D-Link®
Building Networks for People

Network Security Firewall REST API Guide

NetDefendOS

Ver. 11.10.01

Network Security Solution http://www.dlink.com

NetDefend

REST API Guide

DFL-260E/860E/870/1660/2560/2560G

NetDefend Version 11.10.01

D-Link Corporation
No. 289, Sinhu 3rd Rd, Neihu District, Taipei City 114, Taiwan R.O.C.
http://www.DLink.com

Published 2016-12-21

REST API Guide
DFL-260E/860E/870/1660/2560/2560G

NetDefendOS Version 11.10.01

Published 2016-12-21

Copyright Notice

Disclaimer

The information in this document is subject to change without notice. D-Link makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. D-Link reserves the right to revise this publication and to make changes from time to time in the content hereof without any obligation to notify any person or parties of such revision or changes.

Limitations of Liability

UNDER NO CIRCUMSTANCES SHALL D-LINK OR ITS SUPPLIERS BE LIABLE FOR DAMAGES OF ANY CHARACTER (E.G. DAMAGES FOR LOSS OF PROFIT, SOFTWARE RESTORATION, WORK STOPPAGE, LOSS OF SAVED DATA OR ANY OTHER COMMERCIAL DAMAGES OR LOSSES) RESULTING FROM THE APPLICATION OR IMPROPER USE OF THE D-LINK PRODUCT OR FAILURE OF THE PRODUCT, EVEN IF D-LINK IS INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. FURTHERMORE, D-LINK WILL NOT BE LIABLE FOR THIRD-PARTY CLAIMS AGAINST CUSTOMER FOR LOSSES OR DAMAGES. D-LINK WILL IN NO EVENT BE LIABLE FOR ANY DAMAGES IN EXCESS OF THE AMOUNT D-LINK RECEIVED FROM THE END-USER FOR THE PRODUCT.

Table of Contents

Chapter 1: Introduction

The NetDefendOS software provides a REST API which allows certain aspects of its operation to be controlled by an external computer. At this time, only the NetDefendOS authentication subsystem can be controlled in this way.

The REST API functions by an external computer sending HTTP or HTTPS messages to the IP address of a NetDefendOS Ethernet interface over an IP network. NetDefendOS then processes the message and returns any reply in an HTTP/HTTPS message. Any data sent in messages uses the JSON format.

Note: JSON data is shown reformatted for readability

The JSON data sent back by NetDefendOS will be one continuous stream of characters without white space. The JSON output shown in this publication has been reformatted to make it more readable.

Enabling REST API Access in NetDefendOS

Allowing access by the REST API must be explicitly configured in NetDefendOS by creating a *RemoteManagementREST* object. The properties of this object determine what restrictions are placed on REST API access. These properties include the following:

- Protocol used (HTTP or HTTPS). Encrypting messages with HTTPS is recommended.
- The interface to accept connections from external clients on. The external computer will send REST API messages to the IP address of this interface.
- The source IP(s) to accept connections from external clients on.
- The access level permitted to clients (read only or read/write).
- The option to require REST API authentication using a *basic access authentication* schema with username/password credentials.

An Example of Enabling the REST API

Before the REST API can be used, it must be enabled by creating a *REST API Management* object in the NetDefendOS configuration and setting the properties of this object to the appropriate values. The example below shows how this is done.

Example 1.1. Enabling the REST API

This example will create a new *REST_managment* object called *my-rest-api*. This will allow read/write HTTPS communication only on the *if3* interface and only from the network *mgmt-net*. Credentials will be required for access with a username of *my-admin-name* and password of *my-admin-pass*.

Command-Line Interface

```
Device:/> add RemoteManagement RemoteMgmtREST
                        Name=my-rest-api
                        Interface=if3
                        Network=mgmt-net
                        HTTPS=Yes
                        AccessLevel=ReadWrite
                        BasicAUTH=Yes
                        Username=my-admin-name
                        Password=my-admin-pass
```

Web Interface

- Go to: **System > Device > Remote Management**
- Select **Add > REST API Management**
- Enter **Name:** my-rest-api
- Enable **HTTPS**
- Enter **Interface:** if3
- Enter **Network:** mgmt-net
- Select **Access Level** to be *Read/Write*
- Enable **Basic Auth**
- Enter **Username:** my-admin-name
- Enter **Password:** my-admin-pass
- Click **OK**

Chapter 2: Authentication

This section describes the usage of the NetDefendOS REST API with authentication. When the API is used with authentication, an external computer can perform the following management operations on a NetDefendOS configuration:

- List currently authenticated users.
- Retrieve information for a single currently authenticated user.
- Add a new authenticated user.
- End authentication of user.

A REST API Authentication Use Case

A typical use case for the REST API with authentication is where an external DHCP server is handing out IP addresses to clients on a private WiFi network in a public space such as a train station or airport. The clients might access resources such as the public Internet via a NetDefend Firewall.

As the DHCP server allocates IP addresses, it therefore needs to add and delete and possibly list the authenticated users in NetDefendOS. This can be done by using the REST API.

The sections that follow describe how to perform these operations.

2.1. Listing Currently Authenticated Users

To list information about all currently authenticated users, an HTTP *GET* must be sent to the URI */api/oper/userauth*. The *GET* can have only the following optional parameter:

- **num** - The maximum number of users to return (default value: 100).

