Contents

About this Book 5

1 Overview 7

2 Authenticating to Web-Based Event with the U2F Method 9

Register an Endpoint . . . 10
API Call . . . 10
Sample Request . . . 10
Sample Response . . . 10
Create a Session for the Endpoint . . . 10
API Call . . . 10
Sample Request . . . 11
Sample Response . . . 11
Initiate Logon Process with the U2F Method . . . 11
API Call . . . 11
Sample Request . . . 11
Sample Response . . . 12
Gather the Sign Request . . . 13
API Call . . . 13
Sample Request . . . 14
Sample Response . . . 14
Forward Sign Requests to JSAPI . . . 15
API Call . . . 15
Sample Request . . . 16
Sample Response . . . 16
Forward the Sign Response to Server . . . 16
API Call . . . 16
Sample Request . . . 17
Sample Response . . . 17

3 Authenticating with Two Factors - LDAP Password and Fingerprint Methods 21

Prerequisites . . . 21
Create an Endpoint . . . 22
API Call . . . 22
Sample Request . . . 22
Sample Response . . . 22
Create a Session for the Endpoint . . . 22
API Call . . . 22
Sample Request . . . 23
Sample Response . . . 23
Initiate Logon Process . . . 23
API Call . . . 23
Sample Request . . . 23
Sample Response . . . 24
Verify First Factor Authenticator - LDAP Password . . . 25

API Call 25
Sample Request 25
Sample Response 25
Initiate the Fingerprint Authentication 26
API Call 26
Sample Request 26
Sample Response 27
Capture the Fingerprint 27
API Call 28
Sample Response 28
Forward the Fingerprint Image to Server 28
API Call 28
Sample Request 28
Sample Response 29

Advanced Authentication 6.4

API Use-Case Guide

February 2024

Legal Notice

© Copyright 2021 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

About this Book

This Integration Guide is intended for administrators and describes the procedure and technical information about how to integrate a product or application with Advanced Authentication to leverage multi-factor authentication capability.

Intended Audience

This book provides information for administrators trying to integrate their product with Advanced Authentication to secure the product with multi-factor authentication.

1 Overview

This guide describes how to interact with the Advanced Authentication Server using API calls. It provides the expected sequence of calls to the API required to successfully authenticate a user using the enrolled methods in the chain.

2 Authenticating to Web-Based Event with the U2F Method

Let us assume administrator of Reltic Data wants to authenticate users with the enrolled U2F method to the Search Card Portal.

This section explains the prerequisites, sequence of API calls, and sample request details to achieve this.

Prerequisites

Ensure to gather the following details before you execute any API call:

- **{{hostname}}** indicates Advanced Authentication server IP address or host name.
- **{{endpoint_id}}** indicates the unique device ID where the user authenticates.
- **endpoint_secret_hash** indicates the value used to hide the plain text hash an endpoint secret and id.
- **salt** indicates the technique required to hash the password.
- **event** indicates the event name that is configured on the Administration Portal.
- **endpoint_session_id** unique session ID that is generated when the endpoint session is created.
- **logon_process_id** indicates the process ID to the given event. This is generated when you attempt to login to the event.
- Ensure to create a chain with U2F method and associate that chain to the Search Card Portal.
- Ensure Advanced Authentication Device Service is installed on the system from where you initiate API calls to perform U2F authentication.

  For more information, see Installing and Upgrading Device Service.

The administrator must execute the following APIs to authenticate users with the U2F method:

1. Register an Endpoint
2. Create a Session for the Endpoint
3. Initiate Logon Process with the U2F Method
4. Gather the Sign Request
5. Forward Sign Requests to JSAPI
6. Forward the Sign Response to Server

NOTE: Any text enclosed in double flower brackets {{ }} indicates defining or referencing variables. When you include variables in the request body, Postman or other API tool replaces variables with the respective values during the runtime.

Register an Endpoint

Endpoint is a device where the user authenticates. After registering an endpoint, you will get endpoint id and secret.

