Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 31, 2018

K. Grizzle, Ed.
SailPoint
B. Yoder
Thycotic
J. Jones
Bomgar
P. Lieberman
Lieberman

September 27, 2017

SCIM Extension for Privileged Access Management
draft-grizzle-scim-pam-ext-00

Abstract

The System for Cross-domain Identity Management (SCIM) specification [RFC7643] provides schemas that represent common identity information about users and groups. Privileged Access Management (PAM) software typically makes use of common user and group models - as well as defining additional constructs - to provide fine-grained authorization and management for privileged access.

This document contains a SCIM 2.0 extension for Privileged Access Management, which includes extensions to the core User and Group objects, and new resource types and schemas for standard Privileged Access Management constructs. This extension is intended to provide greater interoperability between PAM software and clients, a common language for PAM concepts, and a baseline that can be further extended to support more complex PAM requirements.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 31, 2018.

Table of Contents

1. Overview

Most Privileged Access Management (PAM) software contains external APIs that can be used to manage users, groups, privileged access, and authorization to privileged data. However, these APIs are not consistent across different software (e.g. - some software uses REST and some uses SOAP), and each API exposes different functionality. This makes it difficult for a client to externally manage multiple PAM providers.

The System for Cross-domain Identity Management (SCIM) specification provides schemas that represent common identity information about users and groups. Privileged Access Management (PAM) software typically makes use of common user and group models - as well as defining additional constructs - to provide fine-grained authorization and management for privileged access.

This document contains a SCIM 2.0 extension for Privileged Access Management, which includes extensions to the core User and Group objects, and new resource types and schemas for standard Privileged Access Management constructs. This extension is intended to provide greater interoperability between PAM software and clients, a common language for PAM concepts, and a baseline that can be further extended to support more complex PAM requirements.

Some providers may not support all of the endpoints or data that is described in this extension. When this is encountered, the PAM provider can safely treat endpoints or data as optional.

1.1. Definitions

User: A user account that can be used to access the PAM system to manage or access privileged data. This user can either exist only in the PAM system or can be an external user that is defined in another system (e.g. - Active Directory or LDAP).

Group: A group of users or other groups that can be used to govern access within the PAM system. This group can either exist only in the PAM system or can be an external group that is defined in another system (e.g. - Active Directory or LDAP).

Container: A Container is a logical grouping of privileged data (credentials, etc...) that can be used for organizational or operational purposes. Access control lists (ACLs) can be applied to a container to control which users and groups have permissions to the privileged data in the container.

Privileged Data: Privileged data is secret information that is protected by the PAM system (e.g. - credentials for a privileged account, an SSH key, etc...). Privileged data MAY be stored inside of a Container, but does not have to be. Access control lists (ACLs) can be applied to privileged data to control which users and groups have permissions to the privileged data. More often, the ACL information is inherited from the container.

Access Control List (ACL): An access control list can be associated with a Container or Privileged Data. This contains information about which users and groups have access to the Container or Privileged Data, and what rights they have.

External Store: An external store is a system that contains users and groups (e.g. - Active Directory or LDAP) that can be used by a PAM system. This allows using existing infrastructure and group definitions to provide authorization, authentication, and information within a PAM system.

1.2. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

2. Core Schema Extensions

In a PAM system, users and groups can either be locally or externally defined. When local, the user or group exists only on the PAM system. When external, the user or group is defined in an External Store, and is somehow synchronized into the PAM system. In this case, the PAM system keeps a record of the external user or group, along with a reference that can be used to correlate the record back to the external store. To support this, an optional schema extension "urn:ietf:params:scim:schemas:pam:1.0:LinkedObject" MAY be added to the User and Group resource types.

2.1. Linked Object

The "urn:ietf:params:scim:schemas:pam:1.0:LinkedObject" schema contains the following attributes.

source The name of the External Source from which the User or Group came. If this is a local User or Group, this is null. Required if nativeIdentifier is non-null.

nativeIdentifier The unique identifier of the User or Group on the External Source (e.g. - an LDAP distinguished name). If this is a local User or Group, this is null. Required if source is non-null.

