Authenticatr

Two-factor authentication made simple for Windows network environments

Version 0.9

USER GUIDE

Contents

Legal Stuff

We've tried to keep the legal stuff short and simple, but sadly some of it is unavoidable.

Any product or brand names are trademarks of their respective companies. This includes, but is not necessarily limited to Microsoft, .NET Framework, Google Authenticator and OATH.

You may distribute the demo version, although if you do so you should include a link or reference to https://www.authenticatr.net so users have a chance to find the latest version.

We do not offer any guarantees or warranty for the software beyond those explicit in English law. It's your own responsibility to take any backups of log files, user configuration, etc. Your statutory rights under English law are not affected by this.

By running the Authenticatr software you accept these terms. They are governed by the laws of England and you accept jurisdiction of the English courts. If you don't accept these terms then you must uninstall Authenticatr.

About Authenticatr

Thanks for your interest in Authenticatr. The software is designed to provide a simple way to implement two-factor authentication in a Windows network environment in order to provide improved security for user authentication.

Two-factor authentication is a method whereby the security of user authentication can be improved by ensuring that any user wishing to authenticate must have two different components or factors. This is an improvement over traditional password based systems where only a single factor (the password) is required.

In the most common forms of two-factor authentication the two factors are implemented as follows:

1. "What you have" – a physical device or token that must be possessed by the user wanting to authenticate
2. "What you know" – a secret known only by the user wanting to authenticate

The effectiveness of two-factor authentication relies on the fact that in most scenarios it is unlikely that a non-authorised individual will be in possession of both factors for the same user account.

Authenticatr is designed to allow systems administrators to quickly, easily and securely implement a two-factor solution in a Windows network environment by utilising the increasingly popular OATH TOTP standard for two-factor authentication. This is an open standard agreed back in 2011 and currently implemented and supported by a wide range of organisations including Microsoft, Google, RSA, Amazon, Facebook and others.

Authenticatr fulfils the requirement of two-factor authentication by enforcing the following two factors:

1. The user must possess either a hardware token or a smartphone capable of generating a constantly changing six-digit code (this fulfils the "what you have" factor)
2. The user must also know a PIN code which can be configured in Authenticatr and is unique to the user (this fulfils the "what you know" factor)

It is worth noting that a third factor can also be implemented if a smartphone is used as the "what you have" factor, as the smartphone can optionally be protected with a different PIN, fingerprint lock, or other mechanism to restrict access to the phone to an authorised user.

Installation and prerequisites

Microsoft .NET Framework

Before installing Authenticatr please ensure you have the Microsoft .NET Framework version 4.0 or greater installed on the computer you wish to install Authenticatr onto. As a general rule if you are installing Authenticatr onto a machine running either Windows 8 or greater, or Windows Server 2012 or greater, then Microsoft.NET Framework 4.0 should already be installed (although the relevant feature may need to be enabled on Windows Server)

For Windows Server users the Authenticatr program is fully functional in either Server Core or GUI mode.

If you do not already have Microsoft.NET 4.0 or greater installed or enabled then you will be prompted to download and install the relevant software as part of the Authenticatr installation process.

Running for the first time

The first time you run the Authenticatr software you will be prompted to automatically create the appropriate firewall rules that will allow Authenticatr to run properly. You can find out more about what rules are created or how to manually update firewall rules in Appendix A – Ports and Rules .

If you would like Authenticatr to automatically create the appropriate firewall rules then choose the 'Yes' option, otherwise choose 'No'.

Figure 1 – Allowing firewall rules

Once you have chosen an option you will then see the main Authenticatr control panel.

Authenticatr Control Panel

The Authenticatr Control Panel is the main screen from which you can monitor the health of the Authenticatr installation, configure system settings or add, edit or remove users from Authenticatr.

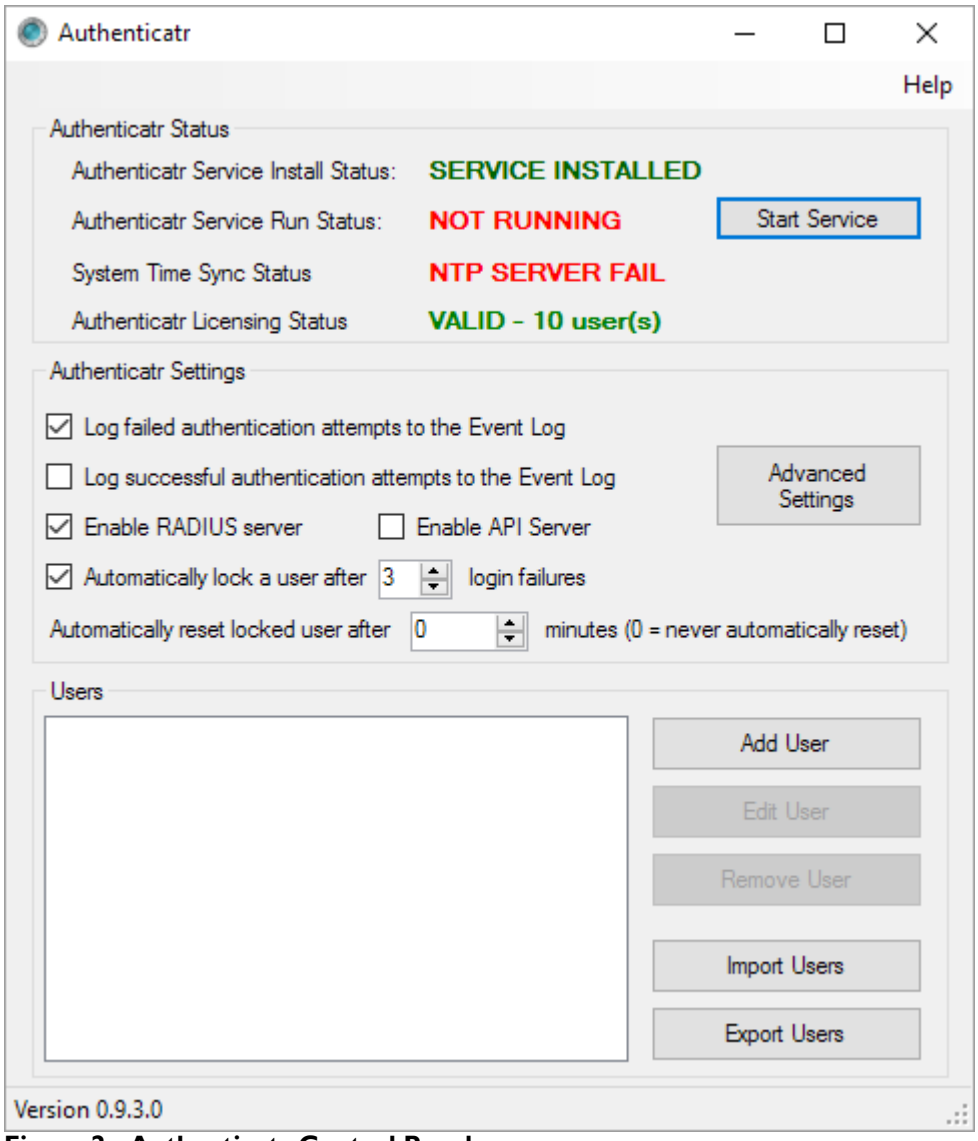

Figure 2 - Authenticatr Control Panel

The first part of the control panel provides an overview of the health of the Authenticatr installation. From here you can see if the Authenticatr system service has been installed, whether the service is running, whether the system running the Authenticatr service has an accurate time source and whether Authenticatr has been properly licensed.