API Call

```
POST https://{{hostname}}/api/v1/endpoints
```

For more information, see Register New Endpoint (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#endpoints-register-new-endpoint-post).

Sample Request

```
{
   "name":"aa.endpoint",
   "typ":3,
   "desc":"Endpoint",
   "is_trusted":true,
   "tenant_name":"TOP",
   "auth_data":{
      "method_id":"PASSWORD:1",
      "user_name":"LOCAL\\admin",
      "password":"novell@123"
   }
}
```

Sample Response

```
{ "id": "1889d45e7d5411ee9ef60242ac110003", "secret":
"H0ca3uY8x9wtZgauLGhX7mLLeptQo2MG" }
```

Create a Session for the Endpoint

To create an endpoint session, `endpoint_secret` and `endpoint_id` from a trusted source are required. Server validates and returns `endpoint_session_id` if provided values are valid.

For more information, see Endpoint Session (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#endpoint-sessions-create-endpoint-session-post).

API Call

```
POST https://{{hostname}}/api/v1/endpoints/{{endpoint_id}}/sessions
```

Sample Request

```
{
    "salt": "i_am_salt",
    "endpoint_secret_hash":
"e7e2dfe2209a17159689d4a731ba4b6f381843f03705c4adc734ea238fad8f2f"

}
```

Sample Response

```
{
"endpoint_session_id": "YV7ebuNLbKj9VuR4wxd7EGqaWG13WjuH"
}
```

Initiate Logon Process with the U2F Method

Initiates the logon process to the given event with the U2F method. The `endpoint_session_id` is required.

API Call

`POST https://{{hostname}}/api/v1/logon`

For more information, see Initiate Logon Process (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#logon-create-logon-process-post).

Sample Request

```
{
   "method_id":"U2F:1",
   "user_name":"denis",
   "event":"Search Card",
   "endpoint_session_id":"A9KKvy1mIcBC6Yx9DSm9iyjP4j4sZjJC"
}
```

Sample Response


```
{
"status": "MORE_DATA",
    "msg": "Process has been started",
    "linked_logon": false,
    "reason": "PROCESS_STARTED",
    "plugins": [],
    "msgid": "AUCORE-1008",
    "chains": [
        {
            "id_hex": "dc2b52a8b44911e98931005056a06a2a",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "Auth Fingerprint",
            "short_name": "Fingerprint Auth",
            "methods": [
                "FINGER:1"
            ],
            "position": 0,
            "apply_for_ep_owner": false,
            "image_name": "FINGER_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        },
        {
            "id_hex": "88dba848897911eca8350242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "Auth FIDO2",
            "short_name": "",
            "methods": [
                "FIDO2:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "8562cfba13cc11edbe400242ac110003",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        },
        {
            "id_hex": "8a32dcc2212511edbe400242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "Auth U2F",
            "short_name": "",
            "methods": [
                "U2F:1"
```

```
            ],
            "position": 2,
            "apply_for_ep_owner": false,
            "image_name": "U2F_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        },
        {
            "id_hex": "49f09872e0b411ecaa4b0242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "Auth FIDO2 & U2F",
            "short_name": "",
            "methods": [
                "FIDO2:1",
                "U2F:1"
            ],
            "position": 3,
            "apply_for_ep_owner": false,
            "image_name": "9f9a133e13cc11edbe400242ac110003",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "U2F:1",
    "completed_methods": [],
    "logon_process_id": "kGBKyKi3rwpxwkyVVzOOrTzodC6nLYeQ",
    "event_name": "NAAF API Testing",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "NAAF API TESTING",
    "data_id": "NAAF API TESTING",
    "category_id": "",
    "current_category_id": ""
}
```


Gather the Sign Request

To ask for the U2F sign request and associated challenge. The `endpoint_session_id` is required. The U2F device must be plugged to the system before executing this call.

API Call

`POST https://{{hostname}}/api/v1/logon/{{logon_process_id}}/do_logon`

For more information, see Common Logon (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#logon-common-do-logon-post).

