속성 기반 액세스 제어

ONTAP 9

NetApp
January 17, 2025

목차

속성 기반 액세스 제어

ONTAP로 속성 기반 액세스 제어

ONTAP를 사용하여 속성 및 속성 기반 액세스 제어(ABAC)로 향상된 RBAC를 구현할 수 있습니다. ONTAP는 NFS 4.2 및 SMB/CIFS를 사용하는 XATTRS 등 고객이 파일 수준 ABAC를 달성하기 위해 사용할 수 있는 몇 가지 접근 방식을 제공합니다.

ABAC(속성 기반 액세스 제어)는 사용자 속성, 리소스 속성 및 환경 조건을 고려하는 정교한 액세스 권한 관리 방법입니다. NIST(National Institute of Standards and Technology)는 ABAC 표준을 확립하여 안전하고 일관된 구현을 위한 프레임워크를 제공합니다.

ONTAP 9.12.1부터 ONTAP가 RBAC(역할 기반 액세스 제어) 및 ABAC(속성 기반 액세스 제어) ID와 통합될 수 있도록 NFSv4.2 보안 레이블 및 확장 특성(XATTRS)을 사용하여 구성할 수 있습니다. 이러한 통합을 통해 ONTAP는 NIST ABAC 준수 데이터 관리 솔루션으로 분류되는 제어 소프트웨어에 액세스할 수 있으므로 PEP(Policy Enforcement Point), PDP(Policy Decision Point) 및 사용자, 리소스, 환경과 관련된 특성을 고려하는 정책을 비롯한 복잡한 환경에서 액세스 권한을 관리할 수 있는 강력하고 진보된 접근 방식을 제공합니다.

NetApp ONTAP와 확장 특성(XATTRS) 및 속성 기반 액세스 제어(ABAC) 소프트웨어의 통합은 NIST 특별 간행물 800-162에 명시된 지침에 따라 ABAC 구현을 위한 NIST 표준을 준수합니다. NFS 4.2 보안 레이블 및 XATTRS를 사용하면 파일에 사용자 정의 속성을 연결할 수 있으며, 액세스 제어 의사 결정에 리소스 속성을 고려하기 위한 NIST ABAC 표준의 요구 사항을 충족합니다. ABAC 소프트웨어의 PEP 및 PDP는 액세스 제어 프로세스에서 이러한 구성 요소에 대한 NIST ABAC 표준의 요구 사항에 부합합니다. 여러 특성과 조건을 고려하는 복잡한 정책을 정의하는 기능은 NIST ABAC 표준의 정책 기반 액세스 제어 요구 사항에 부합합니다.

관련 정보

- "ABAC에 대한 ONTAP의 접근 방식"
- "NFS in NetApp ONTAP: 모범 사례 및 구축 가이드"
- 설명 요청(RFC)
  - RFC 2203: RPCSEC_GSS 프로토콜 사양
  - RFC 3530: NFS(Network File System) 버전 4 프로토콜

ABAC에 대한 ONTAP의 접근 방식

ONTAP는 NFS 및 SMB/CIFS를 사용하는 NFSv4.2 및 XATTRS를 비롯하여 파일 레벨 ABAC를 달성하기 위해 고객이 사용할 수 있는 다양한 접근 방식을 제공합니다.

레이블이 지정된 NFSv4.2

ONTAP 9.9.1부터 NFS라는 NFSv4.2 기능이 지원됩니다.

레이블이 지정된 NFS는 SELinux 레이블 및 MAC(Mandatory Access Control)를 사용하여 세분화된 파일 및 폴더 액세스를 관리하는 방법입니다. 이러한 MAC 레이블은 파일과 폴더에 저장되며 UNIX 사용 권한 및 NFSv4.x ACL과 함께 작동합니다.

레이블이 지정된 NFS에 대한 지원은 이제 ONTAP가 NFS 클라이언트의 SELinux 레이블 설정을 인식하고 이해한다는 의미입니다. 레이블이 지정된 NFS는 RFC-7204에 설명되어 있습니다.

레이블이 지정된 NFSv4.2의 활용 사례는 다음과 같습니다.

- 가상 머신(VM) 이미지의 MAC 레이블 지정
- 공공 부문의 데이터 보안 분류(비밀, 최고 비밀 및 기타 분류)
- 보안 규정 준수
- 디스크 없는 Linux

레이블이 지정된 **NFSv4.2**를 설정합니다

다음 고급 권한 옵션을 사용하여 레이블이 지정된 NFS를 설정하거나 해제할 수 있습니다.