If any of these four status options are showing as red then there is a problem which may need to be investigated in order for Authenticatr to function properly. Please refer to the troubleshooting section later on in this document.

Adding Users

In order for a user to be able to use two-factor authentication a record for that user must first be created within Authenticatr. This is done by clicking the 'Add User' button on the Authenticatr Control Panel window. This will start a wizard which will guide you through the steps necessary to add the new user to Authenticatr.

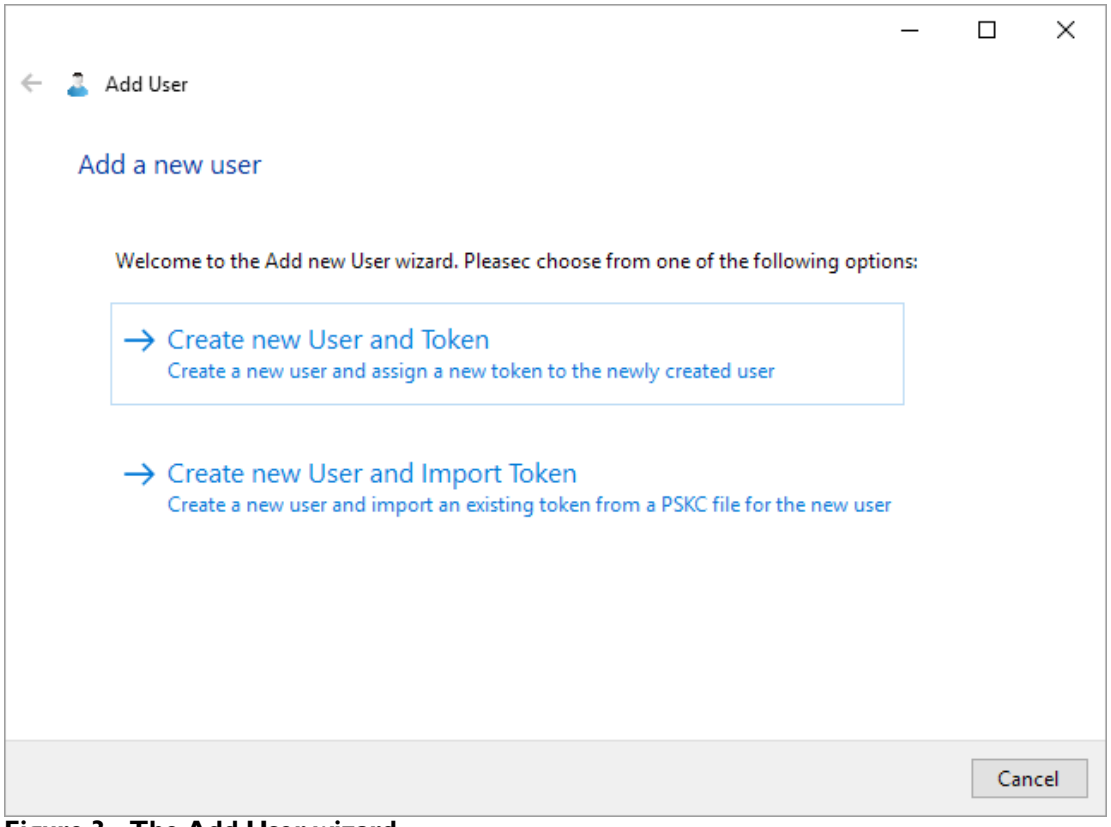

Figure 3 - The Add User wizard

Before you start it is important to know whether you will be creating a new software token for the newly created user or whether you will be importing an existing hardware or software token to be associated with the new user.

If you have an existing hardware token which has been supplied with initialisation information (usually in the form of either a 'seed value' or a 'token file' or similar) then you should choose the 'Create new User and Import Token' option. You should also choose this option if you are importing a user from another two-factor authentication system which provides token information in a standard format, or you are importing token information from another installation of Authenticatr.

In all other cases, including the most common scenario of creating a new user who is going to use a smartphone application such as Google Authenticator, you should choose the 'Create new User and Token' option.

Create new User and Token

If you have chosen the 'Create new User and Token' option then the next page of the wizard will ask you to enter basic information about the new user to be created.

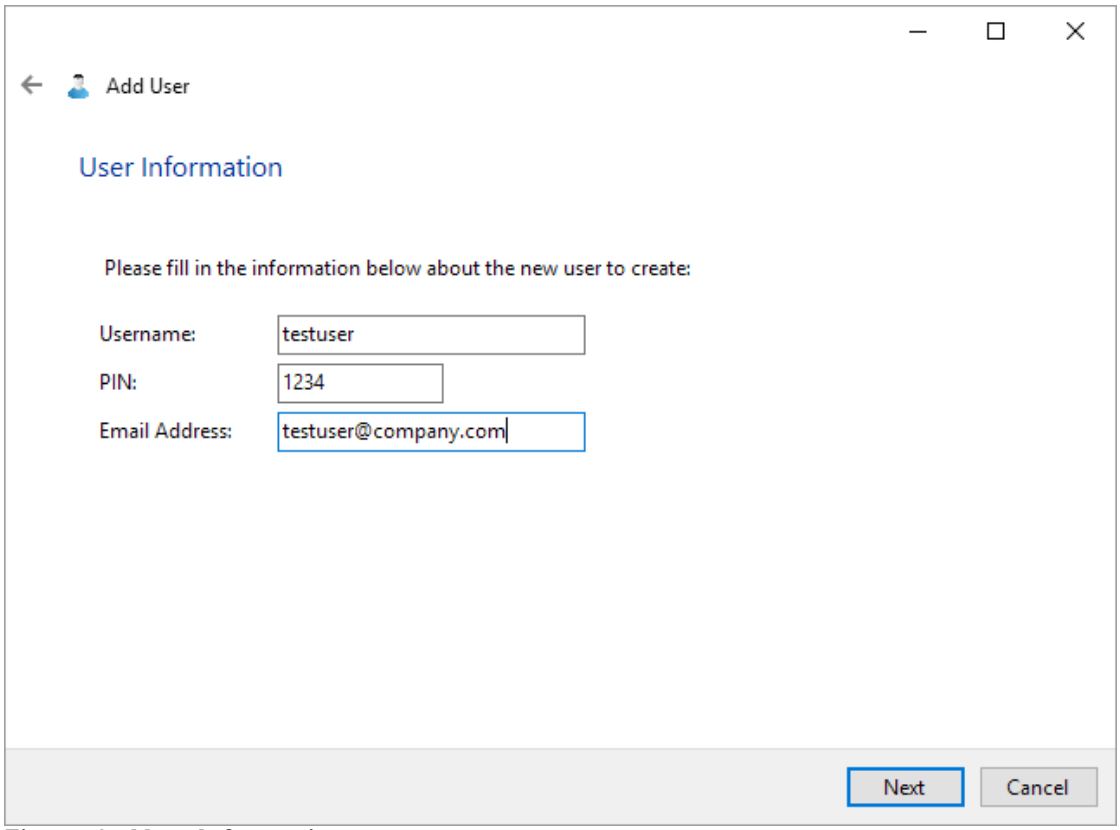

Figure 4 - User Information

The username must be unique to the user. This can be an existing username you use elsewhere or an entirely new username that is unique to Authenticatr. You can even use an email address as a username.