Sample Request


```
{
  "endpoint_session_id": "M9VxO6CMEJfDCX8Efluu2mgqNCeHTwhZ"
}
```


Sample Response


```
{
"webauth_challenge": {
        "challenge":
"093262df46ae483a46ab30e4fb3005eef812f0f763571faa408feec9e67ebfa0",
        "allowCredentials": [
            {
                "type": "public-key",
                "id":
"508fdf1b32afe04272977726200ccb1a2dc15dccf2eec2ce0669212fe252e45b38a43a43c
212959c092ac0d83733dc1ea5d75ed32c2daeb62887e4234eded062",
                "transports": [
                    "usb"
                ]
            }
        ],
        "userVerification": "discouraged",
        "extensions": {
            "appid": "NetIQ"
        }
    },
    "signRequests": [
        {
            "version": "U2F_V2",
            "appId": "https://164.99.184.45",
            "keyHandle": "UI_fGzKv4EJyl3cmIAzLGi3BXczy7sLOBmkhL-
JS5Fs4pDpDwhKVnAkqwNg3M9wepdde0ywtrrYoh-QjTt7QYg",
            "challenge": "4ooKcq6hH4u3sG7IbF3GbPSL29Vm8bv08a3qmhBmz-Y"
        }
    ],
    "status": "MORE_DATA",
    "msg": "Waiting for an authentication response",
    "linked_logon": false,
    "reason": "U2F_WAITING_AUTH_RESPONSE",
    "plugins": [],
    "msgid": "AUCORE-2059",
    "chains": [
        {
            "id_hex": "b0e4cfd05b8d11eeb8230242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "TOTP Only",
            "short_name": "",
            "methods": [
                "TOTP:1"
            ],
```

```
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "TOTP_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        },
        {
            "id_hex": "1a1ddc6c5b7a11eeb8230242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "U2F only",
            "short_name": "",
            "methods": [
                "U2F:1"
            ],
            "position": 3,
            "apply_for_ep_owner": false,
            "image_name": "U2F_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "U2F:1",
    "completed_methods": [],
    "logon_process_id": "qEDeBxofr8fFS2OXrMiM2eJKwAkQPVvq",
    "event_name": "Search Card",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "SEARCH CARD",
    "data_id": "SEARCH CARD",
    "category_id": "",
    "current_category_id": ""
}
```


Forward Sign Requests to JSAPI

Sends the sign requests to JSAPI to activate the U2F device plugged in to the system. Locally installed Device Service handles this request. When this API is executed, the device blinks. User who is physically present is required to accept the challenge by tapping or touching on the blinking device. Finally, some details such as signature, client data, and key handle are fetched when the user taps the device.

API Call

```
POST https://127.0.0.1:8441/api/v1/fidou2f/sign
```

Sample Request

```
{
    "signRequests":
    [{
        "version":"U2F_V2",
        "appId":"https://drbiomtdevap2v.alrajhi.bank",
        "keyHandle":"dzdh2h3JCUD-hJdZ8W-CR-
SGtOmUikx01l1l3gCzxCloFTFNLdeVtCD4UpeEnswbcKp_OtQvUt0N2AeAH735pA",
        "challenge":"LRHPyefadFFXd9arONRK1ViXbuyEz59Ekqoqx0yMAx0"
    }]
}
```

Sample Response

```
{
"signatureData": "AQAAAXwwRQIhALtnK0-DhFVyW3SqRtvNmdY4-
nkNQSlj2znH59v5NrhgAiAwXGWmSG7J_Z4-2GM0OzUAYHc829uAm4cDGTAx3Wc7BA",
    "clientData":
"eyAiY2hhbGxlbmdlIjogIkxSSFB5ZWZhZEZGWGQ5YXJPTlJLMVZpWGJ1eUV6NTlFa3FvcXgwe
U1BeDAiLCAib3JpZ2luIjogImh0dHBzOlwvXC9kcmJpb210ZGV2YXAydi5hbHJhamhpLmJhbms
iLCAidHlwIjogIm5hdmlnYXRvci5pZC5nZXRBc3NlcnRpb24iIH0",
    "keyHandle": "dzdh2h3JCUD-hJdZ8W-CR-
SGtOmUikx01l1l3gCzxCloFTFNLdeVtCD4UpeEnswbcKp_OtQvUt0N2AeAH735pA"
}
```

