



Attribute-based access control

ONTAP Technical Reports

NetApp
January 23, 2026

This PDF was generated from <https://docs.netapp.com/us-en/ontap-technical-reports/abac/abac-overview.html> on January 23, 2026. Always check docs.netapp.com for the latest.

Table of Contents

- Attribute-based access control 1
 - Attribute-based access control with ONTAP 1
- Approaches to attribute-based access control (ABAC) in ONTAP 1
 - NFS v4.2 security labels 1
 - Extended attributes (xattrs) 3
 - Integration with ABAC identity and access control software 5
 - ONTAP cloning and SnapMirror 6
 - Auditing changes to labels 6
 - Examples of controlling access to data 7

Attribute-based access control

Attribute-based access control with ONTAP

Beginning with 9.12.1, you can configure ONTAP with NFSv4.2 security labels and extended attributes (xattrs) to support role-based access control (RBAC) with attributes and attribute-based access control (ABAC).

ABAC is an authorization strategy that defines permissions based on user attributes, resource attributes, and environmental conditions. The integration of ONTAP with NFS v4.2 security labels and xattrs complies with NIST standards for ABAC solutions, as set forth in NIST Special Publication 800-162.

You can use NFS v4.2 security labels and xattrs to assign files user-defined attributes and labels. ONTAP can integrate with ABAC-oriented identity and access management software to enforce granular file and folder access control policies based on these attributes and labels.

Related information

- [Approaches to ABAC with ONTAP](#)
- [NFS in NetApp ONTAP: Best practice and implementation guide](#)

Approaches to attribute-based access control (ABAC) in ONTAP

ONTAP provides several approaches you can use to achieve file-level attribute-based access control (ABAC), including NFS v4.2 security labels and extended attributes (xattrs) using NFS.

NFS v4.2 security labels

Beginning with ONTAP 9.9.1, the NFS v4.2 feature called Labeled NFS is supported.

NFS v4.2 security labels are a way to manage granular file and folder access by using SELinux labels and Mandatory Access Control (MAC). These MAC labels are stored with files and folders, and they work in conjunction with UNIX permissions and NFS v4.x ACLs.

Support for NFS v4.2 security labels means that ONTAP now recognizes and understands the NFS client's SELinux label settings. NFS v4.2 security labels are covered in RFC-7204.

Use cases for NFS v4.2 security labels include the following:

- MAC labeling of virtual machine (VM) images
- Data security classification for the public sector (secret, top secret, and other classifications)
- Security compliance
- Diskless Linux

Enable NFS v4.2 security labels

You can enable or disable NFS v4.2 security labels with the following command (advanced privilege required):

```
vserver nfs modify -vserver <svm_name> -v4.2-seclabel <disabled|enabled>
```

Learn more about `vserver nfs modify` in the [ONTAP command reference](#).

Enforcement modes for NFS v4.2 security labels

Beginning with ONTAP 9.9.1, ONTAP supports the following enforcement modes:

- **Limited Server Mode:** ONTAP cannot enforce the labels but can store and transmit them.



The ability to change MAC labels is up to the client to enforce.

- **Guest Mode:** If the client is not labeled NFS-aware (v4.1 or lower), MAC labels are not transmitted.



ONTAP does not currently support Full Mode (storing and enforcing MAC labels).

NFS v4.2 security labels examples

The following example configuration demonstrates concepts using Red Hat Enterprise Linux release 9.3 (Plow).

The user `jrsmith`, created based on John R. Smith's credentials, has the following account privileges:

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

There are two roles: the admin account that is a privileged user and user `jrsmith` as described in the following MLS privileges table:

Users	Role	Type	Levels
admins	sysadm_r	sysadm_t	t:s0
jrsmith	user_r	user_t	t:s1 - t:s4

In this example environment, user `jrsmith` has access to files at the levels `s0` to `s3`. We can enhance the existing security classifications, as outlined below, to ensure that administrators do not have access to user-specific data.

- `s0` = privilege admin user data
- `s0` = unclassified data
- `s1` = confidential
- `s2` = secret data
- `s3` = top secret data

NFS v4.2 security labels example with MCS

In addition to Multi-Level Security (MLS), another capability called Multi-Category Security (MCS) allows you to define categories such as projects.