Note that if you are in a Windows domain network environment and you wish to use your existing usernames, you do not need to include the domain name (although there is nothing stopping you from doing this if you wish to do so).

The PIN is a numeric value which the user must prefix to their generated token every time they wish to identify themselves using Authenticatr. The PIN can be up to 9 digits long, although most users tend to find a four digit PIN easiest to remember.

Finally, an email address can optionally be specified for the new user. In current versions of Authenticatr this field is not used, but in future versions this value may be used to allow email notification or self-service PIN changes.

Once the user information has been added, the next page of the wizard will display the information that has previously been entered and ask the user to confirm the information.

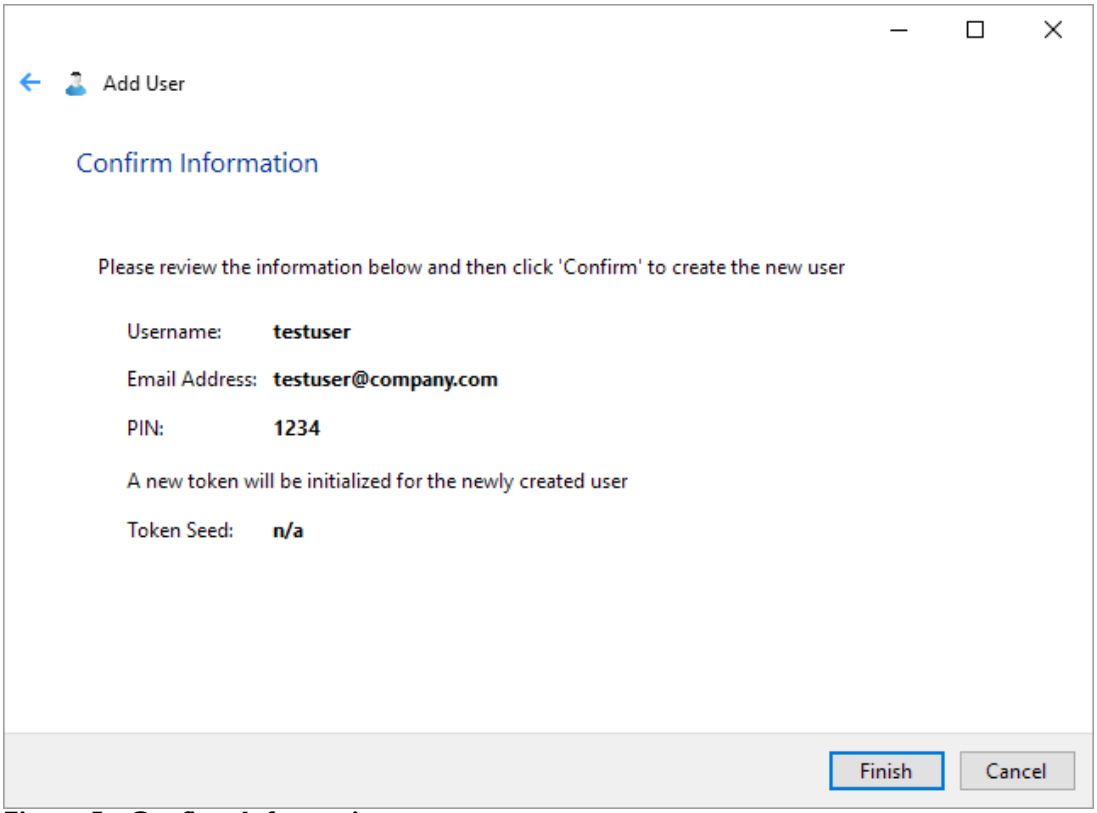

Figure 5 - Confirm Information

Once the 'Finish' button has been clicked the new user will be created, a new token will be generated, and you will be taken to the 'Token Summary' screen for the newly created user and token.

Create new User and Import Token

If you have chosen to import an existing token then the first screen of the wizard will be the same as the 'Create new User and Token' process. On this first screen you will enter the basic information about the new user.

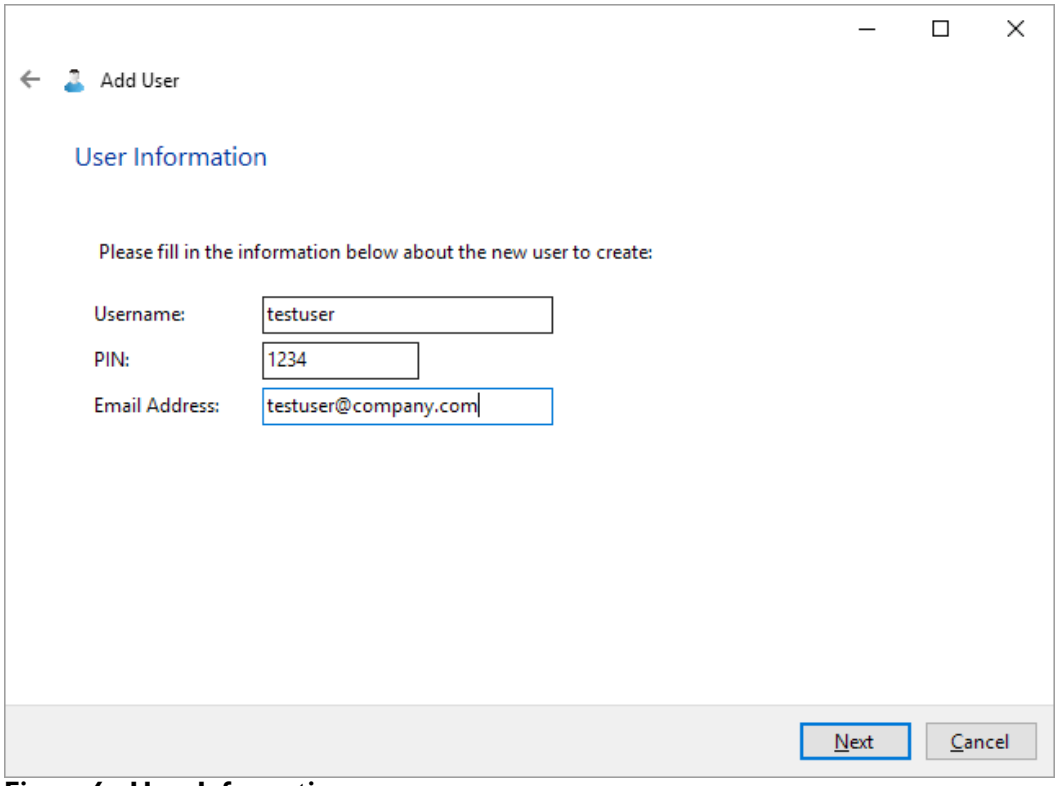

Figure 6 - User Information

Once you have entered the new user information and clicked 'Next' you will be asked how you wish to import the existing token to be assigned to this user.