Forward the Sign Response to Server

Sends the sign response with details of device, such as Signature data, client data, and key handle to the server to validate the details and authenticate users successfully.

API Call

```
POST https://{{hostname}}/api/v1/logon/{{logon_process_id}}/do_logon
```

For more information, see Forward the Sign Response (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#methods-u2f-1-logon-post).

Sample Request


```
{
  "response": {
    "signResponse":
    {
"signatureData":"AQAAAXwwRQIhALtnK0-DhFVyW3SqRtvNmdY4-
nkNQSlj2znH59v5NrhgAiAwXGWmSG7J_Z4-2GM0OzUAYHc829uAm4cDGTAx3Wc7BA",
"clientData":"eyAiY2hhbGxlbmdlIjogIkxSSFB5ZWZhZEZGWGQ5YXJPTlJLMVZpWGJ1eUV6
NTlFa3FvcXgweU1BeDAiLCAib3JpZ2luIjogImh0dHBzOlwvXC9kcmJpb210ZGV2YXAydi5hbH
JhamhpLmJhbmsiLCAidHlwIjogIm5hdmlnYXRvci5pZC5nZXRBc3NlcnRpb24iIH0",
"keyHandle":"dzdh2h3JCUD-hJdZ8W-CR-
SGtOmUikx01l1l3gCzxCloFTFNLdeVtCD4UpeEnswbcKp_OtQvUt0N2AeAH735pA"}
  },
  "endpoint_session_id": "M9VxO6CMEJfDCX8Efluu2mgqNCeHTwhZ"
}
```


Sample Response


```
{
"status": "OK",
    "msg": "Welcome",
    "linked_logon": false,
    "completed_template_id": "2786546914e1d25d156aef1fc099fb6e",
    "plugins": [
        "LdapRules"
    ],
    "login_session_id": "Z345oBnWuItOpubBsC2AaPBwAvzlIh4h",
    "completed_chain": {
        "id_hex": "1a1ddc6c5b7a11eeb8230242ac110003",
        "tenant_id": "def0def0def0def0def0def0def0def0",
        "is_trusted": null,
        "is_enabled": true,
        "name": "U2F only",
        "short_name": "",
        "methods": [
            "U2F:1"
        ],
        "position": 3,
        "apply_for_ep_owner": false,
        "image_name": "U2F_1.png",
        "required_chain_id_hex": null,
        "grace_period": null,
        "mfa_tags": [],
        "risk_level": "NONE"
    },
    "user_id": "b47156ea0c42a034223d0d1f13b591dc",
    "user_name": "JAGUAR\\denis",
    "repo_id": "51f5b2f636a211eeb4be0242ac110003",
    "repo_obj_id": "71a63dbbd81e724cb26048bd8bb4c3e2",
    "user_mobile_phone": "+918746825777",
    "user_email": "denisthemennies@gmail.com",
    "user_name_netbios": "JAGUAR\\denis",
    "tenant_name": "TOP",
```

```
    "user_role_assignments": "FULL ADMINS",
    "user_cn": "denis",
    "user_sid": "S-1-5-21-1982020363-492839410-2784625695-2126",
    "user_sid_hex":
"0105000000000005150000000b3b2376f221601d1f04faa54e080000",
    "user_upn": "denis@jaguar.com",
    "user_dn": "CN=denis,CN=Users,DC=jaguar,DC=com",
    "user_first_name": "denis",
    "msgid": "AUCORE-2001",
    "chains": [
        {
            "id_hex": "b0e4cfd05b8d11eeb8230242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "TOTP Only",
            "short_name": "",
            "methods": [
                "TOTP:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "TOTP_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        },
        {
            "id_hex": "1a1ddc6c5b7a11eeb8230242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "U2F only",
            "short_name": "",
            "methods": [
                "U2F:1"
            ],
            "position": 3,
            "apply_for_ep_owner": false,
            "image_name": "U2F_1.png",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "U2F:1",
```

```
    "completed_methods": [
        "U2F:1"
    ],
    "logon_process_id": "qEDeBxofr8fFS2OXrMiM2eJKwAkQPVvq",
    "event_name": "Search Card",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "SEARCH CARD",
    "data_id": "SEARCH CARD",
    "category_id": "",
    "current_category_id": "",
    "risk_level": "NONE",
    "risk_score": null
}
```