2.1.1. Example

The following is a non-normative example of a User with the LinkedObject extension.

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:pam:1.0:LinkedObject"
  ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "userName": "bjensen",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "givenName": "Barbara",
    "familyName": "Jensen",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "emails": [
    {
      "value": "bjensen@example.com",
      "type": "work",
      "primary": true
    },
    {
      "value": "babs@jensen.org",
      "type": "home"
    }
  ],
  "active": true,
  "groups": [
    {
      "value": "e9e30dba-f08f-4109-8486-d5c6a331660a",
      "$ref": "https://example.com/v2/Groups/e9e30dba-f08f-4109-8486-
d5c6a331660a",
      "display": "Tour Guides",
```

```
      "type": "direct"
```

```
    },
    {
      "value": "fc348aa8-3835-40eb-a20b-c726e15c55b5",
      "$ref": "https://example.com/v2/Groups/fc348aa8-3835-40eb-a20b-
c726e15c55b5",
      "display": "Employees",
      "type": "indirect"
    }
  ],
  "urn:ietf:params:scim:schemas:pam:1.0:LinkedObject": {
    "source": "Corporate Active Directory",
    "nativeIdentifier": "cn=Barbara Jensen,ou=Users,dc=example,dc=com"
  },
  "meta": {
    "resourceType": "User",
    "created": "2010-01-23 04:56:22 UTC",
    "lastModified": "2011-05-13 04:42:34 UTC",
    "location": "https://example.com/v2/Users/
2819c223-7f76-453a-919d-413861904646"
  }
}
```


2.1.2. Considerations for External Groups

Members of external groups are stored and managed on the External Store, and not in the PAM system. As a result, the User and Group representations returned by the PAM system MAY return empty values for the "groups" and "members" attributes, respectively. Additionally, the PAM system MAY choose to return an error response with the 400 status code and "invalidSyntax" error type for requests that attempt to modify or create a group with an invalid configuration. Examples include, but are not limited to:

- An external group with any members.
- An external group with local Users or Groups as members.
- A local group with external Users or Groups as members.

3. Additional ResourceTypes and Schemas

PAM systems define additional constructs to provide enhanced authorization, authentication, and management for privileged data. To support this, the SCIM PAM extension defines additional ResourceTypes and Schemas that MAY be implemented by the service provider. If implemented, these ResourceTypes SHOULD support all SCIM operations [RFC7644]. All attributes defined in the schemas are optional unless explicitly marked as REQUIRED.

3.1. Container

A Container is a logical grouping of privileged data that can be used for organizational or operational purposes.

3.1.1. Resource Type

The Container ResourceType supports reading and managing containers, and has the following properties.

Name: Container

Endpoint: /Containers

Schema: urn:ietf:params:scim:schemas:pam:1.0:Container

3.1.1.1. Filtering

Clients may have a reference to the Container name but not the ID. For this reason, it is RECOMMENDED that service providers implement filtering that allows equality matching on the "name" attribute. Example (note that escaping has been removed for readability):

GET /scim/v2/Containers?filter=name eq 'Admin Accounts'

3.1.2. Schema