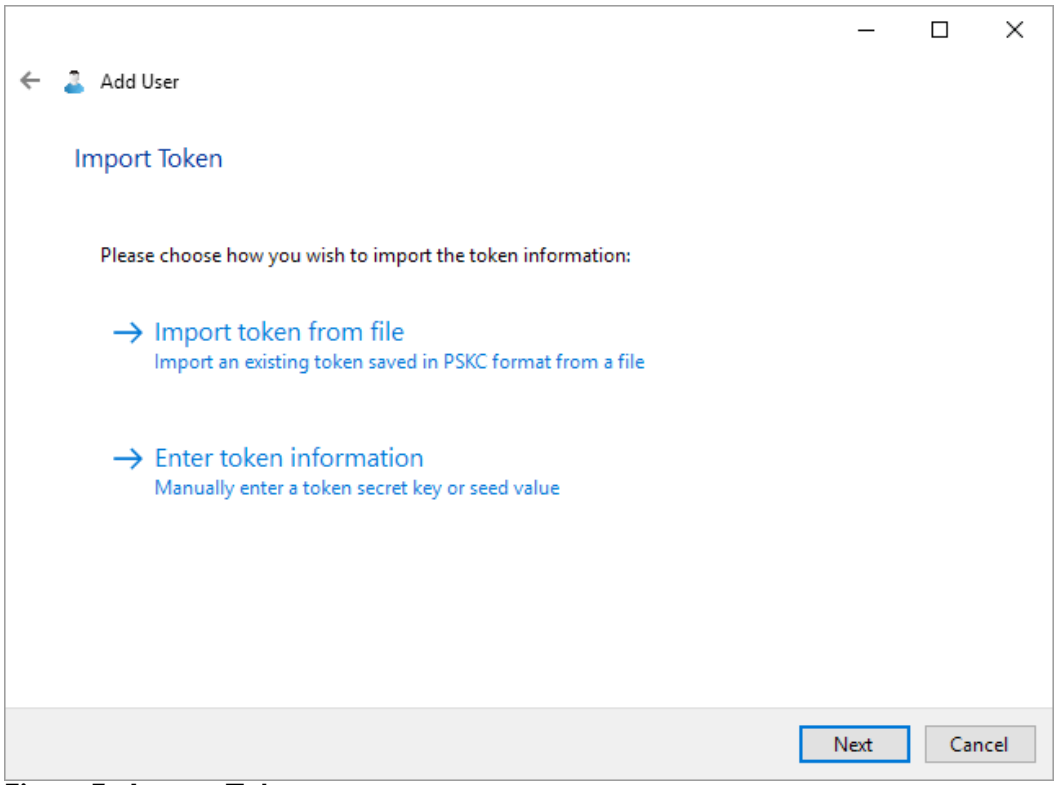

Figure 7 - Import Token

If you have been provided with a PSKC token file then you should choose the 'Import token from file' option.

If you have been provided the token information as plain text (either directly or inside a CSV or TXT file or similar) then you should choose the 'Enter token information' option.

Import token from file

As soon as you click the 'Import token from file' option you will be asked to select the token file. Browse to the file containing the token you wish to import and click the 'Open' button. If the file you wish to open does not have a '.pskc' extension then you may need to change the file type from 'Token files' to 'All files'.

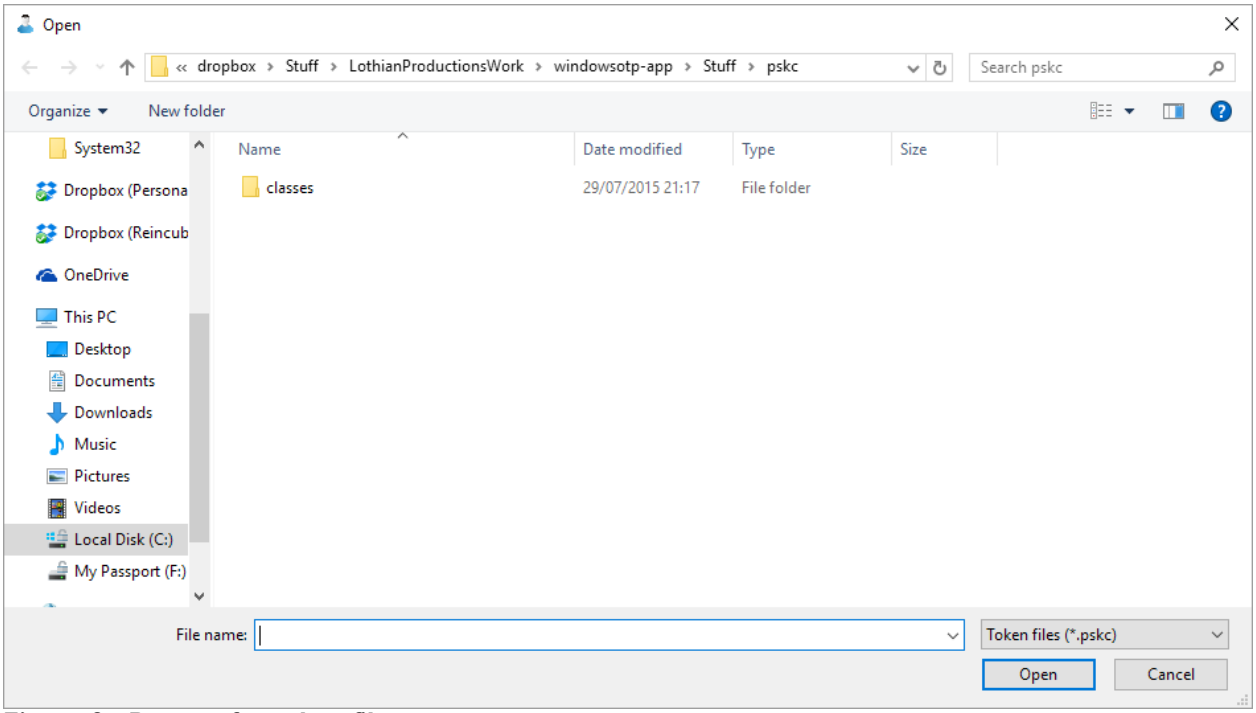

Figure 8 - Browse for token file

Once you have selected the token file it will be examined to determine if it can be opened by Authenticatr.

Many PSKC token files are encrypted for security and can only be opened with a passphrase. If the PSKC file you have selected has been protected in this way then you will be prompted to enter the password for the token file.