NFS security label	Value
entitySecurityMark	t:s01 = UNCLASSIFIED

Extended attributes (xattrs)

Beginning with ONTAP 9.12.1, ONTAP supports xattrs. xattrs allow metadata to be associated with files and directories beyond what is provided by the system, such as access control lists (ACLs) or user-defined attributes.

To implement xattrs, you can use `setfattr` and `getfattr` command-line utilities in Linux. These tools provide a powerful way to manage additional metadata for files and directories. They should be used with care, as improper use can lead to unexpected behavior or security issues. Always refer to the `setfattr` and `getfattr` man pages or other reliable documentation for detailed usage instructions.

When xattrs is enabled on an ONTAP filesystem, users can set, modify, and retrieve arbitrary attributes on files. These attributes can be used to store additional information about the file that is not captured by the standard set of file attributes, such as access control information.

There are several requirements and limits for using xattrs in ONTAP:

- Red Hat Enterprise Linux 8.4 or later
- Ubuntu 22.04 or later
- Each file can have up to 128 xattrs
- Xattr keys are limited to 255 bytes
- The combined key or value size is 1,729 bytes per xattr
- Directories and files can have xattrs
- To set and retrieve xattrs, `w` or write mode bits must be enabled for the user and group

Xattrs are utilized within the user namespace and do not carry any intrinsic significance to ONTAP itself. Instead, their practical applications are determined and managed exclusively by the client-side application that interacts with the file system.

Xattr use case examples:

- Recording the name of the application responsible for creating a file
- Maintaining a reference to the email message from which a file was obtained
- Establishing a categorization framework for organizing file objects
- Labeling files with the URL of their original download source

Commands for managing xattrs

- `setfattr` sets an extended attribute of a file or directory:

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

Sample command:

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

- **getfattr** retrieves the value of a specific extended attribute or lists all extended attributes of a file or directory:

Specific attribute: `getfattr -n <attribute_name> <file or directory name>`

All attributes: `getfattr <file or directory name>`

Sample command:

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

Xattr key value pair examples

The following table shows two xattr key value pair examples:

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

User permissions with ACE for xattrs

An access control entry (ACE) is a component within an ACL that defines the access rights or permissions granted to an individual user or a group of users for a specific resource, such as a file or directory. Each ACE specifies the type of access allowed or denied and is associated with a particular security principal (user or group identity).

Access Control Entry (ACE) required for xattrs

- **Retrieve xattr:** The permissions required for a user to read the extended attributes of a file or directory. The "R" signifies that read permission is necessary.
- **Set xattrs:** The permissions needed to modify or set the extended attributes. "a," "w," and "T" represent different examples of permissions, such append, write, and a specific permission related to xattrs.
- **Files:** Users need append, write, and potentially a special permission related to xattrs to set extended attributes.
- **Directories:** A specific permission "T" is required to set extended attributes.

File type	Retrieve xattr	Set xattrs
File	R	a,w,T

File type	Retrieve xattr	Set xattrs
Directory	R	T

Integration with ABAC identity and access control software

To fully harness the capabilities of ABAC, ONTAP can integrate with an ABAC-oriented identity and access management software.

In an ABAC system, the Policy Enforcement Point (PEP) and Policy Decision Point (PDP) play crucial roles. The PEP is responsible for enforcing access control policies, while the PDP makes the decision on whether to grant or deny access based on the policies.

In a practical setting, an organization would employ a blend of NFS security labels and xattrs. These are used to represent a variety of metadata, including classification, security, application, and content, which are all instrumental in making ABAC decisions. xattrs, for instance, can be used to store the resource attributes that the PDP uses for its decision-making process. An attribute could be defined to represent the classification level of a file (for example, "Unclassified", "Confidential", "Secret", or "Top Secret"). The PDP could then utilize this attribute to enforce a policy that restricts users to access only files that have a classification level equal to or lower than their clearance level.



This content assumes that the customer's identity, authentication, and access services include at minimum a PEP and a PDP that act as intermediaries for access to the file system.

Example process flow for ABAC

1. User presents credentials (for example, PKI, Oauth, SAML) to system access to PEP and gets results from PDP.

The PEP's role is to intercept the user's access request and forward it to the PDP.

