Tetherfi User Session Manager

DEPLOYMENT GUIDE

Document Version: 1.1
Last Updated Date: 2020 April 7

tetherfi

Version History

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
1.0	Ajit	05/10/2018			Initial Draft
1.1	Shreyas K R	31/03/2020	Prathik	03/04/2020	Updated Configuration Keys, TUserSessionManager Service Setup and Setting up Securing endpoints. Added Certificate Binding section.

Copyright

Warranty Information

Tetherfi makes no representations or warranties with respect to the contents or use of this product and specifically disclaims any express or implied warranties of merchantability or fitness for any purpose. Information in this document is subject to change without notice and does not represent a commitment on the part of Tetherfi.

Trademark Information

Tetherfi is a registered trademark of Tetherfi. Other product names may be trademarks or registered trademarks of their respective companies.

Contact Information

For customer inquiries, contact:

Tetherfi

60 Paya Lebar Road,

#06-01 Paya Lebar Square,

Singapore

For technical support inquiries, call:

+65 6715 7048

Contents

1. Introduction 5
2. Deployment 5
2.1 Release Details 5
2.2 Configuration 5
2.3 TUserSessionManager Service Setup 6
2.4 Setting up TUserSessionManager Service (HA) 6
2.5 Setting up WCF REST service 7
2.6 Creating a certificate and enabling the Https settings to use the certificate 7
2.7 Bind certificate using command line 9
2.8 Setting up Securing Endpoints 10
3. Testing TUserSessionManager 11

1. Introduction

Tetherfi User Session Manager (TUserSessionManager), a component to maintain logged in user session handles timeout and provide the same for any application.

TUserSessionManager, exposes WCF service as well as REST API interface to interact with Tetherfi application.

To integrate using WCF service, the application can use TUserSessionManagerClient.dll file as it is a wrapper for the TUserSessionManager WCF service.

2. Deployment

2.1 Release Details

Before starting deployment of TUserSessionManager, checkout the build from SVN repository as shown below. Take the build from base release folder which will have latest version of TUserSessionManager

svn://repo.tetherfi.com/InterLink/Products/TUserSessionManager/Release/Base Binaries

svn://repo.tetherfi.com/InterLink/Products/TUserSessionManager/Release/ReleaseNotes.xlsx

2.2 Configuration

Key	Description	Possible Values
Log4NetConfigFile	Log4Net Config File path.	E:\InterLink\Products\TUserSessionManager\TUserSessionManager\Libs\Log4Net.config
ConnectionString	Kms Encrypted Connection String.	Data Source=xx.xxx.xxx.xx\SQLEXPRESS,xxxxx; Initial Catalog=OCM;User ID=abc; Password=xxxxxxxx
KmsEnabled	Is Kms Enabled.	0 or 1
KmsUrl	Kms Url.	https://xxx.xxxxxxxx.com:xxxxx/kms/
KmsCert	Kms Certificate.	5dxxxxxxxec3b928846xxxxxxxxe5a8ecab
AppTimeout	Application Timeout in minutes.	20
RestApiEnabled	Rest Api Enabled.	0 or 1
SyncUserSessionEnabled	Sync User Session Enabled.	0 or 1
UserSessionManagerSyncServerCount	User Session Sync Servers Count.	2 (server count)