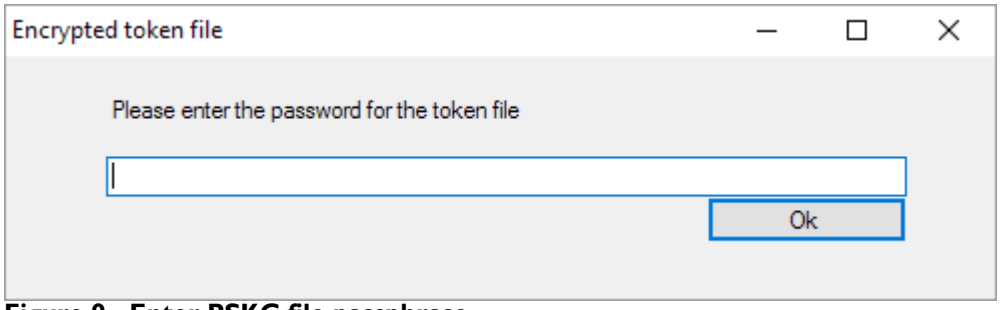

Figure 9 - Enter PSKC file passphrase

If you have entered the correct passphrase then you will see the summary screen. This will display the user information you have previously entered, and will also show the seed value for the token to be associated with the new user.

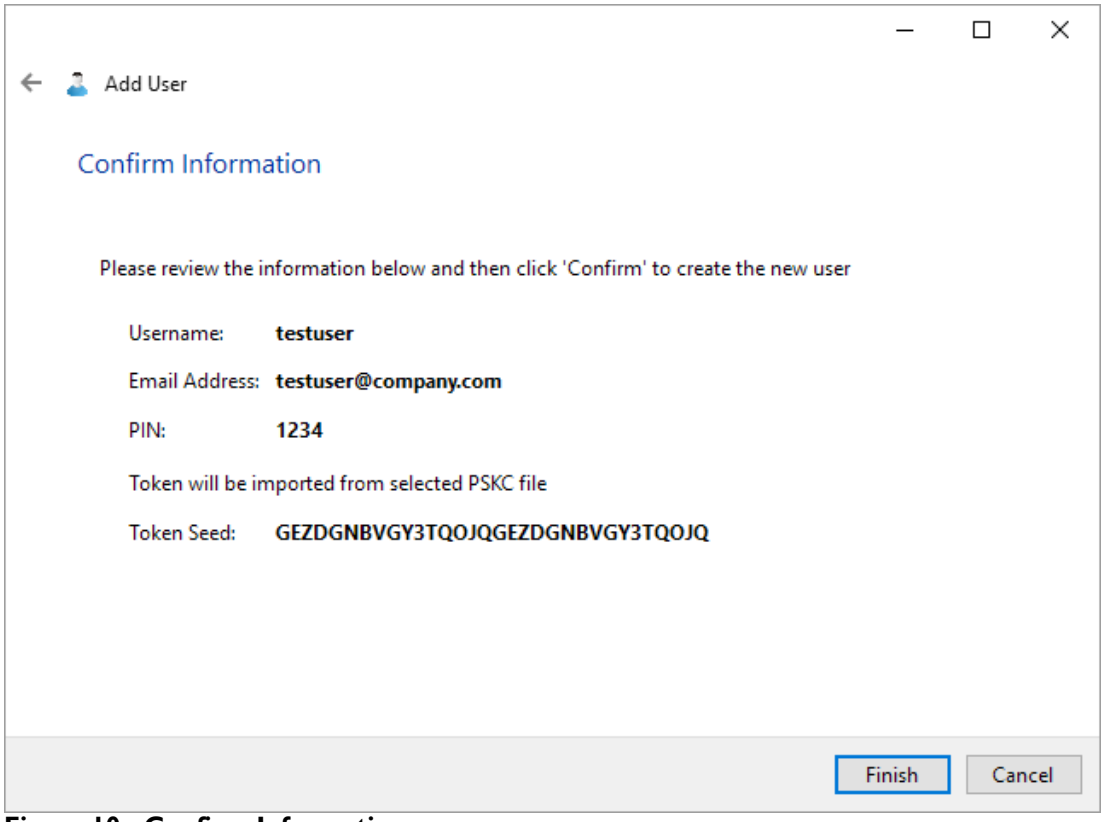

Figure 10 - Confirm Information

If you are happy with the information that is being displayed, click the 'Finish' button to create the new user and assign the token to that user. The wizard will then close and you will automatically be taken to the Token Summary screen for the new user.

Enter token information

If you have chosen to manually enter token information then you will see a token entry screen.

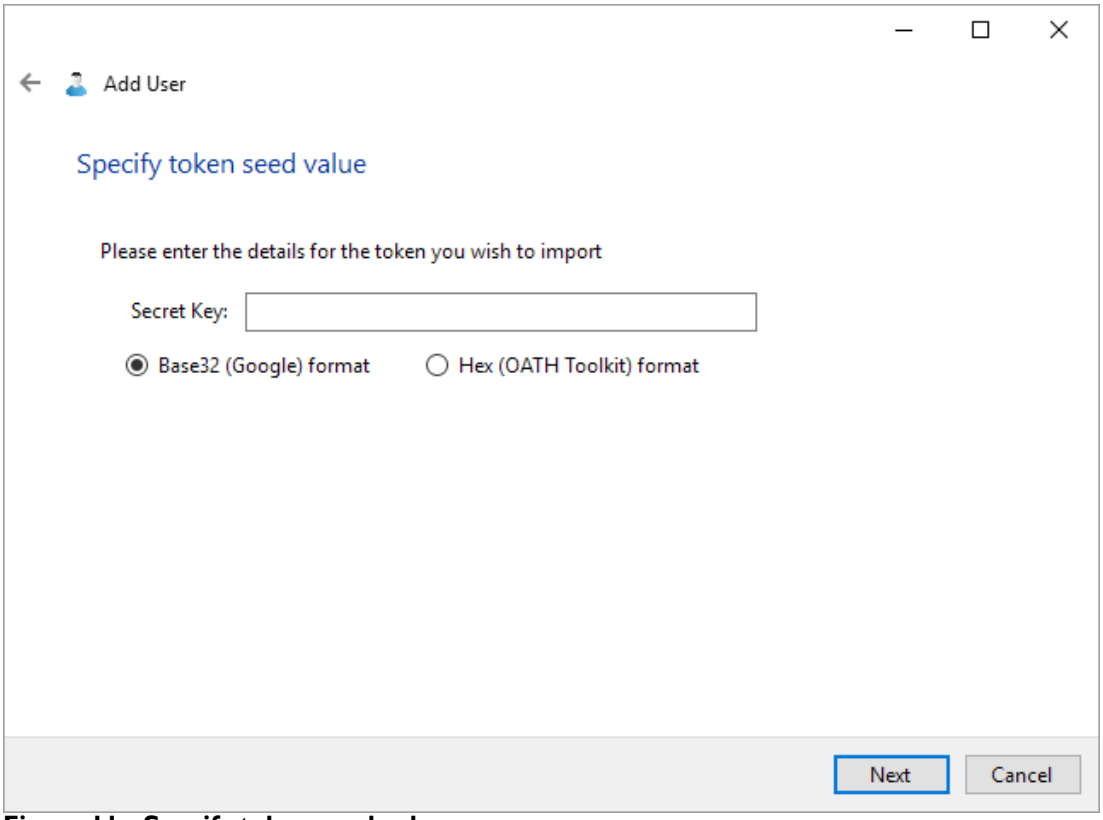

Figure 11 - Specify token seed value

You should enter the supplied token seed value into the 'Secret Key' field. If you are importing an existing token from Google Authenticator, or other similar software which records tokens in Base32 format, then you can leave the key format set to the 'Base32 (Google) format' option.

