richtlinien gewährt oder verweigert werden soll.

Im Kontext des bereitgestellten Python-Code-Snippets fungiert das Skript selbst als PEP. Sie erzwingt die Entscheidung über die Zugriffskontrolle, indem sie entweder den Zugriff auf die Datei gewährt, indem sie sie

öffnet und ihren Inhalt liest oder den Zugriff durch die Erhebung eines verweigert `PermissionError`.

Das PDP hingegen wäre Teil des zugrunde liegenden SELinux-Systems. Wenn das Skript versucht, die Datei mit einem bestimmten SELinux-Kontext zu öffnen, prüft das SELinux-System seine Richtlinien, um zu entscheiden, ob der Zugriff gewährt oder verweigert werden soll. Diese Entscheidung wird dann durch das Skript durchgesetzt.

Nachfolgend finden Sie eine schrittweise Aufschlüsselung der Funktionsweise dieses Codes in einer ABAC-Umgebung:

1. Das Skript setzt den SELinux-Kontext über die Funktion auf den `jrsmith` Kontext `selinux.setcon()`. Dies entspricht dem `jrsmith` Versuch, auf die Datei zuzugreifen.
2. Das Skript versucht, die Datei zu öffnen. Hier kommt das PEP ins Spiel.
3. Das SELinux-System prüft seine Richtlinien, um zu ermitteln, ob `jrsmith` (oder genauer gesagt, ein Benutzer mit `jrsmith` SELinux-Kontext) auf die Datei zugreifen darf. Dies ist die Rolle der PDP.
4. Wenn `jrsmith` auf die Datei zugegriffen werden kann, lässt das SELinux-System das Skript die Datei öffnen, und das Skript liest und druckt den Inhalt der Datei.
5. Wenn `jrsmith` nicht auf die Datei zugegriffen werden kann, verhindert das SELinux-System, dass das Skript die Datei öffnet, und das Skript wirft ein `PermissionError`.
6. Das Skript stellt den ursprünglichen SELinux-Kontext wieder her, um sicherzustellen, dass die temporäre Kontextänderung keine Auswirkungen auf andere Vorgänge hat.