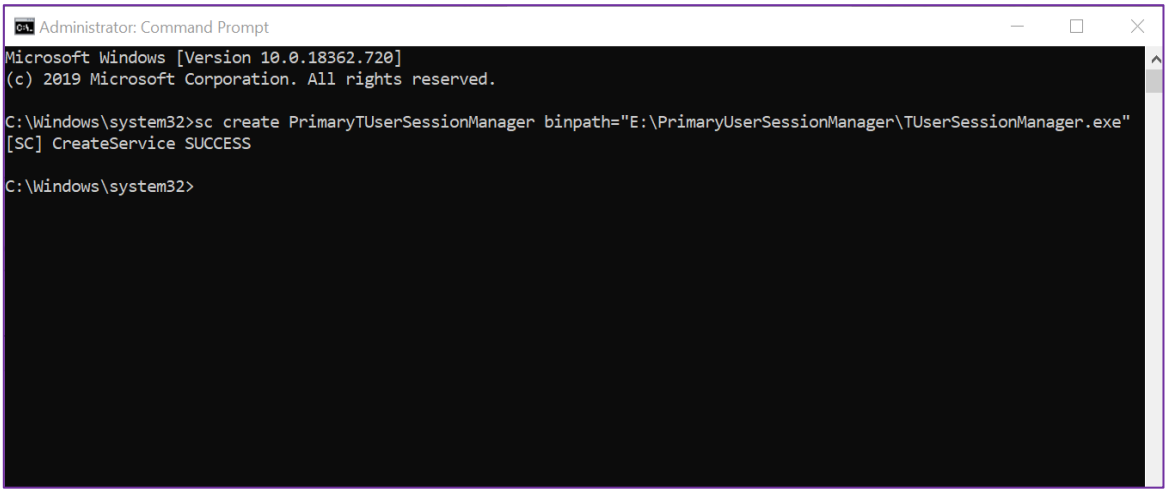
TmacAuthServerEnabled	Tmac Auth Server Enabled.	0 or 1
RemoveOldSessionOnNewSession	Remove old Session on New Session recieved for same user.	0 or 1
SslVersion	Version for Ssl.	Tls12

2.3 TUserSessionManager Service Setup

Create the service using Command Prompt as Administrator with below command:

sc create service_name binPath=" full path of TUserSessionManager.exe

Eg: sc create PrimaryTUserSessionManagerbinPath="E:\PrimaryUserSessionManager \TUserSessionManager.exe"



2.4 Setting up TUserSessionManager Service (HA)

Configure the service by following the above step and do the below config file changes in this service:

- The high availability of TUserSessionManager service is achieved by installing more than one instances of service with proper config changes
- The concept used here is that Service1 and other Service instances (Service2 and Service3) are internally connected and referenced by service endpoints in config with respect to base addresses.
- When one service stops, one of the other services listening will start execution
- The number of service instances to be installed are to be configured with respect to main Service created (Service1).
- This is configured with "UserSessionManagerSyncServerCount" key value as below. If value of the key is 2, meaning along with main service running, two other instances are to be created so totally 3 services will be running simultaneously listening to each other's endpoints.

  <! -- User Session Sync Servers Count-->

  <add key="UserSessionManagerSyncServerCount" value="1"/>

- The service endpoints to be configured as below:
- The config for main service with base address as "…./***PrimaryTetherfiUserSessionManager***" in service end point (as highlighted below)

```
<service name="TUserSessionManager.UserSessionManager" behaviorConfiguration="soap" >
  <endpoint name="soap" address="https:[redacted]/PrimaryTetherfiUserSessionManager"
    binding="basicHttpBinding" bindingConfiguration="basicHttps"
    contract="TUserSessionManager.IUserSessionManager" />
  <host>
    <baseAddresses>
      <add baseAddress="https://[redacted]/PrimaryTetherfiUserSessionManager" />
    </baseAddresses>
  </host>
</service>
```

- The config for main service (***PrimaryTetherfiUserSessionManager***) listening to another service (***SecondaryTetherfiUserSessionManager***) client as **UserSessionManagerServer1**

```
<client>
    <endpoint address="https://[redacted]/SecondaaryTetherfiUserSessionManager"
    binding="basicHttpBinding" bindingConfiguration="basicHttps"
    contract="TetherfiUserSessionManager.IUserSessionManager" name="UserSessionManagerServer1"/>
</client>
```

- Similarly other service with base address as "…./***SecondaryTetherfiUserSessionManager***" listening to another service (***PrimaryTetherfiUserSessionManager***) client as **UserSessionManagerServer1**