However if you have created your token file using a third party product that uses the Hex key format (such as OATH Toolkit) then you will need to choose this option.

If you have been supplied with seed information by a third party hardware token manufacturer then you may need to check with them what format they have supplied the seed value information in.

Tip: If the seed value you have been provided contains the letters G through Z anywhere in the value, then it cannot be in Hex format and thus you should choose the 'Base32 (Google) format' option.

Once you have entered the seed value click the Next button to proceed to the summary screen.

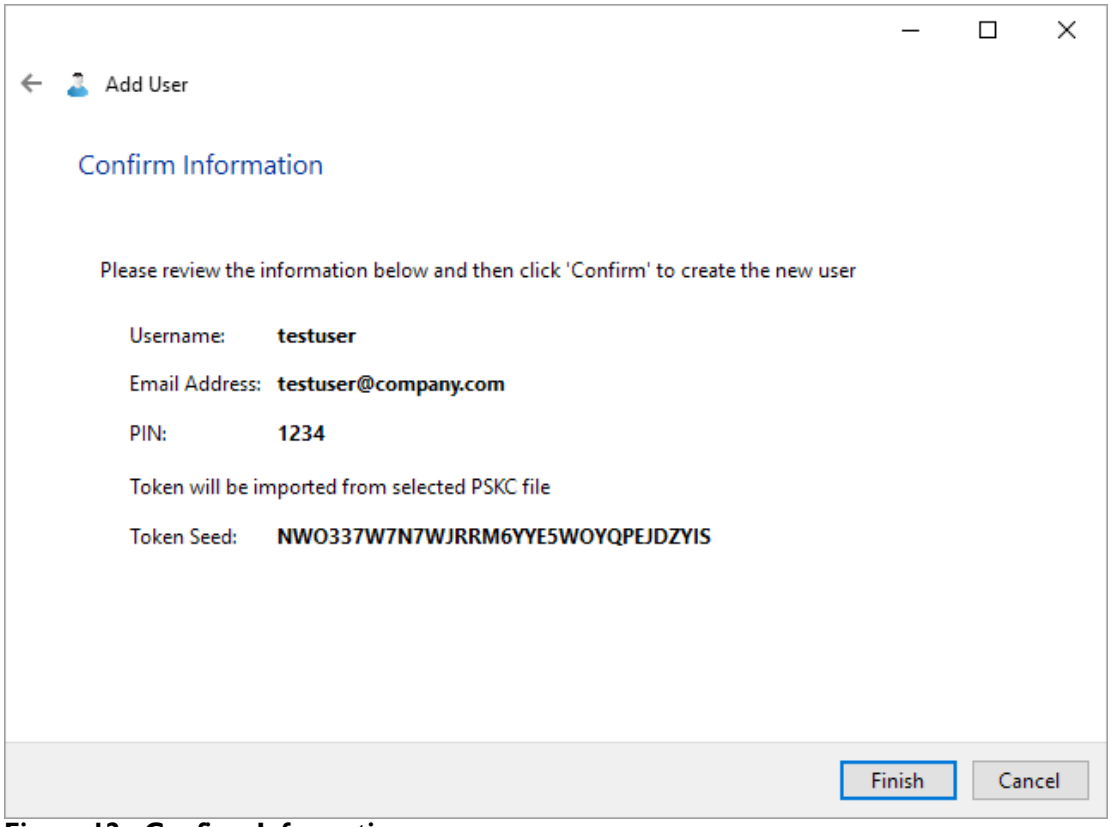

Figure 12 - Confirm Information

As with the 'Import Token from File' option, the confirm information screen will show the user information you have previously entered as well as the seed value you have specified to be associated with the new user.

Note: Seed values are always shown in Base32 format. If your original token information was in Hex format then the value shown on the 'confirm information' screen may not match the value you entered on the previous screen as it will have been automatically converted to Base32 format for you.

Once you are happy the information is correct you can click the 'Finish' button to create the new user and associate the token information with the newly created user. The wizard will then close and you will be taken to the 'Token Summary' for the newly created user.

Token Summary

The Token Summary screen shows a snapshot of information about a given user and their token. This information can be used to test a token, to verify that the token data is accurate or to program a third-party application such as Google Authenticator.

Figure 13 - Token Summary

The current TOTP value for the token is shown as the six digit number under the 'Current Code' heading.

To the left of this is a QR code which can be scanned on most modern smartphones to automatically program the token into third-party authentication applications such as Google Authenticator or Microsoft Authenticator or similar.

At the bottom of the screen is the Authorisation Key, also known as the Token Seed Value. Some older authentication software which does not have the capability to scan QR codes may require this value to be entered instead.

Note: This value is highly sensitive and should not be disclosed to anyone else. Also, if you do not have a separate backup of the Authenticatr database, or you wish to maintain an offline backup of your token values, you may wish to keep a copy of this value (securely!) in order to avoid having to re-provision a new token on to your devices in the event the Authenticatr installation is lost or damaged.

Advanced Settings

In order to use some advanced features of Authenticatr it may be necessary to configure some options within the Advanced Settings screen. This screen is accessed by clicking the 'Advanced Settings' button within the main Authenticatr control panel window.

Please be aware that these settings should only be changed if you know what you are doing, as incorrect values here may cause Authenticatr not to work properly.