2. The PDP then evaluates this request against the established ABAC policies.

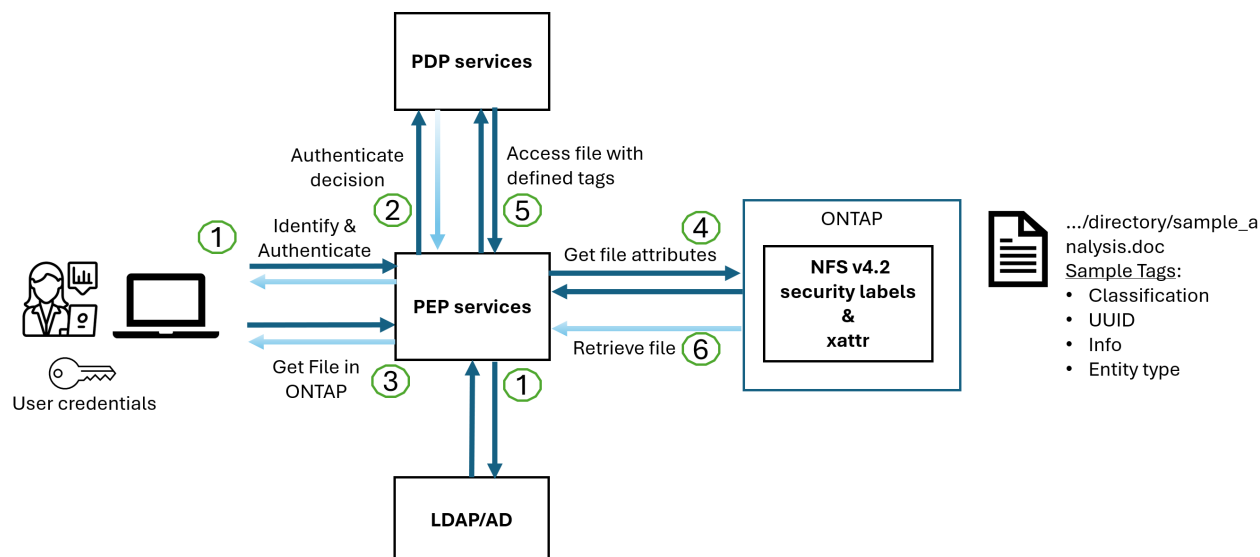
These policies consider various attributes related to the user, the resource in question, and the surrounding environment. Based on these policies, the PDP makes an access decision to either allow or deny and then communicates this decision back to the PEP.

PDP provides policy to PEP to enforce. The PEP then enforces this decision, either granting or denying the user's access request as per the PDP's decision.

3. After a successful request, the user requests a file stored in ONTAP (AFF, AFF-C, for example).
4. If the request is successful, PEP gets fine-grain access control tags from document.
5. PEP requests policy for user based on that user's certs.
6. PEP makes a decision based on policy and tags if the user has access to the file and lets the user retrieve the file.



The actual access might be done using tokens.



ONTAP cloning and SnapMirror

ONTAP's cloning and SnapMirror technologies are designed to provide efficient and reliable data replication and cloning capabilities, ensuring that all aspects of file data, including xattrs, are preserved and transferred along with the file. xattrs are critical as they store additional metadata associated with a file, such as security labels, access control information, and user-defined data, which are essential for maintaining the file's context and integrity.

When a volume is cloned using ONTAP's FlexClone technology, an exact writable replica of the volume is created. This cloning process is instantaneous and space-efficient, and it includes all file data and metadata, ensuring that xattrs are fully replicated. Similarly, SnapMirror ensures that data is mirrored to a secondary system with full fidelity. This includes xattrs, which are crucial for applications that rely on this metadata to function correctly.

By including xattrs in both cloning and replication operations, NetApp ONTAP ensures that the complete dataset, with all its characteristics, is available and consistent across primary and secondary storage systems. This comprehensive approach to data management is vital for organizations that require consistent data protection, quick recovery, and adherence to compliance and regulatory standards. It also simplifies the management of data across different environments, whether on-premises or in the cloud, providing users with the confidence that their data is complete and unaltered during these processes.



NFS v4.2 security labels have the caveats defined in [NFS v4.2 security labels](#).