Mit Python wird der Code zum Abrufen des Kontexts unten angezeigt, wobei der Pfad der variablen Datei das zu prüfende Dokument ist:

```
#Get the current context

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

ONTAP Cloning und SnapMirror

Die Klon- und SnapMirror-Technologien von ONTAP wurden entwickelt, um effiziente und zuverlässige Datenreplizierungs- und Klonfunktionen zu bieten und sicherzustellen, dass alle Aspekte von Dateidaten, einschließlich erweiterter Attribute (xattrs), zusammen mit der Datei erhalten und übertragen werden. Xattrs sind wichtig, da sie zusätzliche Metadaten, die einer Datei zugeordnet sind, wie z. B. Sicherheitsetiketten, Zugriffskontrollinformationen und benutzerdefinierte Daten, die für die Aufrechterhaltung des Kontexts und der Integrität dieser Datei unerlässlich sind.

Wenn ein Volume mit der FlexClone-Technologie von ONTAP geklont wird, wird ein exaktes, beschreibbares Replikat des Volumes erstellt. Dieser Klonprozess ist sofort und platzsparend und umfasst alle Dateidaten und Metadaten, um sicherzustellen, dass xattrs vollständig repliziert werden. SnapMirror sorgt auf ähnliche Weise dafür, dass Daten originalgetreu auf ein sekundäres System gespiegelt werden. Dazu gehört xattrs, die entscheidend sind für Anwendungen, die auf diese Metadaten angewiesen sind, um korrekt zu funktionieren.

Durch die Einbeziehung von xattrs sowohl beim Klonen als auch bei der Replizierung stellt NetApp ONTAP sicher, dass der vollständige Datensatz mit allen seinen Merkmalen verfügbar und konsistent über primäre und sekundäre Storage-Systeme hinweg ist. Dieser umfassende Datenmanagementansatz ist für Unternehmen unerlässlich, die eine konsistente Datensicherung, schnelle Wiederherstellung und die Einhaltung von Compliance- und gesetzlichen Standards benötigen. Zudem vereinfacht sie das Management von Daten in

verschiedenen Umgebungen, sowohl vor Ort als auch in der Cloud. Benutzer können sich darauf verlassen, dass ihre Daten während dieser Prozesse vollständig und unverändert sind.

NFSv4.2 Security Labels haben die in definierten EinschränkungenMit NFSv4.2 gekennzeichnet.

Beispiele für die Kontrolle des Zugriffs auf Daten

Der folgende Beispieleintrag für Daten, die in John R Smiths PKI-Zertifikat gespeichert sind, zeigt, wie der Ansatz von NetApp auf eine Datei angewendet werden kann und eine feingranulare Zugriffskontrolle bietet.

Diese Beispiele dienen zur Veranschaulichung, und es liegt in der Verantwortung der Regierung, zu definieren, welche Metadaten das Sicherheitslabel NFSv4.2 und die xattrs sind. Details zur Aktualisierung und Aufbewahrung von Etiketten werden aus einfachen Grund weggelassen.

Taste	Wert
EntitySecurityMark	t:s01 = NICHT KLASSIFIZIERT

Info	{
	"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"
	}
	}

Taste	Wert
Spezifikation	„DoD“
uuid	B4111349-7875-4115-ad30-0928565f2e15

AdminOrganisation	{
	"value": "DoD"
	}

Briefings	[
	{
	"value": "ABC1000"
	},
	{
	"value": "DEF1001"
	},
	{
	"value": "EFG2000"
	}
	]

Bürgerstatus	{
	"value": "US"
	}

Taste	Wert
Abstände	`[ { "value": "TS" }, { "value": "S" }, { "value": "C" }, { "value": "U" } ]`
LänderOfMitgliedschaften	`[ { "value": "USA" } ]`
DigitalIdentifier	`{ "classification": "UNCLASSIFIED", "value": "cn=smith john r jrsmith, ou=dod, o=u.s. government, c=us" }`
DissTos	`{ "value": "DoD" }`
DytOrganisation	`{ "value": "DoD" }`

Taste	Wert

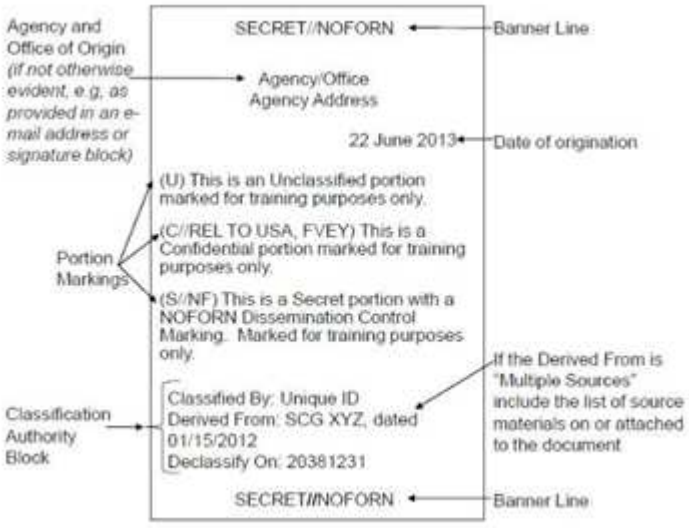
EntityType	<pre>{
	"value": "GOV"
	}</pre>

FineAccessControls	<pre>[
	{
	"value": "SI"
	},
	{
	"value": "TK"
	},
	{
	"value": "NSYS"
	}
	]</pre>

Diese PKI-Berechtigungen zeigen die Zugangsdaten von John R. Smith, einschließlich des Zugriffs nach Datentyp und Zuordnung.

Wenn John R. Smith ein Dokument mit der Bezeichnung *„sample_analysis.doc“* erstellt und gespeichert hat, würde der Benutzer gemäß den entsprechenden Richtlinien-Anweisungen die entsprechenden Banner- und Portionsmarkierungen, die Agentur und das Ursprungsamt sowie den entsprechenden Baustein der Klassifizierungsbehörde basierend auf der Klassifizierung des Dokuments hinzufügen, wie im folgenden Bild gezeigt. Diese umfangreichen Metadaten sind nur verständlich, wenn sie von der Natural Language Processing (NLP) gescannt wurden und Regeln angewendet wurden, um Bedeutung aus den Markierungen zu machen. Tools wie die NetApp BlueXP -Klassifizierung können dies, sind jedoch weniger effizient für Entscheidungen zur Zugriffskontrolle, da sie die Berechtigung zum Einblicken in das Dokument benötigen.

Markierung für nicht klassifizierte CAPCO-Dokumentteile



In Szenarien, in denen IC-TDF-Metadaten getrennt von der Datei gespeichert werden, empfiehlt NetApp eine zusätzliche Ebene feingranularer Zugriffskontrolle. Dabei werden Informationen zur Zugriffssteuerung sowohl auf Verzeichnisebene als auch in Verbindung mit jeder Datei gespeichert. Betrachten Sie als Beispiel die folgenden Tags, die mit einer Datei verknüpft sind:

- NFSv4.2 Security Labels: Verwendet für Sicherheitsentscheidungen
- Xattrs: Geben Sie ergänzende Informationen, die für die Datei und die Anforderungen an das organisatorische Programm relevant sind

Die folgenden Schlüssel-Wert-Paare sind Beispiele für Metadaten, die als xattrs gespeichert werden können und detaillierte Informationen über den Ersteller der Datei und die zugehörigen Sicherheitsklassifizierungen bieten. Diese Metadaten können von den Client-Applikationen genutzt werden, um fundierte Zugriffsentscheidungen zu treffen und Dateien gemäß den Standards und Anforderungen des Unternehmens zu organisieren.

Taste	Wert
`user.uuid`	`"761d2e3c-e778-4ee4-997b-3bb9a6a1d3fa"`
`user.entitySecurityMark`	`"UNCLASSIFIED"`
`user.specification`	`"INFO"`

Taste	Wert

user.Info	{
	"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"
	},

Taste	Wert
user.geo_point	[-78.7941, 35.7956]