The "urn:ietf:params:scim:schemas:pam:1.0:Container" defines all common attributes for a Container.

id The unique identifier of the Container. REQUIRED

name The name of the Container. REQUIRED

displayName The display name of the Container. If null, the name should be used as the display name.

description The description of the Container.

type The type of container. There are no canonical values defined for type, but service providers MAY choose to define the valid types. Optional if the PAM system does not support multiple types of Containers.

parent A complex attribute that defines the parent Container of this Container if the service provider supports hierarchies of containers. The following sub-attributes are defined.

value The ID of the Container that is the parent of this Container in the hierarchy.

$ref A URI reference to the Container that is the parent of this Container in the hierarchy.

display The display name of the Container that is the parent of this Container in the hierarchy.

owner A complex attribute that defines the User that is the owner of this Container. The following sub-attributes are defined.

value The ID of the User that owns this Container.

$ref A URI reference to the User that owns this Container.

display The display name of the user that owns this Container.

privilegedData A multi-valued complex attribute that contains the PrivilegedData that resides in this Container. Service providers MAY choose to make this attribute have a "returned" value of "request" if the list of privileged data could be very large. Using this option will prevent this attribute from being returned upon retrieval unless explicitly requested using the "attributes" query parameter. The following sub-attributes are defined.

value The ID of the PrivilegedData.

$ref A URI reference to the PrivilegedData.

display The displayable value of the PrivilegedData.

type The type of the PrivilegedData.

3.1.3. Example

The following is a non-normative example of a Container.

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:pam:1.0:Container"
  ],
  "id": "ab8e901-883f-4109-8486-bab810943d93e",
  "name": "prodDBAAccounts",
  "displayName": "Production DBA Accounts",
  "description": "This contains all DBA accounts for the production
environment.",
  "type": "safe",
  "parent": {
    "value": "78234914-7fb3-828e-7281-87234abe8300",
    "$ref": "https://example.com/v2/Containers/
78234914-7fb3-828e-7281-87234abe8300",
    "display": "Root Container"
  },
  "owner": {
    "value": "2819c223-7f76-453a-919d-413861904646",
    "$ref": "https://example.com/v2/Users/
2819c223-7f76-453a-919d-413861904646",
    "display": "Babs Jensen"
  },
  "privilegedData": [
    {
      "value": "d973b5-8834f-1784-8734-caf833e9b3efa",
      "$ref": "https://example.com/v2/Containers/d973b5-8834f-1784-8734-
caf833e9b3efa",
      "display": "root @ Oracle Financials Warehouse",
      "type": "credential"
    },
    {
      "value": "d249e9-92759-7883-88723-fa390734beba",
      "$ref": "https://example.com/v2/Containers/d249e9-92759-7883-88723-
fa390734beba",
      "display": "root @ Enterprise Purchase Ordering",
      "type": "credential"
    }
  ],
  "meta": {
    "resourceType": "Container",
    "created": "2010-01-23T04:56:22.000Z",
    "lastModified": "2011-05-13T04:42:34.000Z",
    "location": "https://example.com/v2/Container/ab8e901-883f-4109-8486-
bab810943d93e"
  }
}
```