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

이 매개 변수는 선택 사항이며 기본 설정은 `disabled` 입니다.

레이블이 지정된 **NFSv4.2**에 대한 적용 모드입니다

ONTAP 9.9.1부터 ONTAP는 다음 적용 모드를 지원합니다.

- 제한된 서버 모드: ONTAP는 라벨을 적용할 수 없지만 라벨을 저장 및 전송할 수 있습니다.

  MAC 레이블을 변경하는 기능은 클라이언트가 적용할 수도 있습니다.

- * 게스트 모드 *: 클라이언트가 NFS-Aware(v4.1 이하)로 표시되지 않으면 MAC 레이블이 전송되지 않습니다.

ONTAP는 현재 전체 모드를 지원하지 않습니다(MAC 레이블 저장 및 적용).

레이블이 지정된 **NFSv4.2**의 구성 예

다음 예제 구성은 Red Hat Enterprise Linux 릴리스 9.3(Plow)을 사용하는 개념을 보여 줍니다.

John R. Smith의 자격 증명을 기반으로 생성된 사용자는 `jrsmith` 다음과 같은 계정 Privileges를 가지고 있습니다.

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

다음 MLS Privileges 표에 설명된 대로 권한이 있는 사용자인 관리자 계정과 사용자라는 두 가지 역할이 있습니다. `jrsmith`

사용자	역할	유형	레벨
admins	sysadm_r	sysadm_t	t:s0
jrsmith	user_r	user_t	t:s1 - t:s4

이 예제 환경에서는 사용자가 `jrsmith` 에 있는 `s3` 수준의 파일에 액세스할 수 `s0` 있습니다. 관리자가 사용자 관련

데이터에 액세스하지 못하도록 하기 위해 아래에 설명된 대로 기존 보안 분류를 개선할 수 있습니다.

- S0 = 권한 관리자 사용자 데이터
- S0 = 분류되지 않은 데이터
- S1 = 대외비
- S2 = 비밀 데이터
- S3 = 상위 암호 데이터

조직의 보안 정책을 따릅니다

MCS를 사용하는 **NFSv4.2** 보안 레이블의 예

MLS(다중 수준 보안) 외에도 MCS(다중 범주 보안)라는 또 다른 기능을 사용하여 프로젝트와 같은 범주를 정의할 수 있습니다.

NFS 보안 레이블	값
`entitySecurityMark`	`t:s01 = UNCLASSIFIED`

확장 속성(XATTRS)

ONTAP 9.12.1부터 ONTAP는 xattrs.xattrs를 지원하므로 ACL(액세스 제어 목록) 또는 사용자 정의 속성과 같이 시스템에서 제공하는 것 이상의 파일 및 디렉터리와 메타데이터를 연결할 수 있습니다.

xattrs를 구현하려면 Linux에서 및 `getfattr` 명령줄 유틸리티를 사용하여 파일 시스템 객체의 xattrs를 관리할 수 `setfattr` 있습니다. 이러한 도구는 파일과 디렉토리에 대한 추가 메타데이터를 관리하는 강력한 방법을 제공합니다. 부적절하게 사용하면 예기치 않은 동작이나 보안 문제가 발생할 수 있으므로 주의해서 사용해야 합니다. 자세한 사용 지침은 항상 `setfattr` 및 `getfattr` man 페이지 또는 기타 신뢰할 수 있는 문서를 참조하십시오.

ONTAP 파일 시스템에서 xattrs가 활성화된 경우 사용자는 파일에 대한 임의의 속성을 설정, 수정 및 검색할 수 있습니다. 이러한 특성은 액세스 제어 정보와 같은 표준 파일 속성 집합으로 캡처되지 않은 파일에 대한 추가 정보를 저장하는 데 사용할 수 있습니다.