```
    }
}
```

Prüfen von Änderungen an Beschriftungen

Das Auditing von Änderungen an xattrs oder NFS-Sicherheitsetiketten ist ein wichtiger Aspekt der Verwaltung und Sicherheit von Dateisystemen. Standard-Dateisystemauditing-Tools ermöglichen die Überwachung und Protokollierung aller Änderungen an einem Dateisystem, einschließlich Änderungen an erweiterten Attributen und Sicherheitsbeschriftungen.

In Linux-Umgebungen wird der `auditd` Daemon häufig verwendet, um Auditing für Dateisystemereignisse einzurichten. Es ermöglicht Administratoren, Regeln zu konfigurieren, um auf bestimmte Systemaufrufe im Zusammenhang mit xattr-Änderungen zu achten, wie `setxattr`,, `lsetxattr` und `fsetxattr` um Attribute und, `lremovexattr` zu setzen `removexattr` und `fremovexattr` Attribute zu entfernen.

ONTAP FPolicy erweitert diese Funktionen durch ein robustes Framework für das Monitoring und die Kontrolle von Dateivorgängen in Echtzeit. FPolicy kann zur Unterstützung verschiedener xattr-Ereignisse konfiguriert werden. Dies ermöglicht eine granulare Kontrolle über Dateivorgänge und die Durchsetzung umfassender Datenmanagement-Richtlinien.

Bei Benutzern, die xattrs verwenden, insbesondere in NFSv3- und NFSv4-Umgebungen, werden für die Überwachung nur bestimmte Kombinationen von Dateioperationen und -Filtern unterstützt. Die Liste der unterstützten Dateioperationen und Filterkombinationen für das FPolicy Monitoring von NFSv3 und NFSv4-Dateizugriffsereignissen ist im Folgenden beschrieben:

Unterstützte Dateivorgänge	Unterstützte Filter
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

Beispiel eines auditd-Protokollausschlags für eine setattr-Operation:

```
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"
```

Die Aktivierung von ONTAP FPolicy für Benutzer, die mit xattrs arbeiten, stellt eine Schicht für Sichtbarkeit und Kontrolle dar, die für die Aufrechterhaltung der Integrität und Sicherheit des Filesystems unerlässlich ist.

Mithilfe der erweiterten Monitoring-Funktionen von FPolicy können Unternehmen sicherstellen, dass alle Änderungen an xattrs nachverfolgt, geprüft und an ihren Sicherheits- und Compliance-Standards ausgerichtet werden. Dieser proaktive Ansatz beim Filesystem-Management ist daher die Aktivierung von ONTAP FPolicy nur für Unternehmen empfehlenswert, die ihre Daten-Governance- und Sicherungsstrategien verbessern möchten.

Integration mit ABAC Identitäts- und Zugriffskontrollsoftware

Um die Funktionen der attributbasierten Zugriffskontrolle (ABAC) vollständig zu nutzen, kann ONTAP in eine ABAC-orientierte Identitäts- und Zugriffsmanagement-Software integriert werden.

> Parallel zu diesem Inhalt hat NetApp eine Referenzimplementierung mit Graubox. Eine Annahme für diesen Inhalt ist, dass die Identitäts-, Authentifizierungs- und Zugriffsdienste der Regierung mindestens einen Policy Enforcement Point (PEP) und einen Policy Decision Point (PDP) umfassen, der als Vermittler für den Zugriff auf das Dateisystem fungiert.

In einer praktischen Umgebung würde ein Unternehmen eine Mischung aus NFS-Sicherheitsetiketten und xattrs einsetzen. Diese werden verwendet, um eine Vielzahl von Metadaten darzustellen, einschließlich Klassifizierung, Sicherheit, Anwendung und Inhalt, die alle entscheidend für ABAC-Entscheidungen sind. XATTR kann zum Beispiel verwendet werden, um die Ressourcenattribute zu speichern, die der PDP für seinen Entscheidungsprozess verwendet. Ein Attribut kann definiert werden, um die Klassifizierungsstufe einer Datei darzustellen (z. B. „nicht klassifiziert“, „vertraulich“, „geheim“ oder „streng geheim“). Die PDP könnte dann dieses Attribut nutzen, um eine Richtlinie durchzusetzen, die Benutzern den Zugriff auf Dateien einschränkt, die eine Klassifizierungsstufe haben, die ihrem Sicherheitsniveau entspricht oder kleiner ist.

Beispiel für einen Prozessablauf für ABAC

1. Benutzer stellt Anmeldeinformationen (z. B. PKI, OAuth, SAML) für den Systemzugriff auf PEP bereit und ruft Ergebnisse von PDP ab.

   Die Rolle des PEP besteht darin, die Zugriffsanforderung des Benutzers abzufangen und an das PDP weiterzuleiten.

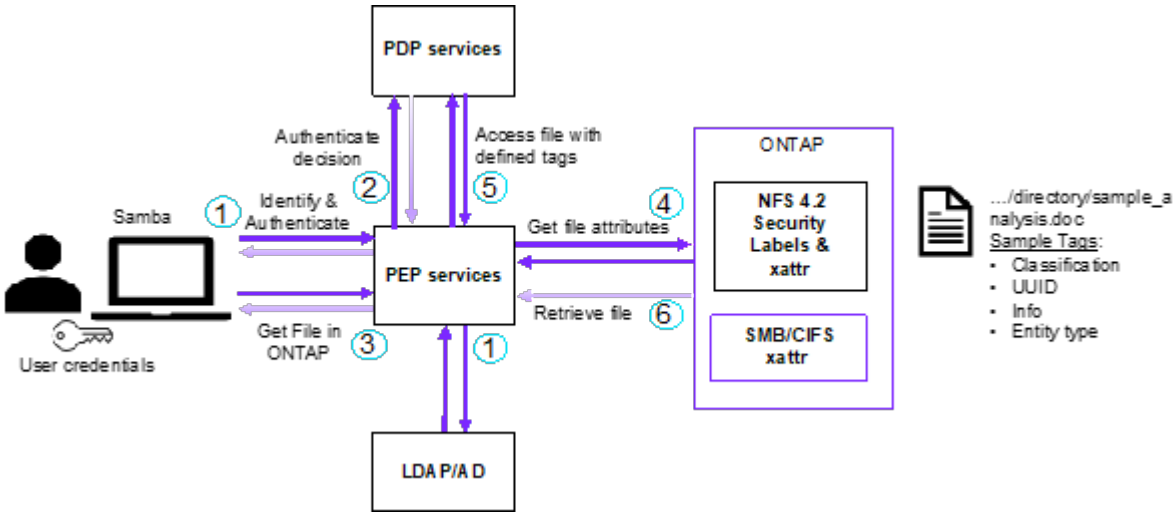
2. Die PDP wertet diese Anforderung dann anhand der festgelegten ABAC-Richtlinien aus.

   In diesen Richtlinien werden verschiedene Attribute berücksichtigt, die sich auf den Benutzer, die betreffende Ressource und die Umgebung beziehen. Auf der Grundlage dieser Richtlinien trifft die PDP eine Zugriffsentscheidung, entweder zuzulassen oder abzulehnen, und teilt diese Entscheidung dann dem PEP zurück.

   PDP stellt PEP Richtlinien zur Durchsetzung bereit. Der PEP erzwingt dann diese Entscheidung, indem er die Zugriffsanfrage des Benutzers gemäß der Entscheidung des PDP entweder gewährt oder ablehnt.

3. Nach einer erfolgreichen Anfrage fordert der Benutzer eine in ONTAP gespeicherte Datei an (z. B. AFF, AFF-C).
4. Wenn die Anforderung erfolgreich war, erhält PEP fein abgestufte Zugangskontroll-Tags aus dem Dokument.
5. PEP fordert die Richtlinie für den Benutzer auf Grundlage der Zertifikate dieses Benutzers an.
6. PEP trifft eine Entscheidung auf der Grundlage von Richtlinien und Tags, wenn der Benutzer Zugriff auf die Datei hat, und lässt den Benutzer die Datei abrufen.

> Der eigentliche Zugriff kann mit Token erfolgen, die nicht über Proxy-Server bereitgestellt werden.



Verwandte Informationen

- "NFS in NetApp ONTAP: Best Practice und Implementierungsleitfaden"
- Anforderung von Kommentaren (RFC)
  - RFC 2203: RPCSEC_GSS-Protokollspezifikation
  - RFC 3530: Network File System (NFS) Version 4 Protocol