3.2. PrivilegedData

Privileged data is secret information that is protected by the PAM system (e.g. - credentials for a privileged account, an SSH key, etc...). Privileged data MAY be stored inside of a Container, but does not have to be.

3.2.1. Resource Type

The PrivilegedData ResourceType supports reading and managing privileged data, and has the following properties.

Name: PrivilegedData

Endpoint: /PrivilegedData

Schema: urn:ietf:params:scim:schemas:pam:1.0:PrivilegedData

3.2.2. Schema

The "urn:ietf:params:scim:schemas:pam:1.0:PrivilegedData" defines all common attributes for a PrivilegedData.

id The unique identifier of the PrivilegedData. REQUIRED

name A descriptive name for this piece of PrivilegedData. For example, root@mylinuxhost. REQUIRED

description A description for this piece of PrivilegedData.

type The type of PrivilegedData. The value will be dependent on what is supported by the PAM system. Examples include 'credential', 'ssh key', 'file', etc...

3.2.3. Example

The following is a non-normative example of a PrivilegedData.

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:pam:1.0:PrivilegedData"
  ],
  "id": "d973b5-8834f-1784-8734-caf833e9b3efa",
  "name": "root @ Oracle Financials Warehouse",
  "description": "Full access to the Oracle Financials Warehouse database.",
  "type": "credential",
  "meta": {
    "resourceType": "PrivilegedData",
    "created": "2010-01-23T04:56:22.000Z",
    "lastModified": "2011-05-13T04:42:34.000Z",
    "location": "https://example.com/v2/PrivilegedData/d973b5-8834f-1784-8734-
caf833e9b3efa"
  }
}
```

3.3. ContainerPermission

A ContainerPermission contains authorization information that describes which rights a User or Group has on a Container. This is a piece of an Access Control List that contains all information about a specific User or Group in relation to a specific Container. Typically, permissions that are granted on a Container apply to all privileged data that resides in the container.

3.3.1. Resource Type

The ContainerPermission ResourceType supports reading and managing permissions that a User or Group have on a Container, and has the following properties.

Name: ContainerPermission

Endpoint: /ContainerPermissions

Schema: urn:ietf:params:scim:schemas:pam:1.0:ContainerPermission

3.3.1.1. Filtering

It is expected that clients will need to find the permissions on a specific Container that are granted to a specific User or Group. For this reason, it is RECOMMENDED that service providers implement filtering that allows equality matching on the "container.value", "user.value", and "group.value" attributes. Example (note that escaping has been removed and newlines added for readability):

```
GET /scim/v2/ContainerPermissions?
        filter=container.value eq '8729e778-9af6-874c-778a3-783956810384' and
               user.value eq '2819c223-7f76-453a-919d-413861904646'
```

3.3.2. Schema

The "urn:ietf:params:scim:schemas:pam:1.0:ContainerPermission" defines all common attributes for a ContainerPermission.

id The unique identifier of the ContainerPermission. REQUIRED

container A complex attribute that references the Container that these permissions apply to. The following sub-attributes are defined. REQUIRED

value The ID of the Container that these permissions apply to.

$ref A URI reference to the Container that these permissions apply to.

name The name of the Container that these permissions apply to.

display The display name of the Container that these permissions apply to.

user A complex attribute that references the User that these permissions apply to. Either this attribute or "group" is required. The following sub-attributes are defined.

value The ID of the User that these permissions apply to.

$ref A URI reference to the User that these permissions apply to.

display The display name of the User that these permissions apply to.

group A complex attribute that references the Group that these permissions apply to. Either this attribute or "user" is required. The following sub-attributes are defined.

value The ID of the Group that these permissions apply to.

$ref A URI reference to the Group that these permissions apply to.

display The display name of the Group that these permissions apply to.

rights An array of strings that are the names of the rights that the User or Group has on this Container. There are no canonical values defined for rights, and these will vary between service providers.

3.3.3. Example

The following is a non-normative example of a ContainerPermission.

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:pam:1.0:ContainerPermission"
  ],
  "id": "c387432-78823-87234-7832-93c9ae93745e",
  "container": {
    "value": "ab8e901-883f-4109-8486-bab810943d93e",
    "$ref": "https://example.com/v2/Containers/ab8e901-883f-4109-8486-
bab810943d93e",
    "display": "Production DBA Accounts",
    "name": "prodDBAAccounts"
  },
  "user": {
    "value": "2819c223-7f76-453a-919d-413861904646",
    "$ref": "https://example.com/v2/Users/
2819c223-7f76-453a-919d-413861904646",
    "display": "Babs Jensen"
  },
  "rights": [
    "Connect",
    "List Accounts",
    "View Password"
  ],
  "meta": {
    "resourceType": "ContainerPermission",
    "created": "2010-01-23T04:56:22.000Z",
    "lastModified": "2011-05-13T04:42:34.000Z",
    "location": "https://example.com/v2/ContainerPermissions/
c387432-78823-87234-7832-93c9ae93745e"
  }
}
```


3.4. PrivilegedDataPermission

A PrivilegedDataPermission contains authorization information that describes which rights a User or Group has on a PrivilegedData. This is a piece of an Access Control List that contains all information about a specific User or Group in relation to a specific piece of privileged data. This resource type and schema are OPTIONAL if the service provider does not support permissions on privileged data.