ONTAP에서 **xattrs**를 사용하기 위한 요구 사항

- Red Hat Enterprise Linux 8.4 이상
- Ubuntu 22.04 이상
- 각 파일에는 최대 128개의 xattrs를 포함할 수 있습니다
- xattr 키는 255바이트로 제한됩니다
- 결합된 키 또는 값 크기는 xattr 당 1,729바이트입니다
- 디렉터리 및 파일에는 xattrs가 있을 수 있습니다
- xattrs를 설정 및 검색하려면 `w` 사용자 및 그룹에 대해 쓰기 모드 비트를 활성화해야 합니다

xattrs에 대한 사용 사례

xattrs는 사용자 네임스페이스 내에서 활용되며 ONTAP 자체에 내재된 의미를 부여하지 않습니다. 대신 실제 애플리케이션은 파일 시스템과 상호 작용하는 클라이언트측 애플리케이션에 의해 결정되고 관리됩니다.

xattr 사용 사례 예:

- 파일 생성을 담당하는 응용 프로그램의 이름을 기록합니다.
- 파일을 가져온 이메일 메시지에 대한 참조 유지 관리
- 파일 객체 구성을 위한 범주화 프레임워크 설정
- 원본 다운로드 소스의 URL로 파일 레이블 지정

xattrs 관리 명령입니다

- `setfattr`: 파일 또는 디렉토리의 확장 속성을 설정합니다.

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

  명령 예:

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

- `getfattr`: 특정 확장 특성의 값을 검색하거나 파일이나 디렉터리의 모든 확장 특성을 나열합니다.

  특정 속성: `getfattr -n <attribute_name> <file or directory name>`

  모든 속성: `getfattr <file or directory name>`

  명령 예:

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

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

확장 특성에 대한 **ACE**를 사용하는 사용자 권한

ACE(액세스 제어 항목)는 ACL(액세스 제어 목록) 내의 구성 요소로, 파일 또는 디렉터리와 같은 특정 리소스에 대해 개별 사용자 또는 사용자 그룹에 부여된 액세스 권한이나 권한을 정의합니다. 각 ACE는 허용 또는 거부된 액세스 유형을 지정하며 특정 보안 주체(사용자 또는 그룹 ID)와 연결됩니다.

파일 형식	**xattr**를 검색합니다	**xattrs**를 설정합니다
파일	R	a, w, T, 키
디렉토리	R	T

xattrs에 필요한 권한에 대한 설명:

- xattr 검색 * : 사용자가 파일이나 디렉토리의 확장된 속성을 읽는 데 필요한 권한입니다. "R"은 읽기 권한이

필요하다는 것을 나타냅니다. * xattrs * 설정: 확장 속성을 수정하거나 설정하는 데 필요한 권한입니다. "a,""w" 및 "T"는 추가, 쓰기 및 xattrs와 관련된 특정 사용 권한 등 다양한 사용 권한의 예를 나타냅니다. * 파일 **:** 사용자는 확장 속성을 설정하려면 추가**,** 쓰기 및 **xattrs**와 관련된 특별 권한이 필요합니다**.** *디렉터리: 확장 특성을 설정하려면 특정 권한 "T"가 필요합니다.

xattrs에 대한 SMB/CIFS 프로토콜 지원

SMB/CIFS 프로토콜에 대한 ONTAP의 지원은 Windows 환경에서 파일 메타데이터의 필수적인 부분인 xattrs를 포괄적으로 처리할 수 있도록 확장되었습니다. 확장 특성을 사용하면 사용자 및 응용 프로그램에서 만든 이 세부 정보, 사용자 지정 보안 설명자 또는 응용 프로그램별 데이터와 같은 표준 파일 특성 집합 이상의 추가 정보를 저장할 수 있습니다. ONTAP의 SMB/CIFS 구현은 이러한 xattrs가 완벽하게 지원되므로 기능 및 정책 적용을 위해 메타데이터에 의존하는 Windows 서비스 및 애플리케이션과 원활하게 통합할 수 있습니다.

ONTAP에서 관리하는 SMB/CIFS 공유를 통해 파일을 액세스하거나 전송할 때 시스템은 xattrs의 무결성을 유지하여 모든 메타데이터가 보존되고 일관성을 유지합니다. 이는 보안 설정을 유지 관리하고 구성 또는 작업에 xattrs를 사용하는 응용 프로그램에 특히 중요합니다. ONTAP는 SMB/CIFS 컨텍스트 내에서 xattrs를 강력하게 처리하므로 서로 다른 플랫폼 및 환경 간에 파일을 공유할 수 있으며, 사용자에게 원활한 환경을 제공하고 관리자가 데이터 거버넌스 정책을 준수할 수 있습니다. 협업, 데이터 아카이빙, 규정 준수 등 어떤 용도로 사용하든 SMB/CIFS 공유 내에서 ONTAP가 관심을 기울이는 것은 혼합 OS 환경에서 탁월한 데이터 관리와 상호 운용성에 대한 약속입니다.