Auditing changes to labels

Auditing changes to xattrs or NFS security labels is a critical aspect of file system management and security. Standard file system auditing tools enable the monitoring and logging of all changes to a file system, including modifications to xattrs and security labels.

In Linux environments, the `auditd` daemon is commonly used to establish auditing for file system events. It allows administrators to configure rules to watch for specific system calls related to xattr changes, such as `setxattr`, `lsetxattr`, and `fsetxattr` for setting attributes and `removexattr`, `lremovexattr`, and `fremovexattr` for removing attributes.

ONTAP FPolicy extends these capabilities by providing a robust framework for real-time monitoring and control of file operations. FPolicy can be configured to support various xattr events, offering granular control over file operations and the ability to enforce comprehensive data management policies.

For users utilizing xattrs, especially in NFS v3 and NFS v4 environments, only certain combinations of file operations and filters are supported for monitoring. The list of supported file operation and filter combinations for FPolicy monitoring of NFS v3 and NFS v4 file access events is detailed below:

Supported file operations	Supported filters
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

Example of an auditd log snippet for a 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"
```

Enabling [ONTAP FPolicy](#) for users working with xattrs provides a layer of visibility and control that is essential for maintaining the integrity and security of the file system. By leveraging FPolicy’s advanced monitoring capabilities, organizations can ensure that all changes to xattrs are tracked, audited, and aligned with their security and compliance standards. This proactive approach to file system management is why enabling ONTAP FPolicy is highly recommended for any organization looking to enhance its data governance and protection strategies.

Examples of controlling access to data

The following example entry for data stored in John R. Smith’s PKI cert shows how NetApp’s approach can be applied to a file and provide fine-grained access control.



These examples are for illustrative purposes, and it is the customer’s responsibility to determine the metadata associated with NFS v4.2 security labels and xattrs. Details on updating and label retention are omitted for simplicity.

Example PKI cert values

Key	Value
entitySecurityMark	t:s01 = UNCLASSIFIED
Info	<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>
specification	"DoD"
uuid	b4111349-7875-4115-ad30-0928565f2e15
adminOrganization	<pre> { "value": "DoD" } </pre>

Key	Value
briefings	<pre>[{ "value": "ABC1000" }, { "value": "DEF1001" }, { "value": "EFG2000" }]</pre>
citizenshipStatus	<pre>{ "value": "US" }</pre>
clearances	<pre>[{ "value": "TS" }, { "value": "S" }, { "value": "C" }, { "value": "U" }]</pre>
countryOfAffiliations	<pre>[{ "value": "USA" }]</pre>

Key	Value
digitalIdentifier	<pre>{ "classification": "UNCLASSIFIED", "value": "cn=smith john r jrsmith, ou=dod, o=u.s. government, c=us" }</pre>
dissemTos	<pre>{ "value": "DoD" }</pre>
dutyOrganization	<pre>{ "value": "DoD" }</pre>
entityType	<pre>{ "value": "GOV" }</pre>
fineAccessControls	<pre>[{ "value": "SI" }, { "value": "TK" }, { "value": "NSYS" }]</pre>

These PKI entitlements show John R. Smith's access details, including access by data type and attribution.

In scenarios where IC-TDF metadata is stored separately from the file, NetApp advocates for an additional layer of fine-grained access control. This involves storing access control information at both the directory level and in association with each file. As an example, consider the following tags linked to a file:

- NFS v4.2 security labels: Utilized for making security decisions

- xattrs: Provide supplementary information pertinent to the file and the organizational program requirements

The following key-value pairs are examples of metadata that could be stored as xattrs and offer detailed information about the file's creator and associated security classifications. This metadata can be leveraged by client applications to make informed access decisions and to organize files according to organizational standards and requirements.

Example of xattr key-value pairs

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

Key	Value
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>

Key	Value
user.geo_point	[-78.7941, 35.7956]

Related information

- [NFS in NetApp ONTAP: Best practice and implementation guide](#)
- [ONTAP command reference](#)
- Request for comments (RFC)
 - [RFC 7204: Requirements for Labeled NFS](#)
 - [RFC 2203: RPCSEC_GSS Protocol Specification](#)
 - [RFC 3530: Network File System \(NFS\) Version 4 Protocol](#)

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.