3.4.1. Resource Type

The PrivilegedDataPermission ResourceType supports reading and managing permissions that a User or Group have on a PrivilegedData, and has the following properties.

Name: PrivilegedDataPermission

Endpoint: /PrivilegedDataPermissions

Schema: urn:ietf:params:scim:schemas:pam:1.0:PrivilegedDataPermissio
n

3.4.1.1. Filtering

It is expected that clients will need to find the permissions on a specific PrivilegedData that are granted to a specific User or Group. For this reason, it is RECOMMENDED that service providers implement filtering that allows equality matching on the "privilegedData.value", "user.value", and "group.value" attributes. Example (note that escaping has been removed and newlines added for readability):

```
GET /scim/v2/PrivilegedDataPermissions?
        filter=privilegedData.value eq '2746c134-59e8-848a-874d3-782303476812'
and
               user.value eq '2819c223-7f76-453a-919d-413861904646'
```

3.4.2. Schema

The "urn:ietf:params:scim:schemas:pam:1.0:PrivilegedDataPermission" defines all common attributes for a PrivilegedDataPermission.

id The unique identifier of the PrivilegedDataPermission. REQUIRED

privilegedData A complex attribute that references the PrivilegedData that these permissions apply to. The following sub-attributes are defined. REQUIRED

value The ID of the PrivilegedData that these permissions apply to.

$ref A URI reference to the PrivilegedData that these permissions apply to.

display The display name of the PrivilegedData that these permissions apply to.

user A complex attribute that references the User that these permissions apply to. Either this attribute or "group" is required. The following sub-attributes are defined.

value The ID of the User that these permissions apply to.

$ref A URI reference to the User that these permissions apply to.

display The display name of the User that these permissions apply to.

group A complex attribute that references the Group that these permissions apply to. Either this attribute or "user" is required. The following sub-attributes are defined.

value The ID of the Group that these permissions apply to.

$ref A URI reference to the Group that these permissions apply to.

display The display name of the Group that these permissions apply to.

rights An array of strings that are the names of the rights that the User or Group has on this PrivilegedData. There are no canonical values defined for rights, and these will vary between service providers.

3.4.3. Example

The following is a non-normative example of a PrivilegedDataPermission.

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:pam:1.0:PrivilegedDataPermission"
  ],
  "id": "f823414-872344-77381-ab93489d83ea87",
  "privilegedData": {
    "value": "d973b5-8834f-1784-8734-caf833e9b3efa",
    "$ref": "https://example.com/v2/PrivilegedData/d973b5-8834f-1784-8734-
caf833e9b3efa",
    "display": "root @ Oracle Financials Warehouse"
  },
  "group": {
    "value": "e9e30dba-f08f-4109-8486-d5c6a331660a",
    "$ref": "https://example.com/v2/Groups/e9e30dba-f08f-4109-8486-
d5c6a331660a",
    "display": "Tour Guides"
  },
  "rights": [
    "Connect",
    "View Password"
  ],
  "meta": {
    "resourceType": "PrivilegedDataPermission",
    "created": "2010-01-23T04:56:22.000Z",
    "lastModified": "2011-05-13T04:42:34.000Z",
    "location": "https://example.com/v2/PrivilegedDataPermissions/
f823414-872344-77381-ab93489d83ea87"
  }
}
```


4. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7643] Hunt, P., Ed., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Core Schema", RFC 7643, DOI 10.17487/RFC7643, September 2015, <https://www.rfc-editor.org/info/rfc7643>.

[RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, <https://www.rfc-editor.org/info/rfc7644>.

Authors' Addresses

Kelly Grizzle (editor)
SailPoint

Email: kelly.grizzle@sailpoint.com

Benjamin Yoder
Thycotic

Email: ben.yoder@thycotic.com

Jason Jones
Bomgar

Email: jjones@bomgar.com

Philip Lieberman
Lieberman Software

Email: phil@liebsoft.com