ABAC의 PEP(Policy Enforcement Point) 및 PDP(Policy Decision Point)입니다

ABAC(속성 기반 액세스 제어) 시스템에서 PEP(정책 적용 지점)와 PDP(정책 결정 지점)가 중요한 역할을 합니다. PEP는 액세스 제어 정책을 적용하는 역할을 담당하며 PDP는 정책에 따라 액세스 허용 또는 거부 여부를 결정합니다.

제공된 Python 코드 스니펫의 맥락에서, 스크립트 자체는 PEP의 역할을 한다. 이 명령은 파일을 열고 내용을 읽어 파일에 대한 액세스 권한을 부여하거나 를 발생시켜 액세스를 거부함으로써 액세스 제어 결정을 `PermissionError` 적용합니다.

반면 PDP는 기본 SELinux 시스템의 일부가 될 것입니다. 스크립트가 특정 SELinux 컨텍스트를 사용하여 파일을 열려고 하면 SELinux 시스템은 해당 정책을 확인하여 액세스 허용 또는 거부 여부를 결정합니다. 그런 다음 이 결정은 스크립트에 의해 적용됩니다.

다음은 ABAC 환경에서 이 코드가 작동하는 방식을 단계별로 설명하는 예제입니다.

1. 스크립트는 함수를 사용하여 `selinux.setcon()` SELinux 컨텍스트를 컨텍스트로 `jrsmith` 설정합니다. 이는 파일에 액세스하는 것과 `jrsmith` 같습니다.
2. 스크립트가 파일을 열려고 합니다. 이 부분에서 PEP가 사용됩니다.
3. SELinux 시스템은 해당 정책을 검사하여 (또는 더 구체적으로 SELinux 컨텍스트가 있는 사용자 `jrsmith`) 파일에 액세스할 수 있는지 `jrsmith` 확인합니다. PDP의 역할입니다.
4. 가 파일에 액세스할 수 있는 경우 `jrsmith` SELinux 시스템은 스크립트가 파일을 열 수 있도록 하고 스크립트는 파일의 내용을 읽고 인쇄합니다.
5. 가 파일에 액세스할 수 없는 경우 `jrsmith` SELinux 시스템은 스크립트가 파일을 열 수 없도록 하고 스크립트가 를 발생시킵니다. `PermissionError`
6. 스크립트는 임시 컨텍스트 변경이 다른 작업에 영향을 미치지 않도록 원래 SELinux 컨텍스트를 복원합니다.