3 Authenticating with Two Factors - LDAP Password and Fingerprint Methods

Let us assume administrator of Reltic Data wants to authenticate users based on two factors, LDAP Password and Fingerprint for accessing the Self-Service Portal.

This section explains the prerequisites, sequence of API calls, and sample request details to achieve this.

Prerequisites

Ensure to gather the following details before you execute any API call:

- **{{hostname}}** indicates Advanced Authentication server IP address or host name.
- **{{endpoint_id}}** indicates the unique device ID where the user authenticates.
- **endpoint_secret_hash** indicates the value used to hide the plain text hash an endpoint secret and id.
- **salt** indicates the technique required to hash the password.
- **event_name** indicates the event name that is configured on the Administration Portal.
- **endpoint_session_id** unique session ID that is generated when the endpoint session is created.
- **logon_process_id** indicates the process ID to the given event. This is generated when you attempt to login to the event.
- Ensure Advanced Authentication Device Service is installed on the system from where you initiate API calls.

  For more information, see Installing and Upgrading Device Service.
- Ensure that the fingerprint reader mode is set appropriately based on the reader in use.

  For more information, see Fingerprint Settings.
- It is required to create a chain with LDAP Password and Fingerprint methods then associate the chain to the Self-Service Portal event.

To authenticate users with the LDAP Password and Fingerprint methods to the Authenticators Management or Self-Service Portal, the administrator need to execute APIs in the following sequence:

1. Create an Endpoint
2. Create a Session for the Endpoint
3. Initiate Logon Process
4. Verify First Factor Authenticator - LDAP Password
5. Initiate the Fingerprint Authentication

6. Capture the Fingerprint
7. Forward the Fingerprint Image to Server

Create an Endpoint

Endpoint is a device where the user authenticates. After registering an endpoint, you will get endpoint id and secret.

API Call

```
POST https://{{hostname}}/api/v1/endpoints
```

For more information, see Register New Endpoint (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#endpoints-register-new-endpoint-post).

Sample Request

```
{
   "name":"new.endpoint",
   "typ":3,
   "desc":"Endpoint",
   "is_trusted":true,
   "tenant_name":"TOP",
   "auth_data":{
      "method_id":"PASSWORD:1",
      "user_name":"LOCAL\\admin",
      "password":"novell@123"
   }
}
```

Sample Response

```
{
    "id": "7454fac094d811ee8a380242ac110003",
    "secret": "GzTcxYZ7MB5dyAYNW2pzp0hmN7rVwvp7"
}
```

Create a Session for the Endpoint

To create an endpoint session, `endpoint_secret` and `endpoint_id` from a trusted source are required. Server validates and returns `endpoint_session_id` if provided values are valid.

For more information, see Endpoint Session (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#endpoint-sessions-create-endpoint-session-).

API Call

```
POST https://{{hostname}}/api/v1/endpoints/{{endpoint_id}}/sessions
```

Sample Request

```
{
   "salt":"i_am_salt",

"endpoint_secret_hash":"e7e2dfe2209a17159689d4a731ba4b6f381843f03705c4adc7
34ea238fad8f2f"
}
```

Sample Response

```
{
    "endpoint_session_id": "PHWtR3mFb6jJxmyXiQ5TDAL9caJZepSW"
}
```

Initiate Logon Process

Initiates the logon process to the Self-Service Portal event with the LDAP Password and Fingerprint methods. The `endpoint_session_id` is required.

