Progress®
Kemp® LoadMaster®

Technical Note Form based to Form based Authentication

Generated on: 16 September 2024

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: Product Documentation Copyright Notice & Trademarks | Progress

Table of Contents

Chapter 1: Introduction 4

Chapter 2: How Form-based to Form-based Authentication Works . . . 5

Chapter 3: Configure Form-based to Form-based Authentication 7

1

Introduction

Introduction

When using a LoadMaster, it is possible to use form-based authentication as both the client and server authentication modes. Customers want to use form-based authentication as the server-side authentication type for several reasons:

- This functionality was previously supported by Microsoft's Threat Management Gateway (TMG) and some customers have existing implementations that depend on this configuration for legacy and security reasons.
- Server-side form-based authentication addresses some outstanding issues relating to password modification.
- There is a long-standing issue with the log-off string not working for Exchange 2016 and a recent Microsoft change has rendered a workaround to this issue obsolete. Using form-based authentication as the server-side authentication type is a more resilient solution.

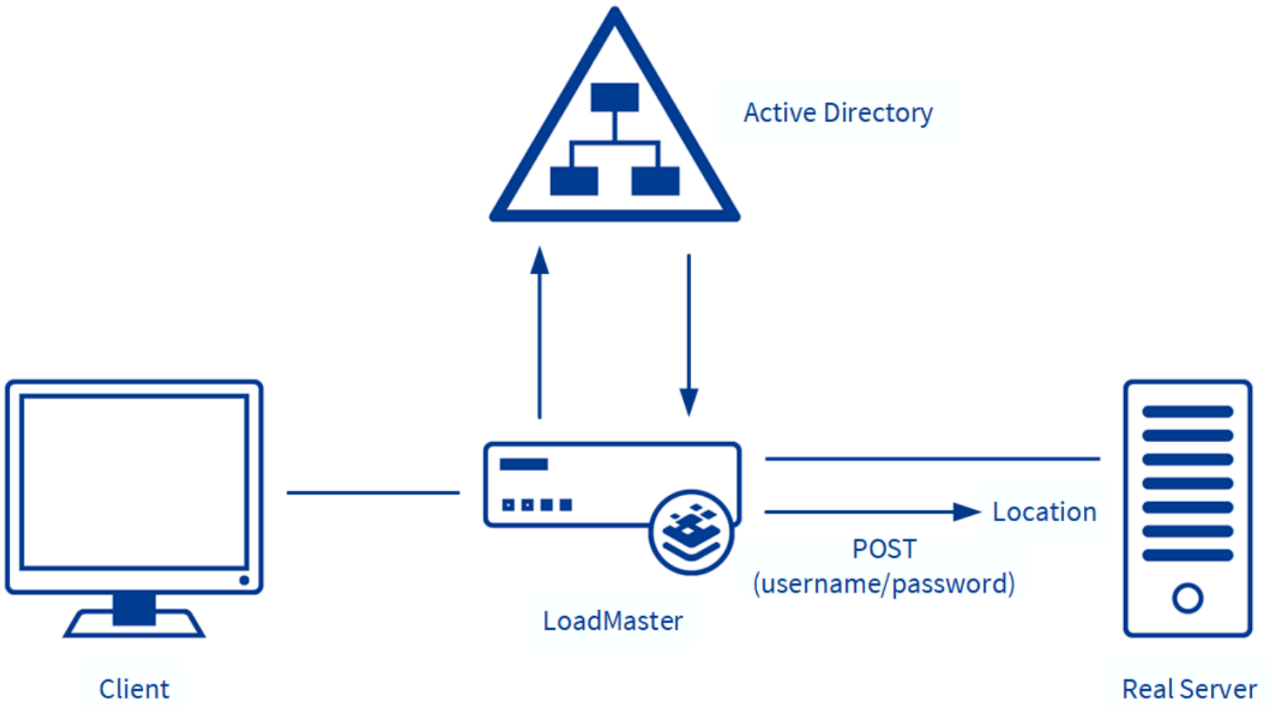
2

How Form-based to Form-based Authentication Works

How Form-based to Form-based Authentication Works

The diagram below depicts how form-based to form-based authentication works:

1. The LoadMaster receives the connection from the client.
2. The LoadMaster uses the Active Directory to authenticate the user.
3. The username and password from the client-side, form-based authentication gets injected into the form POST format to build the POST body. This POST is sent to a specific location. This logs the user in, as if they had logged in using a standard login page.