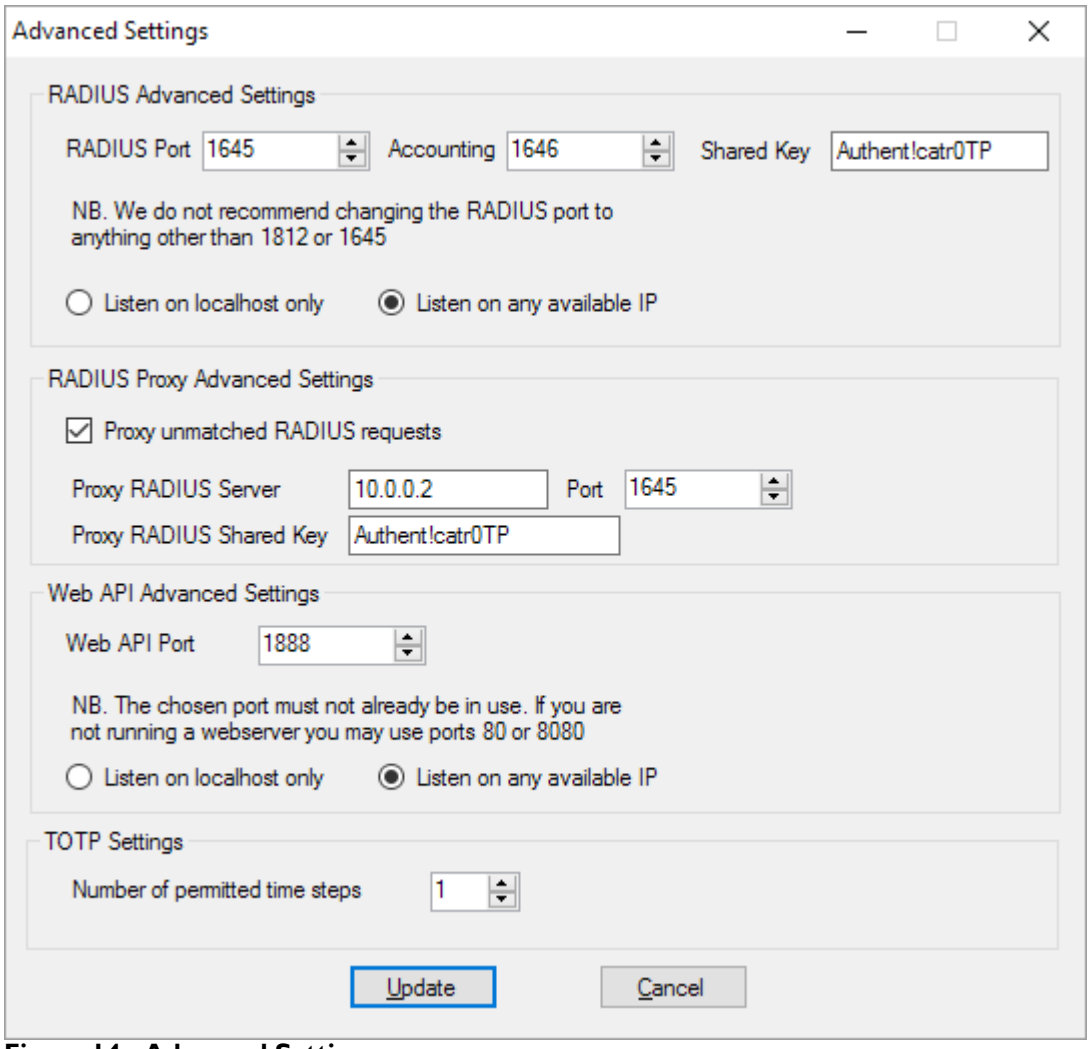

Figure 14 - Advanced Settings

The various settings are grouped under a number of headings, each of which is described below.

RADIUS Advanced Settings

It is possible to use Authenticatr as a RADIUS Server so that any third-party software or equipment which can act as a RADIUS Client can use Authenticatr to provide authentication services. This feature is enabled by default, but will require

some configuration, which is done by selecting the appropriate options in this section of the Advanced Settings screen.

RADIUS Port – This is the TCP port number on which Authenticatr will listen for incoming RADIUS connections. By default this is set to 1645 which is the standard RADIUS port for older network equipment and is often used for compatibility purposes. Some newer network equipment may require this to be changed to port number 1812. If you already have another RADIUS server on the same machine you have installed Authenticatr on then you can choose any free port number.

NB. If you change the port number, when Authenticatr next starts it will attempt to open the appropriate rule in the Windows Firewall. However if you have a hardware firewall you will need to ensure that the relevant port is opened for incoming TCP connections from your RADIUS Client device to the Authenticatr server.

Accounting Port – The RADIUS protocol allows for audit messages to be sent on any authentication attempt – a process known as RADIUS Accounting. Although Authenticatr does not implement RADIUS Accounting some RADIUS Client devices will not work properly unless they can access a valid RADIUS Accounting port. By default Authenticatr will use port 1646 for compatibility. However as with the RADIUS Port above, some newer equipment may require this port to be changed to port number 1813 . If you are using custom port numbers then the RADIUS Accounting port number is usually one number greater than the main port number chosen for RADIUS authentication.

Shared Key – RADIUS secures communication between the RADIUS Client and the RADIUS Server by means of a shared secret key. This is a value which is set on both client and server and must match on both. By default Authenticatr will suggest a value of *Authent!cator0TP* but we strongly recommend you change this for a unique random string.

NB. If you change the Shared Key in Authenticatr don't forget to update the key on your RADIUS Client device or application

Listen On … - By default Authenticatr will listen for RADIUS connections on all IP addresses bound to the machine Authenticatr has been installed on. However if your RADIUS Client software is running on the same machine as Authenticatr then you may wish to change this option so Authenticatr will only listen for RADIUS traffic on the localhost (127.0.0.1) IP address. This may improve security as it will prevent malicious third party RADIUS traffic from being heeded by Authenticatr.

RADIUS proxy Advanced Settings

One of the most useful features of Authenticatr is that it can be used as a RADIUS Proxy as well as a RADIUS Server. For more information on this process see the chapter entitled 'Migrating from existing authentication solutions' .

Proxy unmatched RADIUS requests – If you wish to enable the RADIUS Proxy then ensure this option is ticked. If the RADIUS Proxy is enabled you must configure the proxy information below this option.

Proxy RADIUS Server – The IP address or hostname of the RADIUS Server you wish to proxy requests to.

Proxy RADIUS Port – The port number of the RADIUS Server you wish to proxy requests to. This does not have to be the same as the port number you wish to use Authenticatr on, provided the destination RADIUS Server is not on the same machine.

Proxy RADIUS Shared Key – The shared key defined on the RADIUS Server you wish to proxy RADIUS traffic to.

NB. If you wish to use the RADIUS Proxy option ensure that your destination RADIUS Server is configured to allow incoming RADIUS communications from the server that Authenticatr is installed on, and that any firewalls in-between are configured to allow such traffic.

Web API Advanced Settings

As well as acting as a RADIUS Server, Authenticatr also has the option of a simple web-based API. This allows developers to incorporate Authenticatr into existing internal web or desktop applications to provide secure two-factor authentication services.

Web API Port - The port number that Authenticatr will listen for incoming API requests on. This defaults to port 1888 but can be changed to any free TCP port. You can use ports 80 or 8080, but only if you do not already have a webserver on those ports.

Listen On … - Specifies whether the Authenticatr Web API should listen on all available IP addresses on the server where Authenticatr has been installed, or just on the localhost (127.0.0.1) IP. Only use the latter if you will only ever be accessing the Web API from the same machine where Authenticatr has been installed.

TOTP Settings

By default Authenticatr supports the standard TOTP options as defined in the relevant RFC standards, or as recommended by the OATH standards group. However in some instances it may be desirable to change the default options for ease of use or to maintain compatibility with legacy systems or applications.

Number of permitted time steps – Controls how many time steps (forward or backwards) Authenticatr will check when validating an incoming TOTP request. By default this is set to 1, meaning that as well as the current TOTP window (remember that a default TOTP window is 30 seconds), Authenticatr will also allow tokens which match the next window and the previous window. When using the default 30 second windows, this means in practise that Authenticatr will allow

requests from TOTP clients where their internal clock could be up to 30 seconds out from that on the Authenticatr server.

It may sometimes be desirable to increase this window, especially where it is known that the client devices used to generate TOTP tokens do not have accurate clocks. However increasing the number of allowed time steps does present a potential vulnerability as it increases the number of potential token values that Authenticatr will accept.

NB. Increasing this value does not increase the risk from a replay attack (where a token value is captured in real-time by an attacker and then 'replayed' at a later date) as Authenticatr has built-in anti-replay facilities to prevent token re-use.