2.5 Setting up WCF REST service

- REST API can be enabled by below key

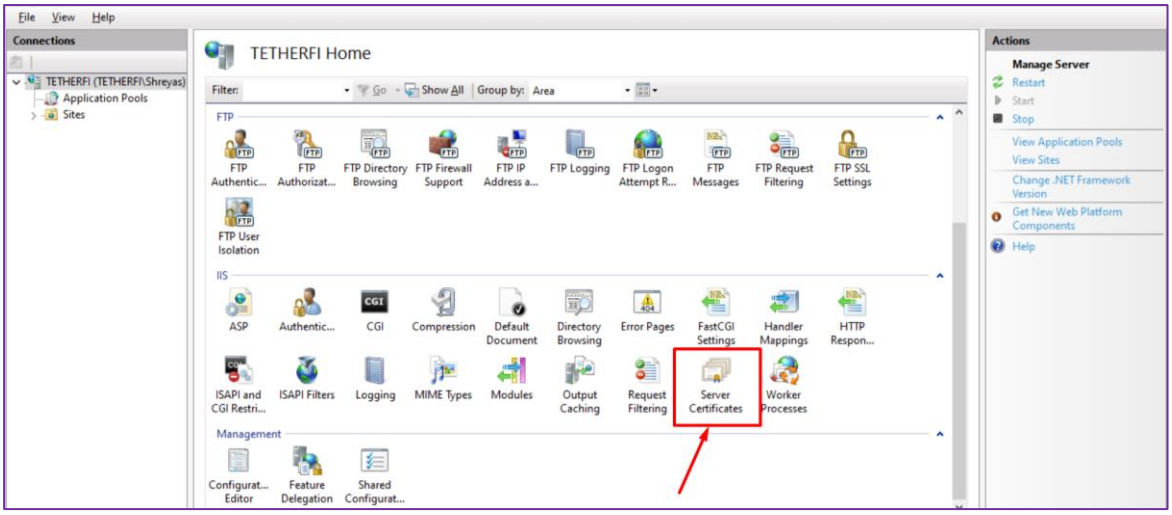
  <! -- Rest API Enabled -->

  <add key="RestApiEnabled" value="1" />
- Configure the base address for RES API as below

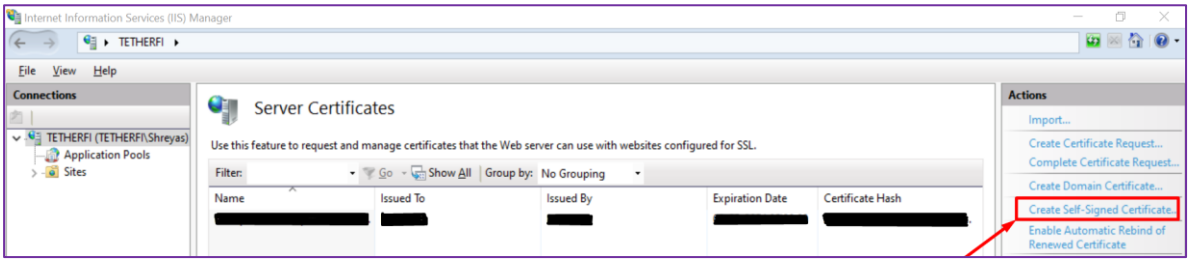
```
<service name="TUserSessionManager.UserSessionManagerRest"  behaviorConfiguration="rest">
  <endpoint name="rest" address="https://[redacted]/PrimaryTetherfiUserSessionManager"
    binding="webHttpBinding" bindingConfiguration="webHttp"
    contract="TUserSessionManager.IUserSessionManager" behaviorConfiguration="rest" />
  <host>
    <baseAddresses>
      <add baseAddress="https://[redacted]/PrimaryTetherfiUserSessionManager" />
    </baseAddresses>
  </host>
</service>
```