API Call

```
POST https://{{hostname}}/api/v1/logon
```

For more information, see Initiate Logon Process (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#logon-create-logon-process-post).

Sample Request

```
{
   "method_id":"LDAP_PASSWORD:1",
   "user_name":"denis",
    "event":"Authenticators Management",
   "endpoint_session_id":"PHWtR3mFb6jJxmyXiQ5TDAL9caJZepSW"
}
```

Sample Response


```
{
    "logon_hints": {
        "policy": {
            "cached_logon_offline_period": 60,
            "cached_logon": false,
            "sspr_enabled": false,
            "sspr_url": "",
            "sspr_caption": "",
            "disable_ldap_change": false,
            "save_password": true
        }
    },
    "status": "MORE_DATA",
    "msg": "Process has been started",
    "linked_logon": false,
    "reason": "PROCESS_STARTED",
    "plugins": [],
    "msgid": "AUCORE-1008",
    "chains": [
        {
            "id_hex": "5a6cd33c8db511ee9c500242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "LDAP + Fingerprint",
            "short_name": "",
            "methods": [
                "LDAP_PASSWORD:1",
                "FINGER:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "default",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "LDAP_PASSWORD:1",
    "completed_methods": [],
    "logon_process_id": "htP5QVP8hR2YGiO1RIIwNT4I4bmHj9HE",
    "event_name": "Authenticators Management",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "AUTHENTICATORS MANAGEMENT",
    "data_id": "AUTHENTICATORS MANAGEMENT",
    "category_id": "",
    "current_category_id": ""
}
```

Verify First Factor Authenticator - LDAP Password

Initiates the logon process to the given event with the LDAP Password method. The `endpoint_session_id` is required. User is expected to specify the password to succeed this method.

API Call

```
POST https://{{hostname}}/api/v1/logon/{{logon_process_id}}/do_logon
```

For more information, see LDAP Password Logon (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#methods-ldap_password-1-logon-post).

Sample Request

```
{
   "response":{
      "answer":"novell"
   },
   "endpoint_session_id":"PHWtR3mFb6jJxmyXiQ5TDAL9caJZepSW"
}
```

Sample Response

```
{
    "status": "NEXT",
    "msg": "Welcome. Continue with the next login method",
    "linked_logon": false,
    "plugins": [
        "LdapRules"
    ],
    "user_id": "a8e32d86c69871a507f45b2e64976138",
    "user_name": "JAGUAR\\denis",
    "repo_id": "f39e3790890411ee9c500242ac110003",
    "repo_obj_id": "71a63dbbd81e724cb26048bd8bb4c3e2",
    "user_mobile_phone": "+918746825777",
    "user_email": "dennisthemenace@gmail.com",
    "user_name_netbios": "JAGUAR\\denis",
    "tenant_name": "TOP",
    "user_role_assignments": "FULL ADMINS",
    "user_cn": "denis",
    "user_sid": "S-1-5-21-1982020363-492839410-2784625695-2126",
    "user_sid_hex":
"0105000000000005150000000b3b2376f221601d1f04faa54e080000",
    "user_upn": "denis@jaguar.com",
    "user_dn": "CN=denis,CN=Users,DC=jaguar,DC=com",
    "user_first_name": "denis",
    "msgid": "AUCORE-1091",
    "chains": [
        {
            "id_hex": "5a6cd33c8db511ee9c500242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
```

```
            "is_trusted": null,
            "is_enabled": true,
            "name": "LDAP + Fingerprint",
            "short_name": "",
            "methods": [
                "LDAP_PASSWORD:1",
                "FINGER:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "default",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "LDAP_PASSWORD:1",
    "completed_methods": [
        "LDAP_PASSWORD:1"
    ],
    "logon_process_id": "htP5QVP8hR2YGiO1RIIwNT4I4bmHj9HE",
    "event_name": "Authenticators Management",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "AUTHENTICATORS MANAGEMENT",
    "data_id": "AUTHENTICATORS MANAGEMENT",
    "category_id": "",
    "current_category_id": ""
}
```


Initiate the Fingerprint Authentication

To initiate the Fingerprint authentication. The `endpoint_session_id` is required. Ensure the fingerprint device is plugged to the system before executing this call.

