Predict | Protect | Prevent

ARCON|PAM

ARCON PAM SCIM API Specifications

www.arconnet.com | Copyright © 2022

arcon

Table of Contents

Disclaimer

The handbook is published to guide users and customers with the details of the ARCON PAM API Details.. If any of the statements in this document are at variance or inconsistent it shall be brought to the notice of ARCON through the support team. Wherever appropriate, references have been made to facilitate a better understanding of the PAM solution. The ARCON PAM team has made every effort to ensure that the information contained in it was correct at the time of publishing.

Copyright Notice

ARCON PAM retains the right to make changes to this document at any time without notice. ARCON PAM makes no warranty for the use of this document and assumes no responsibility for any errors that can appear in the document nor does it make a commitment to update the information contained herein.

Trademarks

Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

Sales Contact

You can directly contact us with sales-related topics at the email address <sales@arconnet.com>, or leave us your contact information and we will call you back.

1 Why SCIM?

SCIM is an open standard that simplifies the process of centrally managing user identities. Compared to integrating directly with LDAP/AD, SQL, or other protocols, SCIM is a more modern and lightweight approach. From a developer standpoint, SCIM is much easier to work with.

One of the primary advantages of SCIM provisioning as an open standards-based approach is standardization which helps in easy integration with Identity and Access Management tools. SCIM, with its built-in accessibility and interoperability, tackles identity maintenance difficulties such as record keeping, manual onboarding, offboarding, or inadequate partner-to-partner communication, and enables safe identity data transmission.

SCIM also helps streamline disparate identity stores, such as during a company acquisition. The same is true for employee provisioning or de-provisioning — SCIM acts as a single source of truth for user identities, removing the hurdles of adding or removing users to many applications within an organization. Taking it one step further, SCIM can enable self-service and self-registration to alleviate the IT department's burden.

2 PAM Overview

Making the API's SCIM compliant simplifies the user experience by automating the user identity lifecycle management process. With SCIM, administrators can create process automation for provisioning and de-provisioning entities in the PAM solution.

The table below depicts the entities and their use in PAM solution

Entity Name	Entity Description	Entity Usage
User	This entity represents the PAM account (user name and password)	PAM user is used by the operational team to connect to the PAM portal and access PAM Services. In PAM there are 2 types of users: • Client – this type of user represents the end-users (members of the operational team) that will use the PAM portal to access target systems with an elevated privilege • Admin – this type of user represents the PAM administrators IAM-PAM integration will address only client users and it will create the user as a domain user and add it to according to user groups. The configuration of the access within the PAM solution is not in scope for IAM-PAM integration, therefore, from an application access perspective all users created by IAM will have the default configuration.
User Group	This entity represents a set of users classified from the functional perspective	User Groups are used to classify the operational team members into Windows Admins, Linux Admins, etc. From PAM applications' perspective, User Groups are mapped to Server Groups in order to allow the users to access services. The mapping between the users and User Groups will be done automatically during the IAM provisioning process. The mappings between User and Server Groups will be done manually in PAM, as part of the initial configuration process and is not in scope for IAM-PAM integration.
LOB	This entity represents a functional grouping of PAM users from the organizational perspective.	LOBs are used to setup: • The scope of the Reports • The target of the Password policies • The VPN servers to be used • The command profilers if they are defined at the User group level The LOB lifecycle management will be manually in PAM and is out of scope for IAM-PAM integration.

3 ARCON PAM SCIM API Details

3.1 ARCON PAM Authentication Token Creation

This API will authenticate the user, after successful authentication, it will generate a token and return it. This token will be used for all further API calls.

1. URL - https://<HostName:Port>/**api/Token/arcontoken**
2. Method - POST
3. URL Params - None
4. Content-Type - application/x-www-form-urlencoded
5. Data Params (Body) –

   a. *Request*
   ```
   {
     "UserName":<Username in Base64 encoded Format>,
     "Password":<Password in Base64 encoded Format>,
     "Roles":[{
       "Role": <Access Type Roles>
     }]
   }
   ```
6. Response

   a. access_token: <token>
   b. token_type: "bearer"
   c. expires_in: <Time for expiry, currently 1 day>
   d. success: "Successful"
   e. message: "OK"

⚠

1. Use Base64 encoded for username and password while passing it to token generation mechanism of API arcontoken .
2. For Responses in XML Format, Use ‘Accept: application/xml’ specified in headers.
3. Similarly for JSON Format, Use ‘Accept: application/json’ specified in Headers, by default response data will be in JSON Format.
4. For every single request you have to send ("Authorization", tokenType + " " + arconToken) in the header section.
5. Token will be valid for 24 hours, you have to regenerate token after every 24 hour. expires_in: <Time for expiry, currently 1 day>