2.6 Creating a certificate and enabling the Https settings to use the certificate

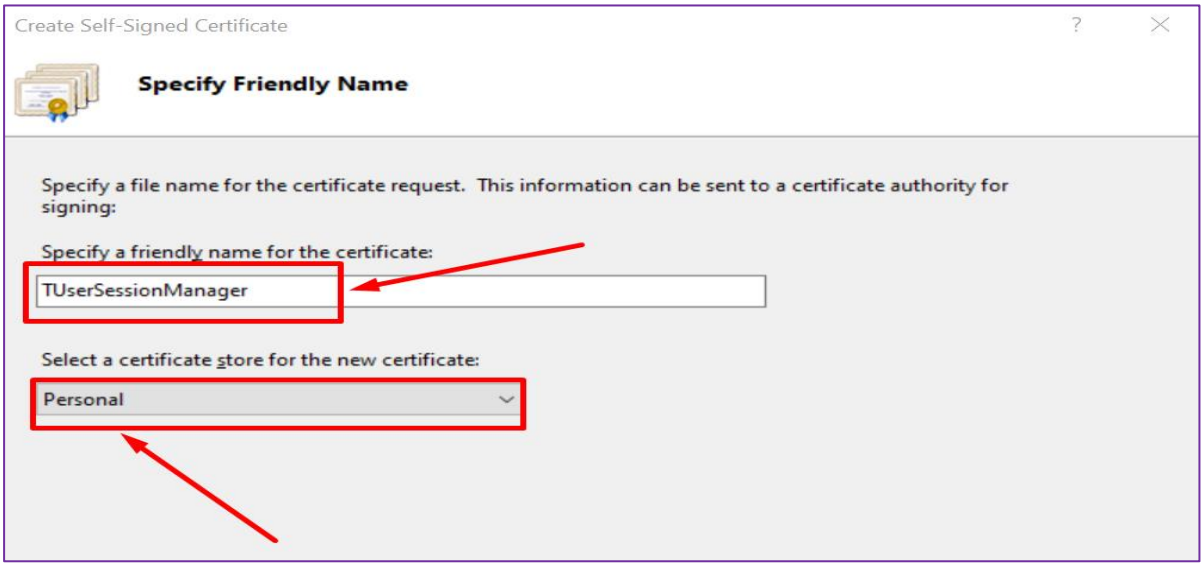
- Go to IIS home directory and double click on **Server Certificates** option.



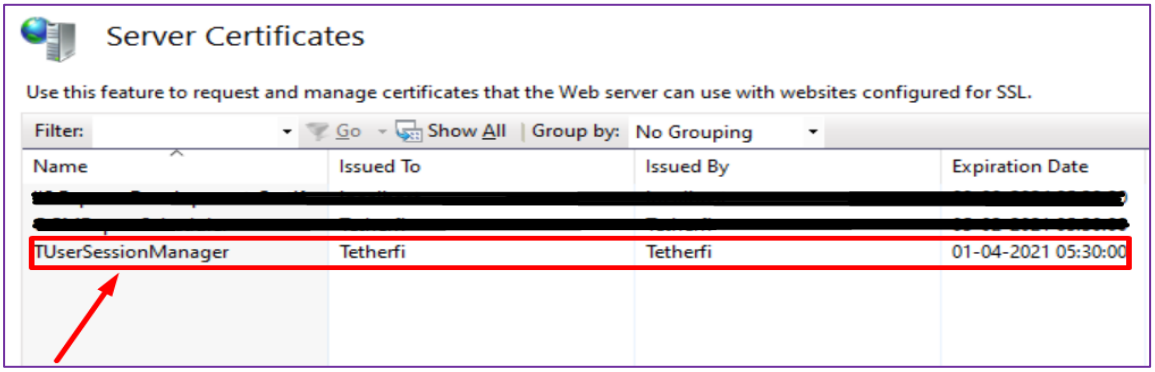
- Create Self-Signed certificate.