Using Authenticatr as a RADIUS Server

Once you have installed Authenticatr and have added at least one user to the system you can then start using Authenticatr to implement two-factor authentication within your organisation.

How you actually do this will depend on exactly what software or network infrastructure you wish to protect with two-factor authentication. In order to work with as wide a range of third-party software and network equipment as possible, Authenticatr supports the industry-standard RADIUS authentication system by acting as a RADIUS Server.

What this means is that anything that can use RADIUS as a client for authentication purposes can be connected to Authenticatr. Then whenever your existing RADIUS Client device needs to authenticate a user, the authentication information will be passed to Authenticatr. Authenticatr will then verify the information and return a pass or fail back to the RADIUS device which can then handle the authentication request appropriately.

To configure Authenticatr to support RADIUS authentication, first open the Authenticatr Admin system and ensure the 'Enable RADIUS Server' option is ticked. This option is normally ticked by default.

You may then need to set or check appropriate RADIUS settings in the 'Advanced Settings' screen – see the Advanced Settings chapter of this user guide for more information on how to do this.

Once you have configured Authenticatr as a RADIUS Server you will need to configure your RADIUS Client. Full details on how to configure RADIUS Clients is beyond the scope of this guide. However we are constantly producing fact sheets which explain how to configure RADIUS on a variety of popular networking equipment, which can be found on our website at www.authenticatr.net

Using the Authenticatr Web API

As well as acting as a RADIUS Server, Authenticatr also provides a basic Web API which can be used to authenticate users and validate their TOTP tokens in internal web and desktop applications, adding powerful two-factor authentication to complement existing login processes.

To use the Web API, go into the Authenticatr control panel and ensure the 'Enable API Server' option is ticked. By default Authenticatr will use TCP port 1888 for the Web API – this can be changed in the 'Advanced Settings' – see the relevant chapter of this user guide for more information.

Implementing the Web API is simple. Simply POST a form containing the following two fields to the root URI (/) on the relevant host/port:

user – The username of the user to authenticate
pass – A salted hash of the user's PIN and TOTP token

To generate the salted hash of the user's PIN and TOTP token for use in the *pass* field, simply generate an MD5 hash of the following three data items concatenated together

username + PIN + TOTP token value

For example, assuming the username is *alice*, the PIN is *1234* and the TOTP token value is *987654* :

1. Concatenate the values to make the string *alice1234987654*
2. Generate an MD5 hash of the string *alice1234987654*
3. Send an HTTP POST with the following form data to the Web API:
    a. *user = alice*
    b. *pass = 750d1a4f550a352d651895975d25a6aa*

NB. The pass field is case insensitive. Also the use of MD5 to generate the pass field does not represent a security issue as the MD5 algorithm here is being used to generate a hash, not to guarantee uniqueness, and thus the implementation in the API is not vulnerable to known MD5 weaknesses.

Once a form has been submitted to the Web API the information will be decoded and verified by Authenticatr. The following response string will be returned to indicate success or failure:

OK = Authentication has succeeded and the user should be authorised/permitted to proceed
FAIL = Authentication was not successful. The user should not be permitted to proceed.

To prevent information leakage further details about the reason for any authentication will not be returned in the API response. However assuming failure logging has been left enabled in the Authenticatr control panel, this information will be available in the Authenticatr event log.

Migrating from an existing authentication solution

Authenticatr has been designed to make it as easy as possible to migrate from an existing RADIUS-based authentication solution such as Cryptocard or suchlike. This is done by configuring Authenticatr to pass through any unmatched authentication requests to the existing authentication system. This allows a staged deployment of Authenticatr to take place without having to migrate all existing users in one "big bang" style deployment.

Whilst the migration process is in place, unmatched authentication requests will pass as follows:

Authentication Client (e.g Firewall) → Authenticatr Server → Existing Authentication System (e.g Cryptocard)

The basic process for migration is typically as follows:

1. Install Authenticatr but do not configure any users

2. Use the RADIUS Proxy feature in the Advanced Settings to configure Authenticatr to forward any unmatched authentication requests to the existing RADIUS authentication system

   NB: Ensure you configure the existing RADIUS authentication system to accept incoming authentication requests from the IP address of the Authenticatr server, and ensure the RADIUS shared secret in the RADIUS Proxy section of the Authenticatr Advanced Settings dialog matches that of the existing RADIUS authentication system

3. Test pass-through by configuring a RADIUS client to point to the Authenticatr server but using the credentials of the existing RADIUS authentication system. Since the username will not be matched by Authenticatr, the authentication request should automatically be forwarded to the existing RADIUS authentication system and the response passed back via Authenticatr to the test client

   NB. If you do not have a suitable existing method of testing RADIUS authentication requests you can use the NTRadPing tool available for free download at https://www.novell.com/coolsolutions/tools/14377.html

4. Once pass-through has been confirmed as working, reconfigure any existing authentication clients to point to the Authenticatr server instead of the existing RADIUS authentication system.

5. Users can now be set up in Authenticatr. Once a user has been created, any authentication request for that user will be handled by Authenticatr and not

passed on to the existing RADIUS authentication system. This allows for users to be migrated individually at your leisure.

6. Once all users are running on Authenticatr, use the Advanced Settings in Authenticatr to disable RADIUS Proxying.

7. Finally, the existing RADIUS authentication system can be decommissioned.

Following the above process should mean there is no downtime at any stage during the migration process. However care should be taken to ensure that the RADIUS Proxy options are properly set and tested (step 3 above) before any existing authentication clients are switched to point at Authenticatr. Depending on the specific authentication client in use this may involve changing the RADIUS Port and Shared Secret as well as changing the IP address of the RADIUS Server.

Troubleshooting and Support

If you have problems with Authenticatr there are a number of initial steps you can take to resolve most issues. If these steps do not help you resolve your issues then you can contact us for assistance.

Cannot install Authenticatr

Please make sure you have the Microsoft .NET Framework (version 4.0 or greater) installed. Ensure you have installed any relevant updates from http://update.microsoft.com and have restarted your computer.

Warning about time sync

The underlying TOTP protocol used by Authenticatr relies on an accurate system time. If this is more than a few seconds out from the client devices being used to generate authentication codes then this can result in problems authenticating.

Authenticatr will attempt to check the system time and display a warning if the time cannot be verified. If such a warning is displayed ensure that the system time on the server running the Authenticatr software is being synced with an accurate time source using NTP or some other similar time sync protocol.

NB. Time sync can be a particular problem if the server running Authenticatr is a virtual server. Please check the documentation for your virtualisation solution to ensure the time in the virtual server is being correctly synchronised.

Authenticatr service not running

As part of the Authenticatr installation process a system service is deployed and set to run automatically when the system is started. If this service is disabled then Authenticatr will not function properly. If this is the case, ensure the *Authenticatr Windows OTP Service* is set to run automatically, and that the service is not blocked from starting by any system policies.

Support

Email

In the first instance please email help@authenticatr.net for any support issues or order queries. We aim to respond to any email queries within 2 working days. If you are a licensed user please include your license details in your email so we can confirm your entitlement to support.

Post

Please address any postal enquiries to:

Andrew Dancy
Lothian Productions c/o Reincubate Ltd
11 Old Jewry
London
EC2R 8DU