API Call

```
POST https://{{hostname}}/api/v1/logon/{{logon_process_id}}/next
```

Sample Request


```
{
   "method_id":"FINGER:1",
   "endpoint_session_id":"PHWtR3mFb6jJxmyXiQ5TDAL9caJZepSW"
}
```

Sample Response


```
{
    "logon_hints": {},
    "status": "MORE_DATA",
    "msg": "Process has been started",
    "linked_logon": false,
    "reason": "PROCESS_STARTED",
    "plugins": [],
    "msgid": "AUCORE-1008",
    "chains": [
        {
            "id_hex": "5a6cd33c8db511ee9c500242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "LDAP + Fingerprint",
            "short_name": "",
            "methods": [
                "LDAP_PASSWORD:1",
                "FINGER:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "default",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "FINGER:1",
    "completed_methods": [
        "LDAP_PASSWORD:1"
    ],
    "logon_process_id": "htP5QVP8hR2YGiO1RIIwNT4I4bmHj9HE",
    "event_name": "Authenticators Management",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "AUTHENTICATORS MANAGEMENT",
    "data_id": "AUTHENTICATORS MANAGEMENT",
    "category_id": "",
    "current_category_id": ""
}
```


Capture the Fingerprint

To activate the Fingerprint device plugged in to the system. Locally installed Device Service handles this request. When this API is executed, there is a call to the device. User who is physically present is required to place the enrolled finger on the Fingerprint device. Device Service captures the fingerprint image.

API Call

```
GET https://127.0.0.1:8442/api/v1/fingerprint/capture
```

Sample Response

```
{
    "ISO":
"Rk1SACAyMAAAAAEsAAABAAC0AMUAxQEAAANaLQAPAEsvYwAXAGTQYwAcAF9HYwAlAC6QYwAlA
EyvYwAmAH93YwAuADKQYwAwAGt3YwAxAFf3YwA0AKAQYwA3ADEHYwA5AEWAYwA+AKUQYwBBAIq
HYwBIAKkQYwBQADVoYwBRAGSYYwBTACvoYwBVAKoYYwBeAI6YYwBfAKoYYwBhAGO3MgBiACHfY
wBkACvYYwBqAKofYwB0AC/YYwB4AHWvYwB5AEZQYwB5AGE3YwB7ABjYYwB9AE/
HYwCCABTYYwCMABlYYwCOAHG3YwCYAGo3YwCZAJMvYwCbACHQYwCbAEtHYwCmAEjHYwDKAJcvY
wDKAKkvYwDOAGY3YwDWAFRAYwDXAFzAYwDrAHq3YwAA",
    "captureStatus": "Ok"
}
```

Forward the Fingerprint Image to Server

Sends the fingerpint image to the server to validate the details and authenticate users successfully.

API Call

```
POST https://{{hostname}}/api/v1/logon/{{logon_process_id}}/do_logon
```

For more information, see Fingerprint Logon (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#methods-finger-1-logon-post).