python을 사용하여 컨텍스트를 가져오는 코드는 아래에 표시되며 여기서 변수 파일 경로는 검사할 문서입니다.

```
#Get the current context

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

ONTAP 클론 복제 및 SnapMirror

ONTAP의 클론 생성 및 SnapMirror 기술은 효율적이고 안정적인 데이터 복제 및 복제 기능을 제공하도록 설계되었으며, 확장된 속성(xattrs)을 포함한 파일 데이터의 모든 측면을 보존하고 파일과 함께 전송합니다. xattrs는 보안 레이블, 액세스 제어 정보, 사용자 정의 데이터 등 파일과 관련된 추가 메타데이터를 저장하는 데 있어 매우 중요합니다.

ONTAP의 FlexClone 기술을 사용하여 볼륨을 클론 복제하면 볼륨의 쓰기 가능한 정확한 복제본이 생성됩니다. 이 복제 프로세스는 즉각적이고 공간 효율적이며 모든 파일 데이터와 메타데이터가 포함되어 xattrs가 완전히 복제되도록 합니다. 마찬가지로, SnapMirror는 데이터가 완벽한 충실도로 보조 시스템에 미러링되도록 보장합니다. 여기에는 이 메타데이터에 의존하는 응용 프로그램이 올바르게 작동하는 데 중요한 xattrs가 포함됩니다.

NetApp ONTAP는 클론 복제 및 복제 작업에 xattrs를 포함함으로써 모든 특성을 갖춘 전체 데이터 세트를 운영 및 2차 스토리지 시스템에서 일관되게 사용할 수 있도록 보장합니다. 일관된 데이터 보호, 빠른 복구, 규정 준수 및 규정 준수 표준을 준수해야 하는 조직에는 이러한 포괄적인 데이터 관리 접근 방식이 필수적입니다. 또한 온프레미스와 클라우드에서 다양한 환경에서 데이터 관리를 간소화하여 이러한 프로세스 중에 데이터가 완전하고 변경되지 않았다는 확신을 사용자에게 제공합니다.

NFSv4.2 보안 레이블에는 에 정의된 주의 사항이 레이블이 지정된 NFSv4.2있습니다.

데이터에 대한 액세스를 제어하는 예

John R Smith의 PKI 인증서에 저장된 데이터에 대한 다음 예제 항목은 NetApp의 접근 방식을 파일에 적용하고 세분화된 액세스 제어를 제공하는 방법을 보여 줍니다.

이러한 예는 설명을 위한 것이며 NFSv4.2 보안 레이블 및 xattrs 메타데이터를 정의하는 것은 정부의 책임입니다. 업데이트 및 레이블 보존에 대한 자세한 내용은 간단한 사용을 위해 생략됩니다.

키	값
entitySecurityMark를 클릭합니다	T:s01 = 분류되지 않음

키	값

정보	{
	"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"
	}
	}

사양	"DoD"
UUID입니다	b4111349-7875-4115-AD30-0928565f2e15

관리자 조직	{
	"value": "DoD"
	}

키	값

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

시민 상태	{
	"value": "US"
	}

여유값	[
	{
	"value": "TS"
	},
	{
	"value": "S"
	},
	{
	"value": "C"
	},
	{
	"value": "U"
	}
	]

국가/지역 제휴	[
	{
	"value": "USA"
	}
	]

키	값

디지털 식별자입니다	<pre>{
	  "classification": "UNCLASSIFIED",
	  "value": "cn=smith john r jrsmith, ou=dod, o=u.s.
	government, c=us"
	}</pre>

파종	<pre>{
	  "value": "DoD"
	}</pre>

DutyOrganization(이중 조직	<pre>{
	  "value": "DoD"
	}</pre>

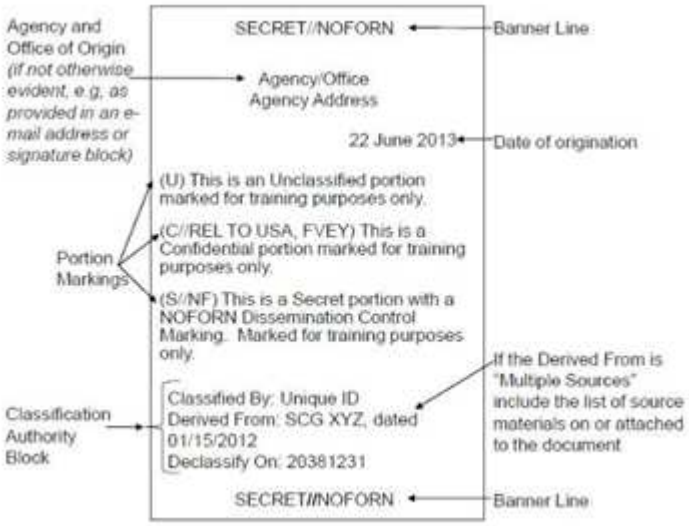
entityType 을 선택합니다	<pre>{
	  "value": "GOV"
	}</pre>

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

이러한 PKI 권한은 데이터 유형 및 특성을 포함한 John R. Smith의 액세스 세부 정보를 보여 줍니다.

John R. Smith가 관련 정책 지침 발급에 따라 _ "sample_analysis.doc" _ 라는 문서를 작성하여 저장한 경우 사용자는 다음 이미지에 표시된 대로 문서의 분류에 따라 적절한 배너 및 부분 표시, 기관 및 원산지 사무소, 적절한 분류 기관 블록을 추가합니다. 이 풍부한 메타데이터는 NLP(Natural Language Processing)로 스캔하고 표시에서 의미를 만들기 위해 규칙을 적용한 후에만 이해할 수 있습니다. NetApp BlueXP 분류와 같은 도구는 이러한 작업을 수행할 수 있지만 문서 내부를 보기 위한 권한이 필요하기 때문에 액세스 제어 의사 결정에 효율성이 떨어집니다.

분류되지 않은 **CAPCO** 문서 부분 표시



IC-TDF 메타데이터가 파일과 별도로 저장되는 시나리오에서 NetApp는 세분화된 액세스 제어 계층을 추가로 지원합니다. 여기에는 디렉토리 레벨 및 각 파일과 관련된 액세스 제어 정보가 모두 저장됩니다. 예를 들어, 파일에 연결된 다음 태그를 고려해 보십시오.

- NFSv4.2 보안 레이블: 보안 결정을 내리는 데 사용됩니다
- xattrs: 파일 및 조직 프로그램 요구 사항과 관련된 보충 정보를 제공합니다

다음 키-값 쌍은 xattrs로 저장될 수 있는 메타데이터의 예이며 파일의 생성자 및 관련 보안 분류에 대한 자세한 정보를 제공합니다. 이 메타데이터는 클라이언트 응용 프로그램에서 정보에 기반한 액세스 결정을 내리고 조직의 표준 및 요구 사항에 따라 파일을 구성하는 데 활용될 수 있습니다.

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

키	값

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"
	},

키	값
user.geo_point	[-78.7941, 35.7956]