3.2 ARCON PAM LOB List

This API is used to fetch LOB list from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Lobs**
2. Method - GET
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>

6. Input Data Parameters - None
7. Response
    a. Status: 200
    b. Description: Success
    c. Result: JSON array of LOB List

<u>**SAMPLE JSON**</u>

When data is returned

```
{
   "Program": "Arcon Micro Service API",
   "Version": "1.0.0.0",
   "StatusCode": 200,
   "Message": "Success",
   "Ticks": "4847.2425",
   "ResultCount": "26",
   "Result": [
   {
      "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:Lob"
      ],
      "meta": {
      "ResourceType": "Lob",
      "Created": "2018-03-13T15:35:57.183",
      "LastModified": "2022-02-25T14:47:53.73",
     "Version": null,
     "Location": null
      },
     "id": 1,
     "lobName": "DEFAULT LOB 2"
   }
  ]
}
```

When no data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "4847.2425",
"ResultCount": "0",
"Result": []
}
```

3.3 ARCON PAM Get all User Details

This API is used to fetch all User Details from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Users**
2. Method - GET
3. URL Parameters - None

4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters
    a. Filter - Using Multiple Lob Id (optional)

    ```
    {
    "LobId":[1,2]
    }
    ```

    b. Response
        i. Status: 200
        ii. Description: Success
        iii. Result: User Details List

SAMPLE JSON

When data is returned

```
{
   "Program": "Arcon Micro Service API",
   "Version": "1.0.0.0",
   "StatusCode": 200,
   "Message": "Success",
   "Ticks": "8316.9",
   "ResultCount": "1",
   "Result": [
   {
      "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:User"
       ],
      "meta": {
      "ResourceType": "User",
      "Created": "2018-03-13T15:35:57.183",
      "LastModified": "2018-03-13T15:35:57.183",
      "Version": null,
      "Location": null
  },
  "id": 35,
  "userName": "ARCOSADMIN",
  "displayName": "ARCOSADMIN",
  "ValidTillDate": "1/1/2058 11:59:59 PM",
  "Enabled": 0,
  "name": {
  "formatted": "ARCOSADMIN",
 "familyName": null,
 "givenName": null,
 "middleName": null
 },
 "emails": {
 "type": null,
 "value": "tejal.solanki@arconnet.com"
 },
 "group": {
 "id": 0,
 "groupName": null
```

```
 },
 "domainName": "ARCOSAUTH",
 "lob": [
 {
 "lobId": 1,
 "lobName": ""
 },
 {
 "lobId": 2,
 "lobName": ""
 },
 {
 "lobId": 3,
 "lobName": ""
 },
 {
 "lobId": 4,
 "lobName": ""
 }
 ],
"phoneNumbers": {
"value": "",
"type": null
}
},
]
}
```

When no data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "4847.2425",
"ResultCount": "0",
"Result": []
}
```

3.4 ARCON PAM Get User Details

This API is used to fetch all User Details by ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Users/23426**
2. Method - GET
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

    a. ID - User ID
    b. Response

        i. Status: 200

ii. Description : Success
iii. Result: User Details List

SAMPLE JSON

When data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "356.6325",
"ResultCount": "1",
"Result": [
{
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:User"
],
"meta": {
"ResourceType": "User",
"Created": "2018-03-13T15:35:57.183",
"LastModified": "2018-03-13T15:35:57.183",
"Version": null,
"Location": null
},
"id": 23426,
"userName": "TESTFOR7714",
"displayName": "TEST2",
"ValidTillDate": "1/1/2058 11:59:59 PM",
"Enabled": 0,
"name": {
"formatted": "TEST2",
"familyName": null,
"givenName": null,
"middleName": null
},
"emails": {
"type": null,
"value": "aditya.virdhe@arconnet.com"
},
"group": {
"id": 0,
"groupName": null
},
"domainName": "ARCOSAUTH",
"lob": [
{
"lobId": 16,
"lobName": ""
}
],
"phoneNumbers": {
"value": "1234567899",
"type": null
},
}
]
}
```

When no data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "4847.2425",
"ResultCount": "0",
"Result": []
}
```

3.5 ARCON PAM Set User Details

This API is used to set User Details from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Users/**
2. Method - POST
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

    a. Request

```
{
"userName": string,
"Active": bool,
"displayName": string,
"lob":[{"lobId":0}],
"group":{"id":0},
"domainName":string,
"userTypeId":int,
"password":string,
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:User"
],
"externalId": "",
"name": {
"formatted": string,
"familyName": string,
"givenName": string
},
"emails":
{
"type": string,
"value": string
}
}
```

b. Response

   i. Status: 200
   ii. Description : Success
   iii. Result: Set User details

SAMPLE JSON

When data Added

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "14.887",
"ResultCount": "0",
"Result": []
}
```

3.6 ARCON PAM Update User Details

This API is used to update User Details by ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Users/**
2. Method - PUT
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

   a. Request