To retrieve a list of all currently authenticated users, perform a GET towards the following URI:

```
/api/oper/userauth
```

NetDefendOS will send its reply back in JSON format. Below is an example of a reply to a *GET*:

```
{
  "error": false,
  "active_users_count": 2,
  "active_users": [
    {
      "username": "user1",
      "ip": "203.0.113.5",
      "groups": "group1,group2",
      "interface": "wan",
      "agent_type": "Identity Awareness",
      "session_timeout": 60,
      "idle_timeout": 60
    },
    {
      "username": "user2",
      "ip": "203.0.113.7",
      "groups": "group4,group5",
      "interface": "wan",
      "agent_type": "Identity Awareness",
      "session_timeout": 160,
      "idle_timeout": 60
    }
  ]
}
```

In the above JSON reply, there are 2 users listed with usernames *user1* and *user2*. The user called *user1* has an IPv4 address of *203.0113.5* and belongs to the groups *group1* and *group2*. The user called *user2* has an IPv4 address of *203.0113.7* and belongs to the groups *group4* and *group5*. Both users have connected to NetDefendOS through the *wan* interface.

The *agent-type* value indicates the origin of the authenticated user. If a user was added to the authenticated list with the REST API then its *agent-type* value will always be *"Identity Awareness"*.

If there are no authenticated users, the response will be the following:

```
{
  "error":false,
  "active_users_count":0,"active_users":[]
}
```

2.2. Listing Information about a Single Authenticated User

To list information about a single currently authenticated user, an HTTP *GET* is sent to the path */api/oper/userauth* but with additional parameters to identify the user. The action can have the following parameters:

- **ip** - The IPv4 address of the user.
- **interface** - The NetDefendOS interface through which the client connected. (Optional)

For example, to list a single user, a *GET* could be sent to the following URI:

```
/api/oper/userauth?ip=203.0.113.5&interface=wan
```

Here, the IP address *203.0.113.5* is the IP address of the user and *wan* is the NetDefendOS interface that the user has connected to.

The JSON reply sent by NetDefendOS will have the following form:

```
{
  "error": false,
  "active_users": [
    {
      "username": "user1",
      "ip": "203.0.113.5",
      "groups": "group1,group2",
      "interface": "wan",
      "agent_type": "Identity Awareness",
      "session_timeout": 60,
      "idle_timeout": 60
    }
  ]
}
```

The ouput is similar in form to the output when a list of all authenticated users is retrieved.

If the specified user cannot be found, a JSON message like the following will be returned:

```
{
  "error":false,
  "active_users_count":5,
  "active_users":[]
}
```

Here, the *active_users* parameter has an empty value. Note also that the *active_users_count* value is the total number of currently authenticated users.

2.3. Adding a New Authenticated User

To add a new authenticated user, the action received by NetDefendOS should be an HTTP *POST* to the path */api/oper/userauth*. The action can have the following parameters:

- **ip** - The IPv4 address of the user.
- **username** - The username of the client.

- **interface** - The NetDefendOS interface through which the client connected. (Optional)
- **session_timeout** - The session timeout in seconds for this user.
- **groups** - A comma separated list of groups to which this user belongs. This is not optional and if there is no group membership then the parameter must be specified without a value.

For example, send a *POST* to the following URI:

```
/api/oper/userauth
```

The *POST* body should contain a string in the following form:

```
ip=192.168.1.10&interface=wan&session_timeout=60
&username=user3&groups=group1,group2
```

Note that the above *POST* body has been split into 2 lines to fit this document's page width. In reality, these last two lines would be a single continuous line.

In the above body, the username of the new authenticated user is specified as *user2* with an IP address of *192.168.1.10*. This user can only connect to NetDefendOS via the *wan* interface and they are a member of the groups called *group1* and *group2*. The client session timeout in seconds can be set as required. Here, it is 60 seconds.

The JSON reply sent by NetDefendOS will confirm that the user has been added and specify the parameter values used for the user. It will have the following form:

```
{
  "error": false,
  "user": {
    "username": "user1",
    "ip": "192.168.1.10",
    "groups": "group1, group2",
    "interface": "wan",
    "agent_type": "Identity Awareness",
    "session_timeout": 60,
    "idle_timeout": 60
  }
}
```

If there is an error, NetDefendOS will return a message indicating the problem. For example, if the interface name is incorrect, the following JSON message would be sent back to the client:

```
{
  "error":true,
  "error_code":400,
  "error_message":"Property 'interface': Unknown interface name"
}
```

2.4. Ending Authentication of a User

In order to delete a user from the list of currently authenticated users, the REST API client must send an HTTP *DELETE* to the URI */api/oper/userauth*. The action can have the following parameters:

- **ip** - The IPv4 address of the user.
- **interface** - The NetDefendOS interface through which the client connected. (Optional)

For example, to remove a user with the IP address *192.168.1.201*, send a DELETE command to the URI:

```
/api/oper/userauth?ip=192.168.1.201
```

The *Content-Type* for this should be *application/x-www-form-urlencoded*.

The following JSON reply will be sent back by NetDefendOS if the deletion is successful:

```
{
  "error": false
}
```

If the user was not found, NetDefendOS will return the following JSON message:

```
{
  "error": true,
  "error_code": 404,
  "error_message": "User does not exist"
}
```

The above message indicates that the no user with the given IP address (and optionally the given interface) is currently authenticated.