Sample Request

```
{
  "endpoint_session_id": "PHWtR3mFb6jJxmyXiQ5TDAL9caJZepSW",
  "response": {
    "capture": {
      "ISO":
"Rk1SACAyMAAAAAEsAAABAAC0AMUAxQEAAANaLQAPAEsvYwAXAGTQYwAcAF9HYwAlAC6QYwAlA
EyvYwAmAH93YwAuADKQYwAwAGt3YwAxAFf3YwA0AKAQYwA3ADEHYwA5AEWAYwA+AKUQYwBBAIq
HYwBIAKkQYwBQADVoYwBRAGSYYwBTACvoYwBVAKoYYwBeAI6YYwBfAKoYYwBhAGO3MgBiACHfY
wBkACvYYwBqAKofYwB0AC/YYwB4AHWvYwB5AEZQYwB5AGE3YwB7ABjYYwB9AE/
HYwCCABTYYwCMABlYYwCOAHG3YwCYAGo3YwCZAJMvYwCbACHQYwCbAEtHYwCmAEjHYwDKAJcvY
wDKAKkvYwDOAGY3YwDWAFRAYwDXAFzAYwDrAHq3YwAA",
      "captureStatus": "OK"
    }
  }
}
```

Sample Response


```
{
    "score": 45,
    "status": "OK",
    "msg": "Welcome",
    "linked_logon": false,
    "reason": "",
    "plugins": [
        "LdapRules"
    ],
    "login_session_id": "F6NncwO3cXxgwoo6UWMfWKOQhfknhN4Z",
    "completed_chain": {
        "id_hex": "5a6cd33c8db511ee9c500242ac110003",
        "tenant_id": "def0def0def0def0def0def0def0def0",
        "is_trusted": null,
        "is_enabled": true,
        "name": "LDAP + Fingerprint",
        "short_name": "",
        "methods": [
            "LDAP_PASSWORD:1",
            "FINGER:1"
        ],
        "position": 1,
        "apply_for_ep_owner": false,
        "image_name": "default",
        "required_chain_id_hex": null,
        "grace_period": null,
        "mfa_tags": [],
        "risk_level": "NONE"
    },
    "user_id": "a8e32d86c69871a507f45b2e64976138",
    "user_name": "JAGUAR\\denis",
    "repo_id": "f39e3790890411ee9c500242ac110003",
    "repo_obj_id": "71a63dbbd81e724cb26048bd8bb4c3e2",
    "user_mobile_phone": "+918746825777",
    "user_email": "denisthemennies@gmail.com",
    "user_name_netbios": "JAGUAR\\denis",
    "tenant_name": "TOP",
    "user_role_assignments": "FULL ADMINS",
    "user_cn": "denis",
    "user_sid": "S-1-5-21-1982020363-492839410-2784625695-2126",
    "user_sid_hex":
"0105000000000005150000000b3b2376f221601d1f04faa54e080000",
    "user_upn": "denis@jaguar.com",
    "user_dn": "CN=denis,CN=Users,DC=jaguar,DC=com",
    "user_first_name": "denis",
    "msgid": "AUCORE-2001",
    "chains": [
        {
            "id_hex": "5a6cd33c8db511ee9c500242ac110003",
            "tenant_id": "def0def0def0def0def0def0def0def0",
            "is_trusted": null,
            "is_enabled": true,
            "name": "LDAP + Fingerprint",
```

```
            "short_name": "",
            "methods": [
                "LDAP_PASSWORD:1",
                "FINGER:1"
            ],
            "position": 1,
            "apply_for_ep_owner": false,
            "image_name": "default",
            "required_chain_id_hex": null,
            "grace_period": null,
            "mfa_tags": [],
            "risk_level": "NONE"
        }
    ],
    "categories": null,
    "current_method": "FINGER:1",
    "completed_methods": [
        "LDAP_PASSWORD:1",
        "FINGER:1"
    ],
    "logon_process_id": "htP5QVP8hR2YGiO1RIIwNT4I4bmHj9HE",
    "event_name": "Authenticators Management",
    "event_type": "Generic",
    "event_all_categories": true,
    "event_data_id": "AUTHENTICATORS MANAGEMENT",
    "data_id": "AUTHENTICATORS MANAGEMENT",
    "category_id": "",
    "current_category_id": "",
    "risk_level": "NONE",
    "risk_score": null
}
```