Active Directory
Client
LoadMaster
Location
POST
(username/password)
Real Server

3

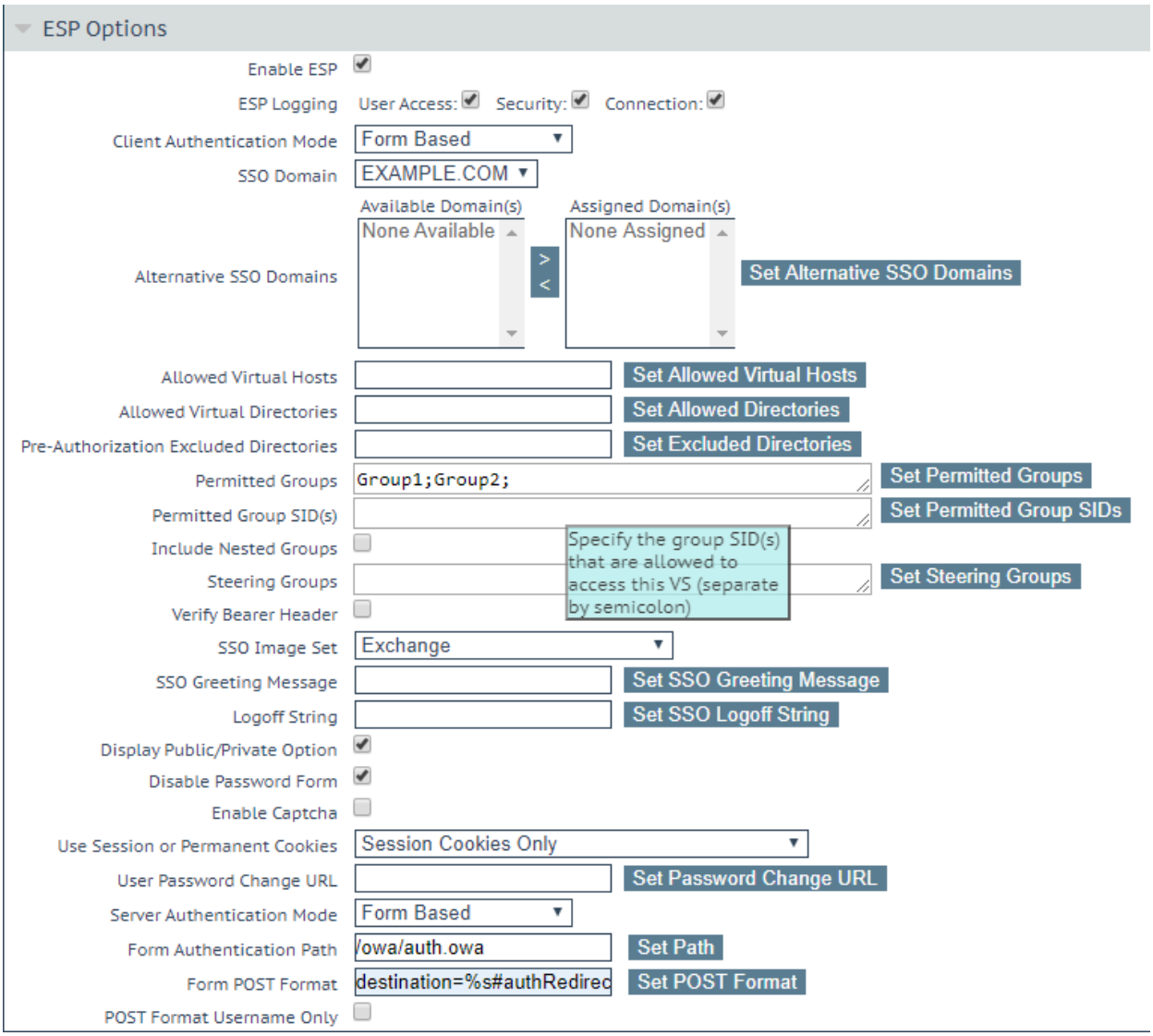
Configure Form-based to Form-based Authentication

Configure Form-based to Form-based Authentication

To configure form-based to form-based authentication on the LoadMaster, first create a client-side Single Sign On (SSO) domain (**Virtual Services > Manage SSO**). Refer to the **Edge Security Pack (ESP) Feature Description** on the Documentation Page for further information.

Then, configure the Virtual Service by following the steps below in the LoadMaster Web User Interface (WUI):

1. In the main menu, go to **Virtual Services > View/Modify Services**.
2. Click **Modify** on the relevant Virtual Service.
3. Expand the **ESP Options** section.



4. Select the **Enable ESP** check box.
5. Select **Form Based** as the **Client Authentication Mode**.
6. Select the relevant **SSO Domain**.
7. Select **Form Based** as the **Server Authentication Mode**.
8. Enter the **Form Authentication Path** and click **Set Path**.
9. Enter the **Form POST Format** and click **Set POST Format**.

This feature is predominantly used in Microsoft Exchange deployments and it has only been tested with Exchange 2013 and 2016. The following strings do not need to be explicitly configured for Exchange 2013/ 2016. They are used by default in the implementation:

- **Form Authentication Path:** /owa/auth.owa
- **Form POST Format:** destination=%s#authRedirect=true&flags=4&forcedownlevel=0&username=%s&password=%s&passwordText=&isUtf8=1

If the deployment is not Exchange, we recommend that the settings are evaluated based on the required

interaction with the Real Server and subsequently set appropriately.

POST Body Format

The server may require a specific POST body format. Essentially, this is the POST with the user credentials in the body.

The credentials are collected from the client (the username and password collected on the Progress Kemp SSO form) and they are printed into this string, along with the destination string (URL string).

To identify POST content for other target Real Servers, you could:

- Connect the client directly to the Real Server
- Use Fiddler, if necessary
- Use the form-based authentication from the Real Server
- Examine what is POSTed to the Real Server when submitting the credentials

For example, if the POST looks like:

example=example&user=JBLOGGS&password=INSECUREPASS&example2=example2

The configuration string should be:

destination=%s&example=example&user=%s&password=%s&example2=example2

Note: "Destination" is not present in the hypothetical real POST, but the holder must be in the configuration string. IF a "destination"-like field is present in the real POST, include it as accurately as possible.

If there is a requirement for more than three dynamic pieces of information, then it will likely not work.