```
    }
}
```

라벨에 대한 변경 감사

xattrs 또는 NFS 보안 레이블의 변경 사항을 감사하는 것은 파일 시스템 관리 및 보안의 중요한 부분입니다. 표준 파일 시스템 감사 툴을 사용하면 확장된 특성 및 보안 레이블 수정을 비롯하여 파일 시스템에 대한 모든 변경 사항을 모니터링하고 기록할 수 있습니다.

Linux 환경에서 `auditd` 데몬은 일반적으로 파일 시스템 이벤트에 대한 감사를 설정하는 데 사용됩니다. 관리자는, `lsetxattr` 등의 xattr 변경과 관련된 특정 시스템 호출을 감시하고 `fsetxattr`, 특성을 설정하고 `removexattr`, `lremovexattr` `fremovexattr` 속성을 제거하는 규칙을 구성할 수 `setxattr` 있습니다.

ONTAP FPolicy는 파일 작업을 실시간으로 모니터링하고 제어하기 위한 강력한 프레임워크를 제공하여 이러한 기능을 확장합니다. 다양한 xattr 이벤트를 지원하도록 FPolicy를 구성하여 파일 작업을 세부적으로 제어하고 포괄적인 데이터 관리 정책을 적용할 수 있습니다.

xattrs를 사용하는 사용자, 특히 NFSv3 및 NFSv4 환경에서는 특정 파일 작업 및 필터 조합만 모니터링하도록 지원됩니다. NFSv3 및 NFSv4 파일 액세스 이벤트의 FPolicy 모니터링을 위해 지원되는 파일 작업 및 필터 조합 목록은 아래에 자세히 설명되어 있습니다.

지원되는 파일 작업	지원되는 필터
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

SetAttr 작업에 대한 **auditd** 로그 스니펫의 예:

```
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"
```

xattrs로 작업하는 사용자를 위한 ONTAP FPolicy를 활성화함으로써 파일 시스템의 무결성 및 보안을 유지하는 데 필수적인 가시성과 제어 계층을 확보할 수 있습니다. FPolicy의 고급 모니터링 기능을 활용하면 xattrs에 대한 모든 변경 사항을 추적하고 감사하며 보안 및 규정 준수 표준에 부합하도록 할 수 있습니다. 파일 시스템 관리에 대한 이러한 사전 예방적 접근 방식 때문에 데이터 거버넌스 및 보호 전략을 개선하려는 모든 조직에 ONTAP FPolicy를 사용하도록 적극 권장합니다.

ABAC ID 및 액세스 제어 소프트웨어와의 통합

ABAC(속성 기반 액세스 제어)의 기능을 최대한 활용하기 위해 ONTAP는 ABAC 지향 ID 및 액세스 관리 소프트웨어와 통합할 수 있습니다.

> 이 콘텐츠와 병행하여 NetApp는 GreyBox를 사용하여 참조 구현을 제공합니다. 이 콘텐츠의 한 가지 가정은 정부의 ID, 인증 및 액세스 서비스에 최소한 파일 시스템에 대한 액세스를 위한 중개인 역할을 하는 정책 적용 지점(PEP)과 정책 결정 지점(PDP)이 포함된다는 것입니다.

실용적인 환경에서 조직은 NFS 보안 레이블과 xattrs를 혼합하여 사용할 수 있습니다. 이러한 메타데이터는 분류, 보안, 애플리케이션 및 콘텐츠를 포함한 다양한 메타데이터를 나타내는 데 사용되며, 이러한 메타데이터는 모두 ABAC 결정을 내리는 데 중요한 역할을 합니다. 예를 들어, XATTR을 사용하여 PDP가 의사 결정 프로세스에 사용하는 리소스 속성을 저장할 수 있습니다. 파일의 분류 수준(예: "분류되지 않음", "기밀", "비밀" 또는 "최고 비밀")을 나타내도록 속성을 정의할 수 있습니다. 그런 다음 PDP는 이 속성을 활용하여 사용자가 분류 수준이 허용 수준 이하인 파일만 액세스하도록 제한하는 정책을 적용할 수 있습니다.

ABAC에 대한 프로세스 흐름의 예

1. 사용자가 PEP에 대한 시스템 액세스에 대한 자격 증명(예: PKI, OAuth, SAML)을 제공하고 PDP에서 결과를 가져옵니다.

   PEP의 역할은 사용자의 액세스 요청을 가로채서 PDP로 전달하는 것입니다.

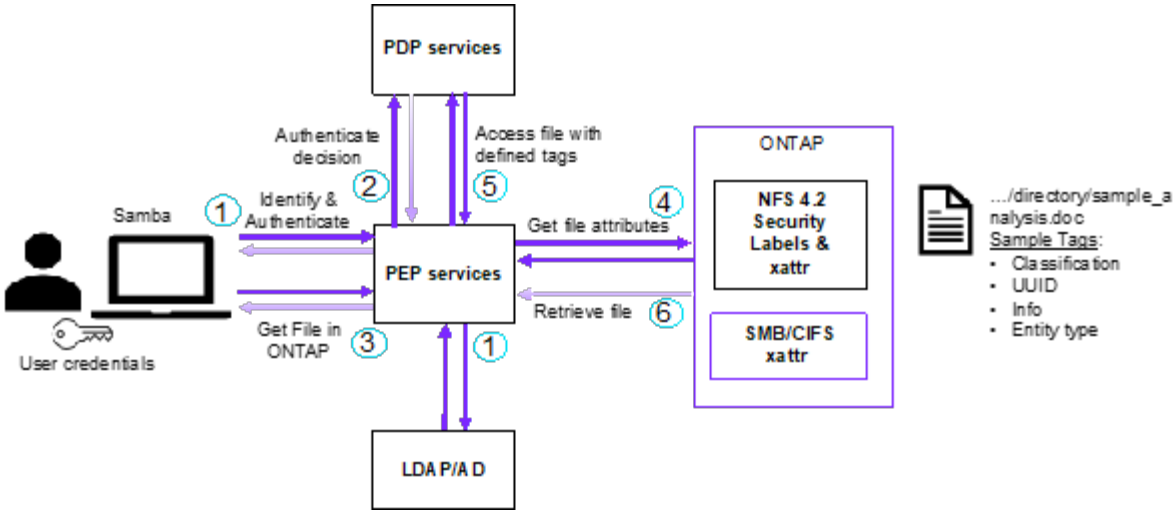
2. 그런 다음 PDP는 설정된 ABAC 정책에 대해 이 요청을 평가합니다.

   이러한 정책에서는 사용자, 해당 리소스 및 주변 환경과 관련된 다양한 특성을 고려합니다. 이러한 정책에 따라 PDP는 액세스 권한을 허용하거나 거부하도록 결정한 다음 이 결정을 다시 PEP에 전달합니다.

   PDP는 PEP에 적용할 정책을 제공합니다. 그런 다음 PEP는 PDP의 결정에 따라 사용자의 액세스 요청을 허용하거나 거부하여 이 결정을 적용합니다.

3. 요청이 성공하면 사용자는 ONTAP에 저장된 파일(예: AFF, AFF-C)을 요청합니다.
4. 요청이 성공하면 PEP는 문서에서 미세 입자 액세스 제어 태그를 가져옵니다.
5. PEP는 해당 사용자의 인증서를 기반으로 사용자에 대한 정책을 요청합니다.
6. PEP는 사용자가 파일에 액세스할 수 있고 사용자가 파일을 검색할 수 있는 경우 정책 및 태그에 따라 결정합니다.

> 실제 액세스는 프록시가 아닌 토큰을 사용하여 수행할 수 있습니다.



관련 정보

- "NFS in NetApp ONTAP: 모범 사례 및 구축 가이드"
- 설명 요청(RFC)
  - RFC 2203: RPCSEC_GSS 프로토콜 사양
  - RFC 3530: NFS(Network File System) 버전 4 프로토콜