```
{
"userName": string,
"Active": bool,
"displayName": string,
"id":0,
"lob":[{"lobId":0}],
"group":{"id":0},
"domainName":string,
"userTypeId":int,
"password":string,
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:User"
],
"externalId": "",
"name": {
"formatted": string,
"familyName": string,
"givenName": string
},
"emails":
{
"type": string,
"value": string
}
}
```

b. Response

   i. Status: 200
   ii. Description : Success
   iii. Result: Update User Details by id

SAMPLE JSON

When data Added

{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "14.887",
"ResultCount": "0",
"Result": []
}

3.7 ARCON PAM Delete User Details

This API is used to Delete User Details by ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/v1/Users/23426
2. Method - DELETE
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

   a. id - User ID(required)
   b. Response

      i. Status: 200
      ii. Description : Success
      iii. Result: Delete user by id

SAMPLE JSON

When data Deleted

{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "14.887",
"ResultCount": "0",
"Result": []
}

3.8 ARCON PAM Get all User Groups

This API is used to fetch all User Groups or filter using Lob ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Groups**
2. Method - GET
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters
    a. Request
    ```
    {
    "LobId":[1,2]
    }
    ```
    b. Response
        i. Status: 200
        ii. Description : Success
        iii. Result: User Groups Details List

SAMPLE JSON

When data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "677.1838",
"ResultCount": "170",
"Result": [
{
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group"
],
"meta": {
"ResourceType": "Group",
"Created": "2017-10-04T18:39:41.797",
"LastModified": "2021-03-15T01:55:55.37",
"Version": null,
"Location": null
},
"id": 6,
"groupName": "ADMINS",
"members": [
{
"userId": 35,
"userName": "ARCOSADMIN",
"userDisplayName": "ARCOSADMIN",
"UserGroupId": 6
}
]
},
{
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group"
],
```

```
"meta": {
"ResourceType": "Group",
"Created": "2018-03-13T15:41:06.953",
"LastModified": "0001-01-01T00:00:00",
"Version": null,
"Location": null
},
"id": 7,
"groupName": "WINDOWS ADMINS",
"members": [
{
"userId": 35,
"userName": "ARCOSADMIN",
"userDisplayName": "ARCOSADMIN",
"UserGroupId": 7
}
]
}
]
}
```

When no data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "4847.2425",
"ResultCount": "0",
"Result": []
}
```

3.9 ARCON PAM Get all User Groups

This API is used to fetch all User Groups by ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Groups/6**
2. Method - GET
3. URL Parameters - None
4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

    a. Response

        i. Status: 200
        ii. Description : Success
        iii. Result: User Groups Details List

SAMPLE JSON

When data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "43.5453",
"ResultCount": "2",
"Result": {
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group"
],
"meta": {
"ResourceType": "Group",
"Created": "2017-10-04T18:39:41.797",
"LastModified": "2021-03-15T01:55:55.37",
"Version": null,
"Location": null
},
"id": 6,
"groupName": "ADMINS",
"members": [
{
"userId": 35,
"userName": "ARCOSADMIN",
"userDisplayName": "ARCOSADMIN",
"UserGroupId": 6
},
{
"userId": 22952,
"userName": "VASANT",
"userDisplayName": "VASANT VERM",
"UserGroupId": 6
}
]
}
}
```

When no data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "4847.2425",
"ResultCount": "0",
"Result": []
}
```

3.10 ARCON PAM Update User Groups

This API is used to fetch all User Groups by ID from ARCON PAM

1. URL - http(s)://<HostName:Port>/**v1/Groups**
2. Method - PATCH
3. URL Parameters - None

4. Content-Type - application/json Or application/xml
5. Authorization - <token_type> <access_token>
6. Input Data Parameters

    a. Response

        i. Status: 200
        ii. Description : Success
        iii. Result: Update User Groups Details List

<u>SAMPLE JSON</u>

When data is returned

```
{
"Program": "Arcon Micro Service API",
"Version": "1.0.0.0",
"StatusCode": 200,
"Message": "Success",
"Ticks": "10577.1298",
"ResultCount": "5",
"Result": {
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group"
],
"meta": {
"ResourceType": "Group",
"Created": "2017-10-04T18:39:41.797",
"LastModified": "2021-03-15T01:55:55.37",
"Version": null,
"Location": null
},
"id": 6,
"groupName": "ADMINS",
"members": []
}
}
```

Privileged Access Management Suite

arcon

Predict | Protect | Prevent