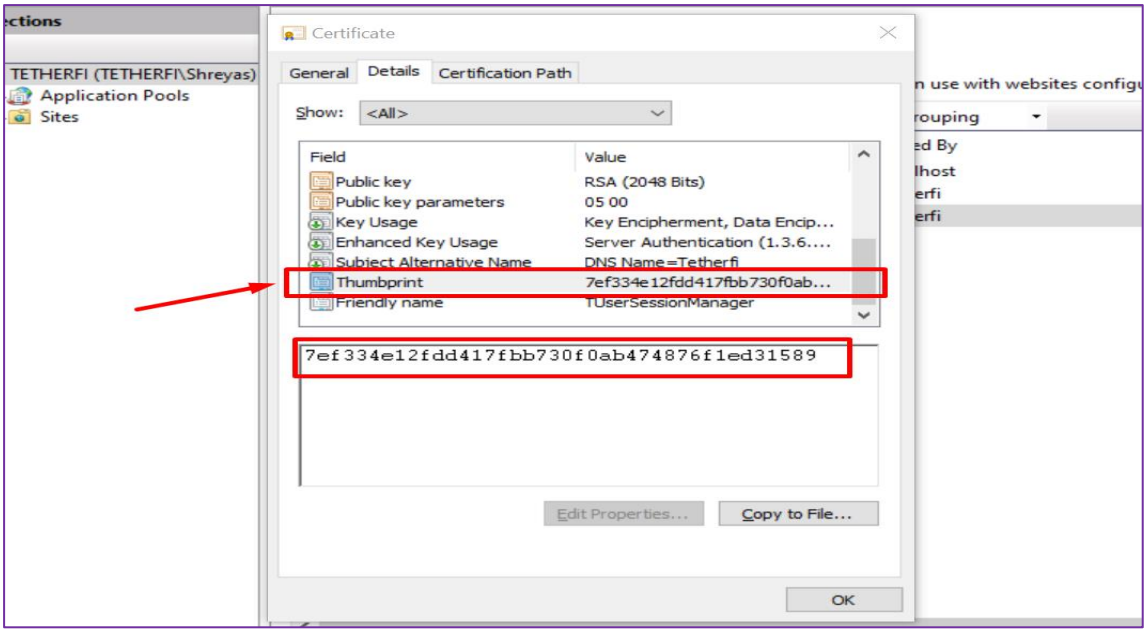
- Enter Friendly Name for certificate (i.e. TUserSessionManager) and Select **Personal** as Certificate store.



- Created certificate Successfully.
- Double click on the created **Server certificate**.



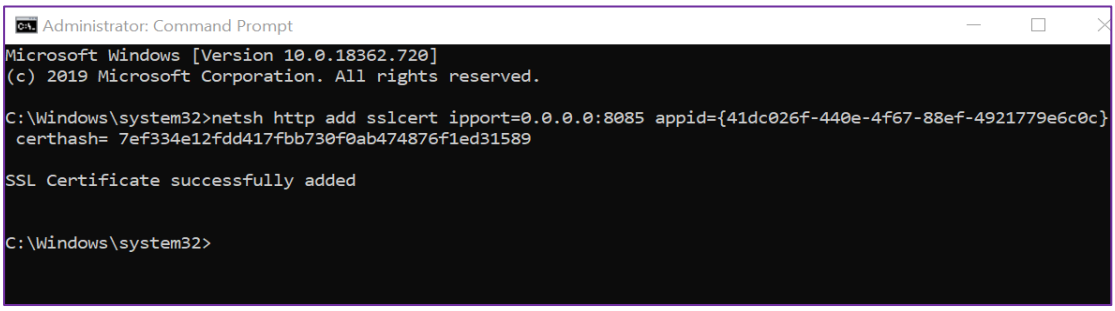
- On Clicking **Details**, you will find the **Thumbprint** field.



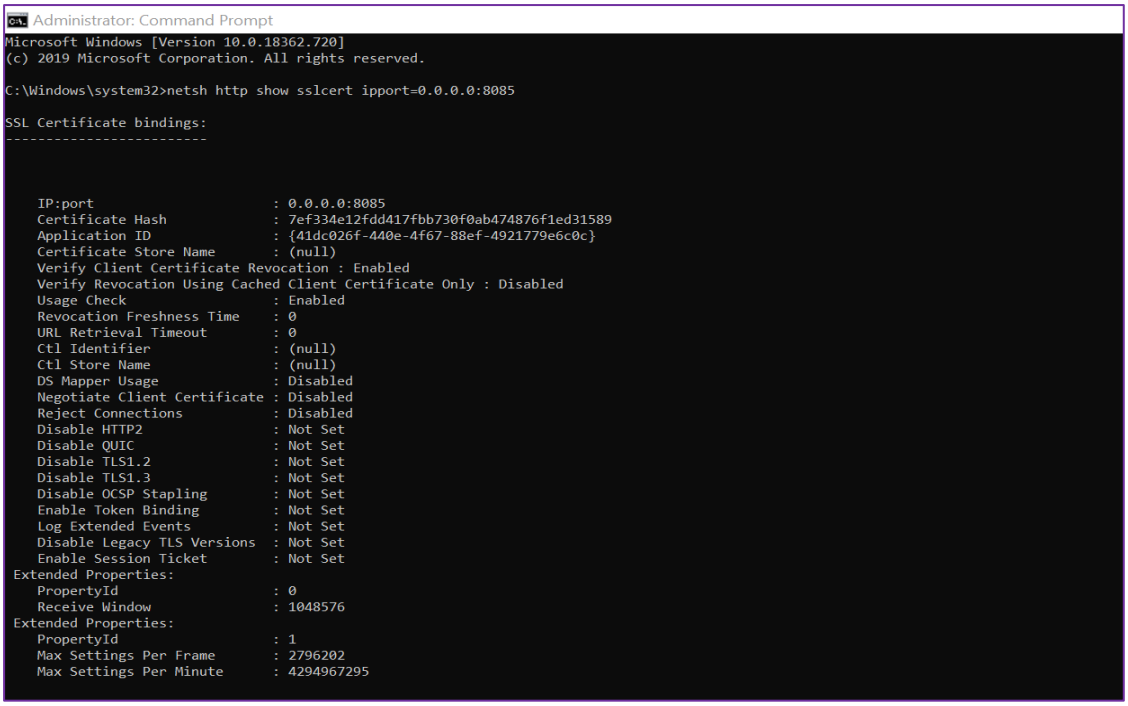
2.7 Bind certificate using command line

To bind the certificate using command line follow the below steps:

1. Run command prompt as Administrator and run command as: "**netsh http add sslcert ipport=0.0.0.0:8085 appid={41dc026f-440e-4f67-88ef-4921779e6c0c} certhash= 7ef334e12fdd417fbb730f0ab474876f1ed31589"** where 'sslcert ipport' is the port, 'appid' is the GUID of the service and 'certhash' is the Thumbprint of the created certificate. On successful run, the message 'SSL Certificate successfully added' will be displayed.



2. To show the bound certificate. Use the following command: "**netsh http show sslcert ipport=0.0.0.0:8085**"



3. This will now bind certificate to OCM TUserSessionManager successfully.

2.8 Setting up Securing Endpoints

To enable the SSL over Http protocol, follow the below steps:

1. For REST Service, under **binding** -> webHttpBinding enable security mode **Transport**.

```
<webHttpBinding>
  <binding name="webHttp">
  <security mode="Transport">
      <transport clientCredentialType="None"/>
    </security>
  </binding>
</webHttpBinding>
```

2. For WCF Service, under **binding** -> basicHttpBinding enable security mode **Transport**.

```
<basicHttpBinding>
  <binding name="basicHttps">
  <security mode="Transport">
      <transport clientCredentialType="None"/>
    </security>
  </binding>
</basicHttpBinding>
```

3. Add/change **httpsGetEnabled** behavior in the config for **REST** and **SOAP** and **httpGetEnabled** can be removed or set false.

```
<serviceBehaviors>
  <behavior name="soap">
  <serviceMetadata httpsGetEnabled="true"/>
  <serviceDebug includeExceptionDetailInFaults="False"/>
  </behavior>
 <behavior name="rest">
    <serviceMetadata httpsGetEnabled="true" />
      <serviceDebug includeExceptionDetailInFaults="False"/>
  </behavior>
</serviceBehaviors>
```

4. Change all host base address **http** to **https**.

3. Testing TUserSessionManager

- To test the service use Soap UI, import the below project xml:

TetherfiUserSession
Manager-soapui-pr

- To test the service use Postman, import the below collection:

TetherfiUserSession
Manager.postman_c